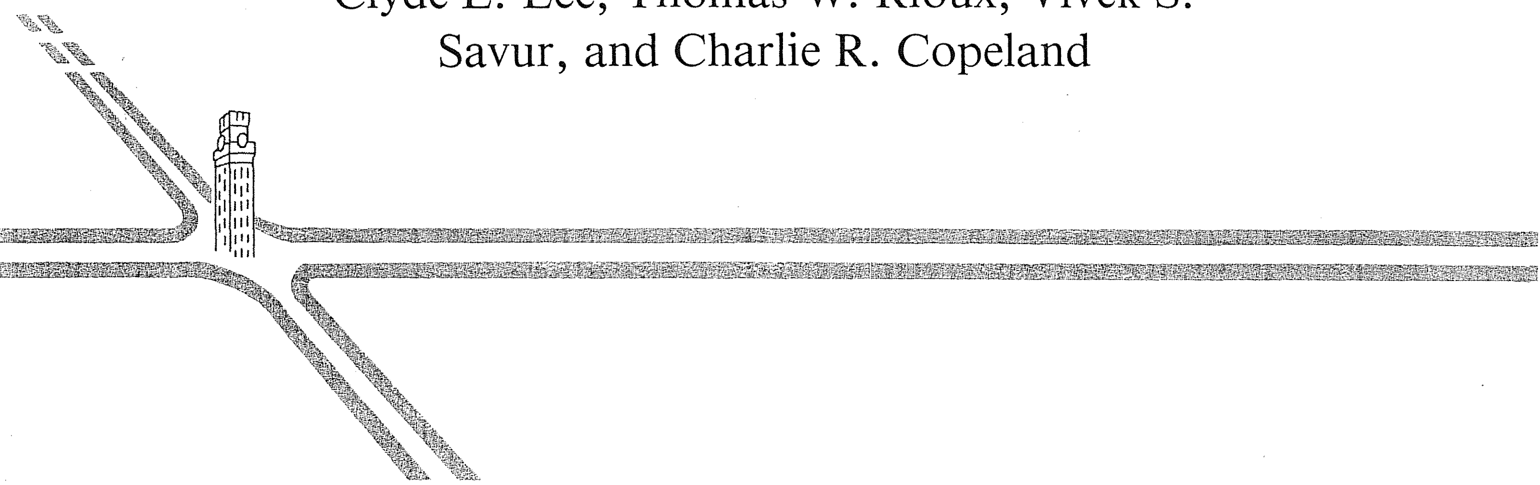
CFHR 3-18-72-184-2

THE TEXAS MODEL FOR INTERSECTION TRAFFIC — PROGRAMMER'S GUIDE

By

Clyde E. Lee, Thomas W. Rioux, Vivek S. Savur, and Charlie R. Copeland



RESEARCH REPORT 184-2

PROJECT 3-18-72-184

COOPERATIVE HIGHWAY RESEARCH PROGRAM
WITH TEXAS
STATE DEPARTMENT OF HIGHWAYS
AND PUBLIC TRANSPORTATION
AND
U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL HIGHWAY ADMINISTRATION

CENTER FOR HIGHWAY RESEARCH
THE UNIVERSITY OF TEXAS AT AUSTIN
DECEMBER 1977

PARTIAL LIST OF RESEARCH REPORTS PUBLISHED BY THE CENTER FOR HIGHWAY RESEARCH

Requests for these reports should be made to Mr. Phillip L. Wilson, State Planning Engineer, Transportation Planning Division, File D-10R, State Department of Highways and Public Transportation, P.O. Box 5051, Austin, Texas 78763.

20-1F "Speed of Vehicles on Grades," by C. Michael Walton and Clyde E. Lee, August 1975.

21-1F "A Performance Survey of Continuously Reinforced Concrete Pavements in Texas," by B. Frank McCullough and Pieter J. Strauss, November 1974.

29-1 "A Guide to the Selection of High-Strength Anchor Bolt Materials," by G. B. Hasselwander, J. O. Jirsa, and J. E. Breen, October 1974.

52-1F "Design Criteria for Overhanging Ends of Bent Caps - Bond and Shear," by Phil M. Ferguson, August 1964.

53-1F "Development Length for Anchor Bolts," by John E. Breen, April 1964.

54-1F "A Portable Electronic Scale for Weighing Vehicles in Motion," by Clyde E. Lee and Nasser I. Al-Rashid, April 1968.

55-3F "Critical Mechanical Properties of Structural Lightweight Concrete and the Effects of These Properties on the Design of the Pavement Structure," by William B. Ledbetter, Ervin S. Perry, James T. Houston, and J. Neils Thompson, January 1965.

56-26 "Application of Slab Analysis Methods to Rigid Pavement Problems," by Harvey J. Treybig, W. Ronald Hudson, and Adnan Abou-Ayyash, May 1972

56-27 "A Summary of Discrete-Element Methods of Analysis for Pavement Slabs," by W. Ronald Hudson, Harvey J. Treybig, and Adnan Abou-Ayyash, August 1972.

56-28 "Finite-Element Analysis of Bridge Decks," by Mohammad R. Abdelraouf and Hudson Matlock, August 1972.

56-29F "Final Report for Project 3-5-63-56," by John J. Panak, August 1973.

73-5F "Development of a System for High-Speed Measurement of Pavement Roughness, Final Report," by Roger S. Walker, W. Ronald Hudson, and Freddy L. Roberts, May 1971.

76-1F "Fatigue Strength of 3/4-Inch Studs in Lightweight Concrete (Push-out Tests)," by H. G. Lehman, H. S. Lew, and A. A. Toprac, May 1965.

77-2F "Fatigue Tests of Welded Hybrid Plate Girders Under Constant Moment," by H. S. Lew and A. A. Toprac, January 1967.

78-1F "Evaluation of Traffic Control at Highway Intersections," by Clyde E. Lee and Walter C. Vodrazka, March 1970.

79-1F "Numerical Methods for Radial Triangulation," by Robert D. Turpin, May 1966.

80-1F "Strength Effect of Cutting Off Tension Bars in Concrete Beams," by Phil M. Ferguson and Syed I. Husain, June 1967.

88-1F "Factors Affecting Anchor Bolt Development," by D. W. Lee and J. E. Breen, August 1966.

89-10 "Lateral Load Behavior of Drilled Shafts," by Robert C. Welch and Lymon C. Reese, May 1972.

89-11F "Criteria for the Design of Axially Loaded Drilled Shafts," by Lymon C. Reese and Michael W. O'Neill, August 1971.

91-1F "Shear Strength of Bent Caps Between Columns," by Phil M. Ferguson and Huey M. Liao, September 1966.

92-5F "Performance of Single and Double Sills for Steep Circular Culverts," by Manam V. P. Rao, Robert J. Brandes, and Frank D. Masch, January 1971.

93-1F "Evaluation of Control Extension," by Robert D. Turpin, David R. Knowles, and J. Stephen Carroll, September 1966.

94-3F "Behavior of Concrete Slab and Girder Bridges," by E. V. Leyendecker and J. E. Breen, May 1969.

96-5F "Fatigue Tests of Hybrid Plate Girders," by A. A. Toprac and M. Natarajan, September 1970.

97-1F "Static Tests on Hybrid Plate Girders," by H. S. Lew and A. A. Toprac, July 1967.

98-10 "Practical Method of Conducting the Indirect Tensile Test," by James N. Anagnos and Thomas W. Kennedy, August 1972.

98-11 "Improved Tensile Strength for Cement-Treated Bases and Subbases," by Robert F. Cauley and Thomas W. Kennedy, December 1972.

98-13 "A Comprehensive Structural Design for Stabilized Pavement Layers," by William O. Hadley, W. Ronald Hudson, and Thomas W. Kennedy, August 1972.

98-14F "Tensile Properties of Subbases for Use in Rigid Pavement Design," by Thomas W. Kennedy and W. Ronald Hudson, February 1973.

102-1F "Flexural Crack Width at the Bars in Reinforced Concrete Beams," by Syed I. Husain and Phil M. Ferguson, June 1968.

108-1F "A Theoretical and Experimental Study of Dynamic Highway Loading," by Nasser I. Al-Rashid, Clyde E. Lee, and William P. Dawkins, May 1972.

112-1F "Corrosion of Reinforcing Steel Embedded in Structural Concrete," by James T. Houston and Ergin Atimtay, March 1972.

113-5F "An Investigation of Creep Due to Bond Between Deformed Bars and Concrete," by Richard W. Furlong, Wing-Cheung Wong, and Sher Ali Mirza, August 1971.

114-1 "Polymer-Impregnated Concrete for Highway Applications," by David W. Fowler, James T. Houston, and Donald R. Paul, February 1973.

114-2 "Corrosion Protection of Reinforcing Steel Provided by Polymer-Impregnated Concrete," by David W. Fowler, Donald R. Paul, and Piti Yimprasert, December 1974.

114-3 "Repair of Concrete with Polymers," by M. M. Jaber, David W. Fowler, and Donald R. Paul, February 1975.

114-5 "Partial Polymer Impregnation of Center Point Road Bridge," by Ronald Webster, David W. Fowler, and Donald R. Paul, January 1976.

115-1F "Verification of Computer Simulation Methods for Slab and Girder Bridge Systems," by A. F. Alani and J. E. Breen, August 1971.

116-2F "Design Aspects and Performance Characteristics of Radial Flow Energy Dissipators," by Khosrow Meshgin and Walter L. Moore, July 1970.

117-3F "Analysis of Foundation with Widely Spaced Batter Piles," by Katsuyuki Awoshika and Lymon C. Reese, February 1971.

118-6 "Measurements of a Swelling Clay in a Ponded Cut," by Gordon Watt and Malcolm L. Steinberg, June 1972.

118-7 "The Waco Ponding Project," by Robert L. McKinney, Jr., James E. Kelly, and Chester McDowell, January 1974.

118-8 "Continuing Measurements of a Swelling Clay in a Ponded Cut," by Malcolm L. Steinberg, December 1974.

121-2 "Epoxy Resins for Jointing Segmentally Constructed Prestressed Concrete Bridges," by S. Kashima and J. E. Breen, August 1974.

121-3 "The Design and Optimization of Segmentally Precast Prestressed Box Girder Bridges," by G. C. Lacey and J .E. Breen, August 1975.

121-4 "Computer Analysis of Segmentally Erected Precast Prestressed Box Girder Bridges," by R. C. Brown, Jr., N. H. Burns, and J. E. Breen, November 1974.

(Continued inside back cover)

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWATX78-184-2		

4. Title and Subtitle	5. Report Date December 1977 6. Performing Organization Code
THE TEXAS MODEL FOR INTERSECTION TRAFFIC - PROGRAMMER'S GUIDE	
7. Author(s) Clyde E. Lee, Thomas W. Rioux, Vivek S. Savur, and Charlie R. Copeland	8. Performing Organization Report No. Research Report 184-2
9. Performing Organization Name and Address Center for Highway Research The University of Texas at Austin Austin, Texas 78712	10. Work Unit No. 11. Contract or Grant No. Research Study 3-18-72-184
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P. O. Box 5051 Austin, Texas 78763	13. Type of Report and Period Covered Interim 14. Sponsoring Agency Code

15. Supplementary Notes

Study conducted in cooperation with the U. S. Department of Transportation, Federal Highway Administration. Research Study Title: "Simulation of Traffic by a Step-Through Technique (Applications)"

16. Abstract

The TEXAS Model for Intersection Traffic is a new microscopic traffic simulation package which can be used as a tool in evaluating the operational effects of various traffic demands, types of traffic control, and/or geometric configurations at isolated intersections.

This report is a complement to Research Report No. 184-1, "The TEXAS Model for Intersection Traffic - Development," and provides detailed documentation on each of the programs which make up the TEXAS Model. The package consists of a geometry processor, GEOPRO, a driver-vehicle processor, DVPRO, a traffic simulation processor, SIMPRO, and an auxiliary headway distribution analysis processor, DISFIT.

Each routine, common block, and variable is defined and cross referenced according to where each is used. Limitations and error handling are documented for each processor. Numerous comments within each program listing provide an explanation of the logic or algorithms which are implemented.

17. Key Words	18. Distribution Statement
model, intersection traffic, simulation, programmer's guide, microscopic, traffic control, processor	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	346	

Form DOT F 1700.7 (8-69)

This page intentionally left blank to facilitate printing on 2 sides.

THE TEXAS MODEL FOR INTERSECTION

TRAFFIC - PROGRAMMER'S GUIDE

by

Clyde E. Lee, Thomas W. Rioux,
Vivek S. Savur, and Charlie R. Copeland

Research Report Number 184-2

Simulation of Traffic by a
Step-Through Technique (Applications)

Research Project 3-18-72-184

conducted for

Texas
State Department of Highways and Public Transportation

in cooperation with the
U. S. Department of Transportation
Federal Highway Administration

by the

CENTER FOR HIGHWAY RESEARCH

THE UNIVERSITY OF TEXAS AT AUSTIN

December 1977

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

PREFACE

This is the second in a series of four reports on Research Study Number 3-18-72-184, "Simulation of Traffic by a Step-Through Technique." This report is an appendage to Research Report No. 184-1 and describes in detail the computer programs that make up the traffic simulation package known as the TEXAS Model for Intersection Traffic.

The traffic simulation package consists of a geometry processor, GEOPRO, a driver-vehicle processor, DVPRO, a traffic simulation processor, SIMPRO, and an auxiliary headway distribution fitting processor, DISFIT. A listing of each program is provided in this report, along with the programmer's documentation.

The programmer's documentation provides where applicable: (1) program limitations; (2) an explanation of input and/or execution errors; (3) definitions of attributes (variables or arrays) in each entity (common block) and where these efficient storage and logic processing methods are used; (4) definitions of variables in each common block and the routines in which each common block is used; (5) definitions of local variables in each subroutine, the routines which can call them, and the routines they call; (6) an alphabetical listing of all routines and the routines which can call them; (7) an alphabetical listing of all variables, their storage type, and the routines in which they are used; and (8) a generalized calling sequence diagram.

Numerous comments within each program provide an explanation of the algorithms or logic which was implemented. Output from COLEASE (a special storage management and logic processor) provides insight into the data structure and its efficiency.

The four reports which deal with the development, use, and application of the TEXAS Model are

Research Report No. 184-1, "The TEXAS Model for Intersection Traffic - Development," Clyde E. Lee, Thomas W. Rioux, and Charlie R. Copeland.

Research Report No. 184-2, "The TEXAS Model for Intersection Traffic - Programmer's Guide," Clyde E. Lee, Thomas W. Rioux, Vivek S. Savur, and Charlie R. Copeland.

Research Report No. 184-3, "The TEXAS Model for Intersection Traffic - User's Guide," Clyde E. Lee, Glenn E. Grayson, Charlie R. Copeland, Jeff W. Miller, Thomas W. Rioux, and Vivek S. Savur.

Research Report No. 184-4, "The TEXAS Model for Intersection Traffic - Analysis of Signal Warrants and Intersection Capacity," Clyde E. Lee, Vivek S. Savur, and Glenn E. Grayson.

ABSTRACT

The TEXAS Model for Intersection Traffic is a new microscopic traffic simulation package which can be used as a tool in evaluating the operational effects of various traffic demands, types of traffic control, and/or geometric configurations at isolated intersections.

This report is a complement to Research Report No. 184-1, "The TEXAS Model for Intersection Traffic - Development," and provides detailed documentation on each of the programs which make up the TEXAS Model. The package consists of a geometry processor, GEOPRO, a driver-vehicle processor, DVPRO, a traffic simulation processor, SIMPRO, and an auxiliary headway distribution analysis processor, DISFIT.

Each routine, common block, and variable is defined and cross referenced according to where each is used. Limitations and error handling are documented for each processor. Numerous comments within each program listing provide an explanation of the logic or algorithms which are implemented.

This page intentionally left blank to facilitate printing on 2 sides.

SUMMARY

This report documents extensively the processors of the traffic simulation package called the TEXAS Model for Intersection Traffic. The three main processors are the geometry processor, called GEOPRO, the driver-vehicle processor, called DVPRO, and the traffic simulation processor, called SIMPRO, and an auxiliary headway distribution analysis processor, called DISFIT, aids the user in selecting headway distributions to be used by DVPRO.

Each processor is listed in this report, and in each listing comprehensive comments provide explanation of the algorithms or logic. Following the listing of each processor is its programmer's documentation, which defines and locates the data storage elements and routines, describes error handling procedures, and gives programming limitations on the simulation.

Changes to the program code should be made very cautiously since parts of the code have implications that need to be considered in the logic and algorithms in many routines. Changes which seem minor may make radical changes in driver-vehicle unit response which can affect intersection performance statistics. If changes are implemented the simulation package may need to be recalibrated and/or revalidated.

This page intentionally left blank to facilitate printing on 2 sides.

IMPLEMENTATION STATEMENT

As users gain experience using the TEXAS Model for Intersection Traffic they may want to modify certain parts of the code to represent specific conditions. If changes are to be made successfully, the programmer will need to be thoroughly familiar with the development of the model as described in Research Report No. 184-1 as well as with the documentation contained in this report.

Extreme caution is advisable in altering the code because of the complexity and the dependence of routines on each other as well as on many descriptive variables located throughout each program. Variable names and definitions remain constant throughout the processors wherever feasible.

This report provides definitions, limitations, and the organization of the routines and variables for each processor. Comment statements in the program listings provide valuable assistance in understanding the logic and algorithms used in the traffic simulation.

This page intentionally left blank to facilitate printing on 2 sides.

TABLE OF CONTENTS

This page intentionally left blank to facilitate printing on 2 sides.

APPENDIX B

ADDITIONAL INFORMATION FOR THE GEOMETRY PROCESSOR

This page intentionally left blank to facilitate printing on 2 sides.

```
C IDENTIFY,GEOPRO,60,3,GEOMETRY PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKA
C   FILES,INPUT=513,OUTPUT=513,TAPE8=513,TAPE5=INPUT
C ENTITY
C   NAME,APPRO,12,***** ENTITY FOR APPROACHES *****
C   ORDINARY,IALEFT,12,IARGHT,12,NLANES,6,LLANES(6),50,IAPX,2250
C   ORDINARY,IAPY,2250,ISLIM,118,NSDR,5,ISDRN(5),30,ISDRA(5),12
C   ORDINARY,IAAZIM,360,NDEGST,45,NDEGUT,45
C   NAME,ARC,20,***** ENTITY FOR ARCS *****
C   ORDINARY,IARCX,2250,IARCY,2250,IARCAZ,360,IARCSW,720,IARCR,127
C   ORDINARY,IDUMAR,0
C   NAME,CONFLT,1000,***** ENTITY FOR INTERSECTION CONFLICTS *****
C   ORDINARY,ICONP(2),125,ICONA(2),12,ICOND(2),250,ICONAN,360
C   ORDINARY,ICONI(2),60,IDUMCO,0
C   NAME,LANE,50,***** ENTITY FOR APPROACH LANES *****
C   ORDINARY,LWID,15,NLL,50,NLR,50,ISNA,12,NPINT,7,LINTP(7),125
C   ORDINARY,LTURN,15,LGEOM(4),1000,LTYPE,2,IDX,90,IBLN,25
C   NAME,LINE,100,***** ENTITY FOR LINES *****
C   ORDINARY,ILX1,2250,ILY1,2250,ILX2,2250,ILY2,2250
C   NAME,PATH,125,***** ENTITY FOR INTERSECTION PATHS *****
C   ORDINARY,IGEOCP(60),1000,IXL(2),2250,IYL(2),2250,JXL(2),2250
C   ORDINARY,JYL(2),2250,IXA(2),4050,IYA(2),4050,LL1,250,LA1,250,LA2,250
C   ORDINARY,LL2,250,IIA,12,IIL,6,IOA,12,IOL,6,IOPT,1,ILCH,1,IBA(2),360
C   ORDINARY,IDA(2),720,IRA(2),900,IPTURN,8,LENP,250,LIBL,50,LOBL,50
C   ORDINARY,LIMP,118,NGEOCP,60
C   NAME,SDR,30,***** ENTITY FOR AVAILABLE APPROACH SIGHT DISTANCE *****
C   ORDINARY,ICANSE(40),1000
C EXECUTIVE
C   ROUTINE,READAP,APPRO           ,LANE     ,NOATTB
C   ROUTINE,READAI     ,ARC                  ,NOATTB
C   ROUTINE,READLI                      ,LINE,NOATTB
C   ROUTINE,WRITAL     ,ARC             ,LINE
C   ROUTINE,FNDXYP,APPRO
C   ROUTINE,FNDSDR,APPRO           ,LANE                   ,SDR
C   ROUTINE,WRITAP,APPRO
C   ROUTINE,DRWAPR,APPRO,ARC       ,LANE,LINE
C   ROUTINE,DRWBOX,APPRO
C   ROUTINE,DRWINT,APPRO,ARC       ,LANE,LINE
C   ROUTINE,DRWUTA,APPRO
C   ROUTINE,FNDPTH                           ,NOATTB,PATH
C   ROUTINE,ADDPTH                                  ,PATH
C   ROUTINE,DRWPTH                                  ,PATH
C   ROUTINE,CHKPTH,APPRO           ,LANE
C   ROUTINE,WRITLA                 ,LANE                   ,SDR
C   ROUTINE,FNDCON                                  ,PATH
C   ROUTINE,CLTOLC                                  ,PATH
C   ROUTINE,ADDCON           ,CONFLT
C   ROUTINE,CLTOAC                                  ,PATH
C   ROUTINE,ADDLA                                   ,PATH
C   ROUTINE,CATOLC                                  ,PATH
C   ROUTINE,ADDAL                                   ,PATH
C   ROUTINE,CATOAC                                  ,PATH
C   ROUTINE,ADDAA                                   ,PATH
C   ROUTINE,SRTCON           ,CONFLT                ,PATH
C   ROUTINE,WRITPA                                  ,PATH
C   ROUTINE,NDXCON           ,CONFLT                ,PATH
C   ROUTINE,WRITCO           ,CONFLT
C   ROUTINE,ABORTR,APPRO,ARC,CONFLT,LANE,LINE,NOATTB,PATH,SDR
C   ROUTINE,ECHO  ,APPRO,ARC,CONFLT,LANE,LINE,NOATTB,PATH,SDR
C   EXECUTE,EXEC
C TASKS
      PROGRAM    GEOPRO ( INPUT=513,OUTPUT=513,TAPE8=513,TAPE5=INPUT )   COLEASE
      COMMON / APPRO  / IALEFT(    26)                                   COLEASE
      COMMON / ARC    / IARCX (     6)                                   COLEASE
      COMMON / CONFLT / ICONP (    10)                                   COLEASE
      COMMON / LANE   / LWID  (    20)                                   COLEASE
      COMMON / LINE   / ILX1  (     4)                                   COLEASE
      COMMON / PATH   / IGEOCP(    94)                                   COLEASE
      COMMON / SDR    / ICANSE(    40)                                   COLEASE
      COMMON / ATTB   / IAT (3,  200)                                    COLEASE
      COMMON / ENTITY / IFN (9,    7)                                    COLEASE
      COMMON / STACK  / IS     ( 3391)                                   COLEASE
      DO 1010  I = 1 ,  200                                              COLEASE
      IALEFT(I) = 0                                                      COLEASE
      IAT(3,I) = LSHIFT(1,IAT(3,I)) - 1                                  COLEASE
      IAT(3,I) = LSHIFT(IAT(3,I),IAT(2,I))                               COLEASE
 1010 CONTINUE                                                           COLEASE
      DO 1030  I = 1 , 3391                                              COLEASE
      IS(I) = 0                                                          COLEASE
 1030 CONTINUE                                                           COLEASE
      CALL  EXEC                                                         COLEASE
      CALL  EXIT                                                         COLEASE
      STOP                                                               COLEASE
      END                                                                COLEASE
```

```
      BLOCK DATA                                                      COLEASE
      COMMON / ATTB   / IAT1(300),IAT2(300)                           COLEASE
      COMMON / ENTITY / IEN (9,     7)                                COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                  COLEASE
      COMMON / NOATTB / NOATTB( 7)                                    COLEASE
      DATA      IAT1    /  0, 0, 4, 0, 4, 4, 0, 8, 3, 0,11, 6, 0,17, 6,   COLEASE
     *                     0,23, 6, 0,29, 6, 0,35, 6, 0,41, 6, 0,47,12,   COLEASE
     *                     1, 0,12, 1,12, 7, 1,19, 3, 1,22, 5, 1,27, 5,   COLEASE
     *                     1,32, 5, 1,37, 5, 1,42, 5, 1,47, 4, 1,51, 4,   COLEASE
     *                     1,55, 4, 2, 0, 4, 2, 4, 4, 2, 8, 9, 2,17, 6,   COLEASE
     *                     2,23, 6, 0, 0,12, 0,12,12, 0,24, 9, 0,33,10,   COLEASE
     *                     0,43, 7, 0,50, 0, 0, 0, 7, 0, 7, 7, 0,14, 4,   COLEASE
     *                     0,18, 4, 0,22, 8, 0,30, 8, 0,38, 9, 0,47, 6,   COLEASE
     *                     0,53, 6, 0,59, 0, 0, 0, 4, 0, 4, 6, 0,10, 6,   COLEASE
     *                     0,16, 4, 0,20, 3, 0,23, 7, 0,30, 7, 0,37, 7,   COLEASE
     *                     0,44, 7, 0,51, 7, 1, 0, 7, 1, 7, 7, 1,14, 4,   COLEASE
     *                     1,18,10, 1,28,10, 1,38,10, 1,48,10, 1,58, 2,   COLEASE
     *                     2, 0, 7, 2, 7, 5, 0, 0,12, 0,12,12, 0,24,12,   COLEASE
     *                     0,36,12, 0, 0,10, 0,10,10, 0,20,10, 0,30,10,   COLEASE
     *                     0,40,10, 0,50,10, 1, 0,10, 1,10,10, 1,20,10,   COLEASE
     *                     1,30,10, 1,40,10, 1,50,10, 2, 0,10, 2,10,10,   COLEASE
     *                     2,20,10, 2,30,10, 2,40,10, 2,50,10, 3, 0,10,   COLEASE
     *                     3,10,10, 3,20,10, 3,30,10, 3,40,10, 3,50,10,   COLEASE
     *                     4, 0,10, 4,10,10, 4,20,10, 4,30,10, 4,40,10,   COLEASE
     *                     4,50,10, 5, 0,10, 5,10,10, 5,20,10, 5,30,10/   COLEASE
      DATA      IAT2    /  5,40,10, 5,50,10, 6, 0,10, 6,10,10, 6,20,10,   COLEASE
     *                     6,30,10, 6,40,10, 6,50,10, 7, 0,10, 7,10,10,   COLEASE
     *                     7,20,10, 7,30,10, 7,40,10, 7,50,10, 8, 0,10,   COLEASE
     *                     8,10,10, 8,20,10, 8,30,10, 8,40,10, 8,50,10,   COLEASE
     *                     9, 0,10, 9,10,10, 9,20,10, 9,30,10, 9,40,10,   COLEASE
     *                     9,50,10,10, 0,12,10,12,12,10,24,12,10,36,12,   COLEASE
     *                    10,48,12,11, 0,12,11,12,12,11,24,12,11,36,12,   COLEASE
     *                    11,48,12,12, 0,12,12,12,12,12,24, 8,12,32, 8,   COLEASE
     *                    12,40, 8,12,48, 8,12,56, 4,13, 0, 3,13, 3, 4,   COLEASE
     *                    13, 7, 3,13,10, 1,13,11, 1,13,12, 9,13,21, 9,   COLEASE
     *                    13,30,10,13,40,10,13,50,10,14, 0,10,14,10, 4,   COLEASE
     *                    14,14, 8,14,22, 6,14,28, 6,14,34, 7,14,41, 6,   COLEASE
     *                     0, 0,10, 0,10,10, 0,20,10, 0,30,10, 0,40,10,   COLEASE
     *                     0,50,10, 1, 0,10, 1,10,10, 1,20,10, 1,30,10,   COLEASE
     *                     1,40,10, 1,50,10, 2, 0,10, 2,10,10, 2,20,10,   COLEASE
     *                     2,30,10, 2,40,10, 2,50,10, 3, 0,10, 3,10,10,   COLEASE
     *                     3,20,10, 3,30,10, 3,40,10, 3,50,10, 4, 0,10,   COLEASE
     *                     4,10,10, 4,20,10, 4,30,10, 4,40,10, 4,50,10,   COLEASE
     *                     5, 0,10, 5,10,10, 5,20,10, 5,30,10, 5,40,10,   COLEASE
     *                     5,50,10, 6, 0,10, 6,10,10, 6,20,10, 6,30,10/   COLEASE
      DATA      IEN     /   12,  26,   3,    1,   0,   0,   0,   0,   1,  COLEASE
     *                      20,   6,   1,   37,   0,   0,   0,   0,  27,  COLEASE
     *                    1000,  10,   1,   57,   0,   0,   0,   0,  33,  COLEASE
     *                      50,  20,   3, 1057,   0,   0,   0,   0,  43,  COLEASE
     *                     100,   4,   1, 1207,   0,   0,   0,   0,  63,  COLEASE
     *                     125,  94,  15, 1307,   0,   0,   0,   0,  67,  COLEASE
     *                      30,  40,   7, 3182,   0,   0,   0,   0, 161/  COLEASE
      DATA      LTRUE   / 1 /                                         COLEASE
      DATA      LFALSE  / 2 /                                         COLEASE
      DATA      NOATTB  /  26,  6, 10, 20,  4, 94, 40 /               COLEASE
C
C-----USER DEFINED BLOCK DATA
C
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / GEOCP  / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MRA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / ZTEMPD(105)
      DATA      D0P0    / 0.0D+00 /
      DATA      LINES   / 61 /
      DATA      MAXXA   /    0 /
      DATA      MAXXI   /    0 /
      DATA      MAXYA   /    0 /
      DATA      MAXYI   /    0 /
      DATA      MINXA   / 2250 /
      DATA      MINXI   / 2250 /
      DATA      MINYA   / 2250 /
      DATA      MINYI   / 2250 /
      DATA      MODELT  / 8 /
      DATA      NCONFS  / 0 /
      DATA      NIBL    / 0 /
      DATA      NOBL    / 0 /
      DATA      NPAGE   / 1 /
      DATA      NPATHS  / 0 /
      DATA      NSDRS   / 0 /
      DATA      NTABL   / 1 /
      DATA      XROUND  / 0.500001D+00 /
      DATA      ZERO    / 0.000001D+00 /
      END                                                             BLOCK D
```

```
      SUBROUTINE EXEC
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
C=    DIMENSION         MSG(6)
CA    DIMENSION         IBUF(513),IFET(8),MSGERR(2)
C=    DATA      MSG    / 4H FAT,4HAL E,4HXECU,4HTION,4H ERR,4HOR   /
CA    DATA      MSGERR / 13L ISLCPF ERROR /
C
C-----SUBROUTINE EXEC CONTROLS THE CALLING OF THE OTHER SUBROUTINES
C-----TO PROCESS THE INTERSECTION
C
C-----CA = TEKTRONIX PLOT
C-----C= = CDC ONLY CODE
C-----C; = IBM ONLY CODE
C
CA    IRET = ISLCPF( 7LPLTFILE,7LPLTFILE,IFET,8,IBUF,513 )
CA                      IF ( IRET . NE . 0 )          CALL ABORT ( MSGERR )
C-----READ INPUT DATA AND CHECK FOR ERRORS
      CALL  READIN
C=    ASSIGN 101 TO NRECAD
C=    CALL  XMIT     ( NRECAD )
C-----WRITE THE TITLE FOR GEOPRO, THE ARC INFORMATION, AND THE LINE
C-----INFORMATION ONTO TAPE MODELT FOR SIMPRO
      CALL  WRITAL
C-----FIND THE X AND Y COORDINATES FOR A POINT AT THE MIDDLE AND END
C-----OF EACH INBOUND LANE AND AT THE MIDDLE AND START OF EACH OUTBOUND
C-----LANE THAT IS AVAILABLE AT THE INTERSECTION, FIND THE BOUNDARIES
C-----FOR PLOTTING, AND FIND THE PLOT SCALE FACTORS
      CALL  FNDXYP
C-----FIND THE SIGHT DISTANCE RESTRICTIONS BETWEEN THE INBOUND
C-----APPROACHES
      CALL  FNDSDR
C-----WRITE THE APPROACH INFORMATION ONTO TAPE MODELT FOR SIMPRO
      CALL  WRITAP
C-----INITIALIZE PLOTTING
      CALL  INIPLT
C-----FIND THE INTERSECTION PATHS WITHIN THE INTERSECTION
      CALL  FNDPTH
C-----CHECK EACH INBOUND LANE THAT IS AVAILABLE AT THE INTERSECTION TO
C-----SEE IF AN INTERSECTION PATH WAS CALCULATED FOR EACH TURNING
C-----MOVEMENT SPECIFIED FOR THE INBOUND LANE
      CALL  CHKPTH
C-----WRITE THE LANE INFORMATION AND THE SIGHT DISTANCE RESTRICTION
C-----INFORMATION ONTO TAPE MODELT FOR SIMPRO
      CALL  WRITLA
C-----FIND THE INTERSECTION CONFLICTS BETWEEN THE INTERSECTION PATHS
      CALL  FNDCON
C-----SORT THE INTERSECTION CONFLICTS FOR EACH INTERSECTION PATH BY THE
C-----DISTANCE DOWN THE INTERSECTION PATH TO THE INTERSECTION CONFLICT
      CALL  SRTCON
C-----WRITE THE INTERSECTION PATH INFORMATION ONTO TAPE MODELT FOR
C-----SIMPRO
      CALL  WRITPA
C-----CROSS INDEX THE INTERSECTION CONFLICTS WITH THE INTERSECTION PATHS
      CALL  NDXCON
C-----WRITE THE CONFLICT INFORMATION ONTO TAPE MODELT FOR SIMPRO
      CALL  WRITCO
C-----FINISH PROCESSING
      ENDFILE MODELT
                        IF ( IPLOT . EQ . 3 )          RETURN
C=    CALL  ENDPLT
C;    CALL  PLOT     ( 0.0,0.0,999 )
      RETURN
C=101 CONTINUE
C=    CALL  ABORTR   ( MSG,22 )
C=    STOP
C=102 GO TO NRECAD
      END
                                                                  *DEBUG*
                                                                  EXEC
```

```
      SUBROUTINE READIN
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / ZTEMPD(105)
  501 FORMAT(20A4)
C
C-----SUBROUTINE READIN READS INPUT DATA AND CHECKS FOR ERRORS
C
      PI = 4.0D+00*DATAN(1.0D+00)
      RADIAN = PI/180.0D+00
      FPSMPH = 88.0D+00/60.0D+00
C-----READ 80 CHARACTER TITLE FOR GEOPRO
      READ 501 , ITITLE
      CALL  HEADER
C-----READ THE NUMBER AND LIST OF INBOUND AND OUTBOUND APPROACHES AND
C-----CHECK FOR ERRORS
      CALL  READIO
C-----READ THE APPROACH INFORMATION AND CHECK FOR ERRORS
      CALL  READAP
C-----FIND THE APPROACH TO THE LEFT AND THE APPROACH TO THE RIGHT FOR
C-----EACH INBOUND APPROACH
      CALL  APPLAR   ( NIBA,LIBA )
C-----FIND THE APPROACH TO THE LEFT AND THE APPROACH TO THE RIGHT FOR
C-----EACH OUTBOUND APPROACH
      CALL  APPLAR   ( NOBA,LOBA )
C-----READ THE ARC INFORMATION AND CHECK FOR ERRORS
      CALL  READAI
C-----READ THE LINE INFORMATION AND CHECK FOR ERRORS
      CALL  READLI
C-----READ SIGHT DISTANCE RESTRICTION COORDINATE INFORMATION AND
C-----CHECK FOR ERRORS
      CALL  READSI
C-----READ THE GEOMETRY PROCESSOR OPTIONS AND CHECK FOR ERRORS
      CALL  READOP
      RETURN
      END                                                         READIN
```

```
      SUBROUTINE HEADER
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / TITLE  / ITITLE(20)
  601 FORMAT(1H1,10X,47HGEOMETRY PROCESSOR FOR THE TEXAS TRAFFIC SIMULA,
     *        12HTION PACKAGE,4X,4HPAGE,I3,/)
  602 FORMAT(1X,20A4,//)
C
C-----SUBROUTINE HEADER SKIPS TO THE TOP OF A NEW PAGE, PRINTS THE
C-----HEADER MESSAGE, AND PRINTS THE TITLE FOR GEOPRO
C
      PRINT 601 , NPAGE
      NLINE = 2
      NPAGE = NPAGE + 1
      PRINT 602 , ITITLE
      NLINE = NLINE + 3
      RETURN
      END                                                         HEADER
```

```
      SUBROUTINE READIO
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / IANP1,NTEST,ZTEMPD(103)
  501 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,33H  -  LISTING OF INBOUND APPROACH ,
     *        7HNUMBERS,//)
  602 FORMAT(16X,I6)
  603 FORMAT(//,12X,37HTOTAL NUMBER OF INBOUND APPROACHES = ,I2,///)
  604 FORMAT(8X,5HTABLE,I3,34H  -  LISTING OF OUTBOUND APPROACH ,
     *        7HNUMBERS,//)
  605 FORMAT(16X,I6)
  606 FORMAT(//,12X,38HTOTAL NUMBER OF OUTBOUND APPROACHES = ,I2)
  607 FORMAT(///,12X,47HTOTAL NUMBER OF INBOUND AND OUTBOUND APPROACHES,
     *        3H = ,I2,///)
  801 FORMAT(32H0NUMBER OF INBOUND APPROACHES = ,I3,16H IS LE 0 OR GT 6)
  802 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,17H IS LE 0 OR GT 12)
  803 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO INBOUND ,
     *        8HAPPROACH,I3,3H = ,I3)
  804 FORMAT(32H0NUMBER OF OUTBOUND APPROACHES =,I3,16H IS LE 0 OR GT 6)
  805 FORMAT(18H0OUTBOUND APPROACH,I3,3H = ,I3,17H IS LE 0 OR GT 12)
  806 FORMAT(18H0OUTBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO OUTBOUND,
     *        9H APPROACH,I3,3H = ,I3)
  807 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO OUTBOUND,
     *        9H APPROACH,I3,3H = ,I3)
  808 FORMAT(24H0NUMBER OF APPROACHES = ,I3,17H IS LT 2 OR GT 12)
  809 FORMAT(53H0NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND,
     *        14H APPROACHES = ,I3,30H IS NE NUMBER OF APPROACHES = ,I3)
C
C-----SUBROUTINE READIO READS THE NUMBER AND LIST OF INBOUND AND
C-----OUTBOUND APPROACHES AND CHECK FOR ERRORS
C
C-----READ NUMBER OF INBOUND APPROACHES
      READ 501 , NIBA
                      IF ( NIBA . LE .  0 )          GO TO 8010
                      IF ( NIBA . GT .  6 )          GO TO 8010
            IF ( NLINE+NIBA+9 . GT . LINES )         CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
C-----READ LIST OF INBOUND APPROACHES
      READ 501 , (LIBA(IAN),IAN=1,NIBA)
      PRINT 602 , (LIBA(IAN),IAN=1,NIBA)
      NLINE = NLINE + NIBA
      DO 1020  IAN = 1 , NIBA
                      IF ( LIBA(IAN) . LE .  0 )     GO TO 8020
                      IF ( LIBA(IAN) . GT . 12 )     GO TO 8020
                      IF ( NIBA . EQ . 1 )           GO TO 1020
                      IF ( IAN . EQ . NIBA )         GO TO 1020
C-----CHECK IF APPROACH IS DUPLICATED ON LIST OF INBOUND APPROACHES
      IANP1 = IAN + 1
      DO 1010  JAN = IANP1 , NIBA
                      IF ( LIBA(IAN).EQ.LIBA(JAN) )GO TO 8030
 1010 CONTINUE
 1020 CONTINUE
      PRINT 603 , NIBA
      NLINE = NLINE + 6
C-----READ NUMBER OF OUTBOUND APPROACHES
      READ 501 , NOBA
                      IF ( NOBA . LE .  0 )          GO TO 8040
                      IF ( NOBA . GT .  6 )          GO TO 8040
            IF ( NLINE+NOBA+13 . GT . LINES )        CALL  HEADER
      PRINT 604 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
C-----READ LIST OF OUTBOUND APPROACHES
      READ 501 , (LOBA(IAN),IAN=1,NOBA)
      PRINT 605 , (LOBA(IAN),IAN=1,NOBA)
      NLINE = NLINE + NOBA
```

```
      DO 1040  IAN = 1 , NOBA
                    IF ( LOBA(IAN) , LE ,  0 )    GO TO 8050
                    IF ( LOBA(IAN) , GT , 12 )    GO TO 8050
                    IF ( NOBA , EQ , 1 )          GO TO 1040
                    IF ( IAN , EQ , NOBA )        GO TO 1040
C-----CHECK IF APPROACH IS DUPLICATED ON LIST OF OUTBOUND APPROACHES
      IANP1 = IAN + 1
      DO 1030  JAN = IANP1 , NOBA
                    IF ( LOBA(IAN).EQ.LOBA(JAN) )GO TO 8060
 1030 CONTINUE
 1040 CONTINUE
      PRINT 606 , NOBA
      NLINE = NLINE + 3
C-----CHECK IF APPROACH NUMBER IS ON LIST OF INBOUND APPROACHES AND
C-----ALSO ON LIST OF OUTBOUND APPROACHES
      DO 1060  IAN = 1 , NIBA
      DO 1050  JAN = 1 , NOBA
                    IF ( LIBA(IAN).EQ.LOBA(JAN) )GO TO 8070
 1050 CONTINUE
 1060 CONTINUE
C-----READ NUMBER OF APPROACHES
      READ 501 , NAP
                    IF ( NAP , LT ,  2 )          GO TO 8080
                    IF ( NAP , GT , 12 )          GO TO 8080
      NTEST = NIBA + NOBA
                    IF ( NTEST , NE , NAP )       GO TO 8090
      PRINT 607 , NAP
      NLINE = NLINE + 7
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8010 CONTINUE
      PRINT 801 , NIBA
      STOP  801
 8020 CONTINUE
      PRINT 802 , IAN,LIBA(IAN)
      STOP  802
 8030 CONTINUE
      PRINT 803 , IAN,LIBA(IAN),JAN,LIBA(JAN)
      STOP  803
 8040 CONTINUE
      PRINT 804 , NOBA
      STOP  804
 8050 CONTINUE
      PRINT 805 , IAN,LOBA(IAN)
      STOP  805
 8060 CONTINUE
      PRINT 806 , IAN,LOBA(IAN),JAN,LOBA(JAN)
      STOP  806
 8070 CONTINUE
      PRINT 807 , IAN,LIBA(IAN),JAN,LOBA(JAN)
      STOP  807
 8080 CONTINUE
      PRINT 808 , NAP
      STOP  808
 8090 CONTINUE
      PRINT 809 , NTEST,NAP
      STOP  809
      END                                                               READIO
```

```
      SUBROUTINE READAP                                                 COLEASE
C   TASK,READAP
      COMMON / APPRO  / IALEFT     ,IARGHT     ,NLANES     ,LLANES( 6), COLEASE
     *                  IAPX       ,IAPY       ,ISLIM      ,NSDR      , COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM       ,NDEGST    , COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / LANE   / LWID       ,NLL        ,NLR        ,ISNA      , COLEASE
     *                  NPINT      ,LINTP ( 7),LTURN       ,LGEOM ( 4), COLEASE
     *                  LTYPE      ,IDX        ,IBLN                    COLEASE
      COMMON / NOATTB / NOATTB( 7)                                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / I,ILT,IRT,IST,ITEST,IUSED(12),IUT,IYES,IZ,JBLN,
     *                  LGEOM1,LLTYPE,LTEST,NEXTL(9),NUM,ZTEMPD(71)
      DIMENSION         IENT1(1),IENT4(1)
      EQUIVALENCE       (IALEFT,IENT1(1)),(LWID,IENT4(1))
      DATA      NBLANK / 4H     /
      DATA      NL     / 1HL    /
      DATA      NR     / 1HR    /
      DATA      NS     / 1HS    /
      DATA      NU     / 1HU    /
      DATA      NYES   / 3HYES  /
  501 FORMAT(6I4,2I3,1X,A4,42X,A3)
  502 FORMAT(20A4)
  503 FORMAT(5I4,1X,4A1,15X,5I4,1X,4A1)
  601 FORMAT(8X,5HTABLE,I3,26H  -  LISTING OF APPROACHES,//)
  602 FORMAT(12X,35HAPPROACH NUMBER -------------------,I5,/,
     *       12X,35HAPPROACH AZIMUTH ------------------,I5,/,
     *       12X,35HBEGINNING CENTERLINE X COORDINATE -,I5,/,
     *       12X,35HBEGINNING CENTERLINE Y COORDINATE -,I5,/,
     *       12X,35HSPEED LIMIT (MPH) -----------------,I5,/,
     *       12X,35HNUMBER OF DEGREES FOR STRAIGHT ----,I5,/,
     *       12X,35HNUMBER OF DEGREES FOR U-TURN ------,I5,/,
     *       12X,35HNUMBER OF LANES -------------------,I5,//,
     *       12X,50HLANE IL IBLN WIDTH ---LANE GEOMETRY--- LEGAL TURNS)
  603 FORMAT(12X,I3,2I4,I5,2X,4I5,4H   (,4A1,1H))
  604 FORMAT(1H+,65X,13H(MEDIAN LANE))
  605 FORMAT(1H+,65X,11H(CURB LANE))
  606 FORMAT(/)
  607 FORMAT(12X,29HTOTAL NUMBER OF APPROACHES = ,I2,///)
  810 FORMAT(16H0APPROACH NUMBER,I3,17H IS LE 0 OR GT 12)
  811 FORMAT(16H0APPROACH NUMBER,I3,23H IS USED MORE THAN ONCE)
  812 FORMAT(16H0APPROACH NUMBER,I3,10H AZIMUTH =,I4,15H IS LT 0 OR GE ,
     *       3H360)
  813 FORMAT(16H0APPROACH NUMBER,I3,15H X COORDINATE =,I5,9H IS LT 0 ,
     *       10HOR GT 2250)
  814 FORMAT(16H0APPROACH NUMBER,I3,15H Y COORDINATE =,I5,9H IS LT 0 ,
     *       10HOR GT 2250)
  815 FORMAT(16H0APPROACH NUMBER,I3,14H SPEED LIMIT =,I3,9H IS LT 10,
     *       9H OR GT 80)
  816 FORMAT(16H0APPROACH NUMBER,I3,18H NUMBER OF LANES =,I2,6H IS LE,
     *       10H 0 OR GT 6)
  817 FORMAT(16H0APPROACH NUMBER,I3,30H NUMBER OF DEGREES FOR STRAIGH,
     *       4HT = ,I3,17H IS LT 0 OR GT 45)
  818 FORMAT(16H0APPROACH NUMBER,I3,30H NUMBER OF DEGREES FOR U-TURN ,
     *       2H= ,I3,17H IS LT 0 OR GT 45)
  819 FORMAT(16H0APPROACH NUMBER,I3,30H IS NOT ON INBOUND OR OUTBOUND,
     *       6H LISTS)
  820 FORMAT(16H0APPROACH NUMBER,I3,32H IS ON INBOUND LIST YET HAS OUTB,
     *       19HOUND DATA SPECIFIED)
  821 FORMAT(27H0NUMBER OF INBOUND LANES = ,I3,9H IS GT 25)
  822 FORMAT(16H0APPROACH NUMBER,I3,32H IS ON OUTBOUND LIST YET HAS INB,
     *       19HOUND DATA SPECIFIED)
  823 FORMAT(28H0NUMBER OF OUTBOUND LANES = ,I3,9H IS GT 25)
  824 FORMAT(16H0APPROACH NUMBER,I3,32H IS OUTBOUND YET HAS DATA FOR PE,
     *       53HRCENT OF EACH VEHICLE CLASS MAKING THE TRAFFIC STREAM)
  825 FORMAT(12H0LANE NUMBER,I3,13H LANE WIDTH =,I3,14H IS LT 8 OR GT,
```

```
     *         3H 15)
  826 FORMAT(12H0LANE NUMBER,I3,14H LANE GEOMETRY,I3,2H =,I5,6H IS LT,
     *         13H 0 OR GT 1000)
  827 FORMAT(12H0LANE NUMBER,I3,30H LANE GEOMETRY ORDER INCORRECT)
  828 FORMAT(12H0LANE NUMBER,I3,18H LANE GEOMETRY 1 =,I5,11H IS NE LANE,
     *         27H GEOMETRY 1 FOR LAST LANE =,I5)
  829 FORMAT(12H0LANE NUMBER,I3,14H TURN CODE = (,A1,12H) IS NOT ( ),
     *         7H OR (U))
  830 FORMAT(12H0LANE NUMBER,I3,14H TURN CODE = (,A1,12H) IS NOT ( ),
     *         7H OR (L))
  831 FORMAT(12H0LANE NUMBER,I3,14H TURN CODE = (,A1,12H) IS NOT ( ),
     *         7H OR (S))
  832 FORMAT(12H0LANE NUMBER,I3,14H TURN CODE = (,A1,12H) IS NOT ( ),
     *         7H OR (R))
  833 FORMAT(12H0LANE NUMBER,I3,23H NO TURN CODE SPECIFIED)
  834 FORMAT(25H0INFORMATION FOR APPROACH,I3,17H IS NOT SPECIFIED)
C
C-----SUBROUTINE READAP READS THE APPROACH INFORMATION AND CHECKS FOR
C-----ERRORS
C
                  IF ( NLINE+21 , GT , LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
      IL = 0
      JBLN = 0
      DO 1010  IZ = 1 , 12
      IUSED(IZ) = 0
 1010 CONTINUE
C-----READ INFORMATION FOR EACH APPROACH
      DO 2090  I = 1 , NAP
      NUM = NOATTB(1)
      DO 1020  IZ = 1 , NUM
      IENT1(IZ) = 0
 1020 CONTINUE
C-----READ APPROACH INFORMATION
      READ 501 , IA,IAAZIM,IAPX,IAPY,ISLIM,NLANES,NDEGST,NDEGUT,ITEST,
     *           IYES
                  IF ( NDEGST , EQ , 0 )        NDEGST = 20
                  IF ( NDEGUT , EQ , 0 )        NDEGUT = 10
      LTEST = NLINE + NLANES + 12
                  IF ( I , EQ , NAP )           LTEST = LTEST + 4
                  IF ( LTEST , GT , LINES )     CALL HEADER
      PRINT 602 , IA,IAAZIM,IAPX,IAPY,ISLIM,NDEGST,NDEGUT,NLANES
      NLINE = NLINE + 10
                  IF ( IA , LE ,  0 )           GO TO 8100
                  IF ( IA , GT , 12 )           GO TO 8100
                  IF ( IUSED(IA) , NE , 0 )     GO TO 8110
                  IF ( IAAZIM , LT ,     0 )    GO TO 8120
                  IF ( IAAZIM , GE ,   360 )    GO TO 8120
                  IF ( IAPX   , LT ,     0 )    GO TO 8130
                  IF ( IAPX   , GT ,  2250 )    GO TO 8130
                  IF ( IAPY   , LT ,     0 )    GO TO 8140
                  IF ( IAPY   , GT ,  2250 )    GO TO 8140
                  IF ( ISLIM  , LT ,    10 )    GO TO 8150
                  IF ( ISLIM  , GT ,    80 )    GO TO 8150
                  IF ( NLANES , LE ,     0 )    GO TO 8160
                  IF ( NLANES , GT ,     6 )    GO TO 8160
                  IF ( NDEGST , LT ,     0 )    GO TO 8170
                  IF ( NDEGST , GT ,    45 )    GO TO 8170
                  IF ( NDEGUT , LT ,     0 )    GO TO 8180
                  IF ( NDEGUT , GT ,    45 )    GO TO 8180
C-----CHECK IF APPROACH IS ON LIST OF INBOUND APPROACHES
      DO 1030  IAN = 1 , NIBA
                  IF ( IA , EQ , LIBA(IAN) )    GO TO 1050
 1030 CONTINUE
C-----CHECK IF APPROACH IS ON LIST OF OUTBOUND APPROACHES
      DO 1040  IAN = 1 , NOBA
                  IF ( IA , EQ , LOBA(IAN) )    GO TO 1060
 1040 CONTINUE
      GO TO 8190
 1050 CONTINUE
C-----APPROACH IS INBOUND
                  IF ( ITEST , EQ , NBLANK )    GO TO 8200
      NIBL = NIBL + NLANES
      LLTYPE = 1
                  IF ( NIBL , GT , 25 )         GO TO 8210
      GO TO 1070
 1060 CONTINUE
C-----APPROACH IS OUTBOUND
                  IF ( ITEST , NE , NBLANK )    GO TO 8220
      NOBL = NOBL + NLANES
      LLTYPE = 2
                  IF ( NOBL , GT , 25 )         GO TO 8230
 1070 CONTINUE
      IUSED(IA) = 1
      ISLIM = ISLIM*FPSMPH + XROUND
      ILN = 1
      LGEOM1 = -1
                  IF ( IYES , NE , NYES )       GO TO 2010
                  IF ( LLTYPE , EQ , 2 )        GO TO 8240
C-----DUMMY READ PERCENT OF EACH VEHICLE CLASS MAKING UP THE TRAFFIC
C-----STREAM
      READ 502
 2010 CONTINUE
      NUM = NOATTB(4)
      DO 2020  IZ = 1 , NUM
      IENT4(IZ) = 0
 2020 CONTINUE
C-----READ LANE INFORMATION (NEXTL IS FOR SECOND LANE ON CARD)
      READ 503 , LWID,LGEOM,IUT,ILT,IST,IRT,NEXTL
 2030 CONTINUE
      IL = IL + 1
      IBLN = 0
                  IF ( LLTYPE , EQ , 2 )        GO TO 2040
      JBLN = JBLN + 1
      IBLN = JBLN
 2040 CONTINUE
      PRINT 603 , ILN,IL,IBLN,LWID,LGEOM,IUT,ILT,IST,IRT
                  IF ( ILN , EQ , 1 )           PRINT 604
            IF ( ILN,EQ,NLANES , AND , ILN,NE,1 )PRINT 605
      NLINE = NLINE + 1
                  IF ( LWID , LT ,  8 )         GO TO 8250
                  IF ( LWID , GT , 15 )         GO TO 8250
C-----CHECK LANE GEOMETRY
      DO 2050  IZ = 1 , 4
                  IF ( LGEOM(IZ) , LT ,    0 ) GO TO 8260
                  IF ( LGEOM(IZ) , GT , 1000 ) GO TO 8260
 2050 CONTINUE
                  IF ( LGEOM(1),EQ,LGEOM(3),AND,
     *                 LGEOM(2),EQ,LGEOM(4),AND,
     *                 LGEOM(2),GT,LGEOM(1) )  GO TO 2060
                  IF ( LGEOM(1),EQ,LGEOM(2),AND,
     *                 LGEOM(3),GT,LGEOM(2),AND,
     *                 LGEOM(4),GT,LGEOM(3) )  GO TO 2060
                  IF ( LGEOM(3),EQ,LGEOM(4),AND,
     *                 LGEOM(2),GT,LGEOM(1),AND,
     *                 LGEOM(3),GT,LGEOM(2) )  GO TO 2060
                  IF ( LGEOM(2),GT,LGEOM(1),AND,
     *                 LGEOM(3),GT,LGEOM(2),AND,
     *                 LGEOM(4),GT,LGEOM(3) )  GO TO 2060
      GO TO 8270
 2060 CONTINUE
      IF ( ILN,NE,1,AND,LGEOM(1),NE,LGEOM1,AND,LLTYPE,EQ,1 )
     *                                         GO TO 8280
C-----CHECK TURNING MOVEMENTS THAT ARE LEGAL
      LTURN = 0
            IF ( IUT,NE,NBLANK,AND,IUT,NE,NU )   GO TO 8290
                  IF ( IUT , EQ , NU )         LTURN = LTURN + 8
            IF ( ILT,NE,NBLANK,AND,ILT,NE,NL )   GO TO 8300
                  IF ( ILT , EQ , NL )         LTURN = LTURN + 4
            IF ( IST,NE,NBLANK,AND,IST,NE,NS )   GO TO 8310
```

```
                    IF ( IST . EQ . NS )          LTURN = LTURN + 2
            IF ( IRT.NE.NBLANK.AND.IRT.NE.NR )    GO TO 8320
                    IF ( IRT . EQ . NR )          LTURN = LTURN + 1
      IF ( LTURN.LE.0 . AND . LGEOM(3).NE.LGEOM(4) . AND . LLTYPE.EQ.1 )
     *                                            GO TO 8330
      IF ( LTURN.LE.0 . AND . LGEOM(1).NE.LGEOM(2) . AND . LLTYPE.EQ.2 )
     *                                            GO TO 8330
      LLANES(ILN) = IL
      ISNA = IA
      LTYPE = LLTYPE
C-----FIND LANE TO THE LEFT AND THE RIGHT
      NLL = IL - 1
                    IF ( ILN . EQ . 1 )           NLL = 0
      NLR = IL + 1
                    IF ( ILN . EQ . NLANES )      NLR = 0
C-----STORE LANE INFORMATION IN ENTRY IL OF ENTITY LANE
C   COLEASE,REPACK,LANE,IL
      CALL  REPACK  (      4,IL     )                                   COLEASE
      ILN = ILN + 1
                    IF ( (ILN/2)*2 . NE . ILN )   GO TO 2080
                    IF ( ILN . GT . NLANES  )     GO TO 2080
C-----PROCESS SECOND LANE ON CARD
      NUM = NOATTB(4)
      DO 2070  IZ = 1 , NUM
      IENT4(IZ) = 0
 2070 CONTINUE
      LWID = NEXTL(1)
      LGEOM1 = LGEOM(1)
      LGEOM(1) = NEXTL(2)
      LGEOM(2) = NEXTL(3)
      LGEOM(3) = NEXTL(4)
      LGEOM(4) = NEXTL(5)
      IUT = NEXTL(6)
      ILT = NEXTL(7)
      IST = NEXTL(8)
      IRT = NEXTL(9)
      GO TO 2030
 2080 CONTINUE
                    IF ( ILN . LE . NLANES )      GO TO 2010
C-----END OF LANE LOOP
      PRINT 606
      NLINE = NLINE + 2
C-----STORE APPROACH INFORMATION IN ENTRY IA OF ENTITY APPRO
C   COLEASE,REPACK,APPRO,IA
      CALL  REPACK  (      1,IA     )                                   COLEASE
C-----END OF APPROACH LOOP
 2090 CONTINUE
C-----CHECK IF INFORMATION FOR EACH INBOUND APPROACH WAS SPECIFIED
      DO 3010  IAN = 1 , NIBA
      IA = LIBA(IAN)
                    IF ( IUSED(IA) . EQ . 0 )     GO TO 8340
 3010 CONTINUE
C-----CHECK IF INFORMATION FOR EACH OUTBOUND APPROACH WAS SPECIFIED
      DO 3020  IAN = 1 , NOBA
      IA = LOBA(IAN)
                    IF ( IUSED(IA) . EQ . 0 )     GO TO 8340
 3020 CONTINUE
      PRINT 607 , NAP
      NLINE = NLINE + 4
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8100 CONTINUE
      PRINT 810 , IA
      STOP  810
 8110 CONTINUE
      PRINT 811 , IA
      STOP  811
 8120 CONTINUE
      PRINT 812 , IA,IAAZIM
      STOP  812
 8130 CONTINUE
      PRINT 813 , IA,IAPX
      STOP  813
 8140 CONTINUE
      PRINT 814 , IA,IAPY
      STOP  814
 8150 CONTINUE
      PRINT 815 , IA,ISLIM
      STOP  815
 8160 CONTINUE
      PRINT 816 , IA,NLANES
      STOP  816
 8170 CONTINUE
      PRINT 817 , IA,NDEGST
      STOP  817
 8180 CONTINUE
      PRINT 818 , IA,NDEGUT
      STOP  818
 8190 CONTINUE
      PRINT 819 , IA
      STOP  819
 8200 CONTINUE
      PRINT 820 , IA
      STOP  820
 8210 CONTINUE
      PRINT 821 , NIBL
      STOP  821
 8220 CONTINUE
      PRINT 822 , IA
      STOP  822
 8230 CONTINUE
      PRINT 823 , NOBL
      STOP  823
 8240 CONTINUE
      PRINT 824 , IA
      STOP  824
 8250 CONTINUE
      PRINT 825 , ILN,LWID
      STOP  825
 8260 CONTINUE
      PRINT 826 , ILN,IZ,LGEOM(IZ)
      STOP  826
 8270 CONTINUE
      PRINT 827 , ILN
      STOP  827
 8280 CONTINUE
      PRINT 828 , ILN,LGEOM(1),LGEOM1
      STOP  828
 8290 CONTINUE
      PRINT 829 , ILN,IUT
      STOP  829
 8300 CONTINUE
      PRINT 830 , ILN,ILT
      STOP  830
 8310 CONTINUE
      PRINT 831 , ILN,IST
      STOP  831
 8320 CONTINUE
      PRINT 832 , ILN,IRT
      STOP  832
 8330 CONTINUE
      PRINT 833 , ILN
      STOP  833
 8340 CONTINUE
      PRINT 834 , IA
      STOP  834
      END                                                               READAP
```

```
      SUBROUTINE APPLAR ( NBA,LBA )
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / ZTEMPD / IALEFT,IARGHT,IMAXAZ,IMINAZ,JAAZIM,KAAZIM,
     *                  LAAZIM,ZTEMPD(98)
      DIMENSION         LBA(1)
C
C-----SUBROUTINE APPLAR FINDS THE APPROACH TO THE LEFT AND THE APPROACH
C-----TO THE RIGHT FOR EACH APPROACH ON THE LBA LIST
C
C-----PROCESS EACH APPROACH ON THE LBA LIST
      DO 1030  IAN = 1 , NBA
      IA = LBA(IAN)
C   COLEASE,FIND,JAAZIM,APPRO,IA,IAAZIM
      CALL  FIND    (JAAZIM,      1,IA     ,     24)                    COLEASE
      IMAXAZ = 0
      IMINAZ = 360
C-----CHECK AGAINST EACH OTHER APPROACH ON THE LBA LIST
      DO 1020  JAN = 1 , NBA
                        IF ( IAN . EQ . JAN )         GO TO 1020
      JA = LBA(JAN)
C   COLEASE,FIND,KAAZIM,APPRO,JA,IAAZIM
      CALL  FIND    (KAAZIM,      1,JA     ,     24)                    COLEASE
                        IF ( KAAZIM . LT . JAAZIM )  KAAZIM = KAAZIM + 360
      LAAZIM = KAAZIM - JAAZIM
                        IF ( LAAZIM . GT . IMINAZ )  GO TO 1010
C-----APPROACH TO THE LEFT HAS THE MINIMUM AZIMUTH DIFFERENCE
      IMINAZ = LAAZIM
      IALEFT = JA
 1010 CONTINUE
                        IF ( LAAZIM . LT . IMAXAZ )  GO TO 1020
C-----APPROACH TO THE RIGHT HAS THE MAXIMUM AZIMUTH DIFFERENCE
      IMAXAZ = LAAZIM
      IARGHT = JA
C-----END OF OTHER APPROACH LOOP
 1020 CONTINUE
C-----STORE APPROACH TO THE LEFT FOR ENTRY IA OF ENTITY APPRO
C   COLEASE,STORE,IALEFT,APPRO,IA,IALEFT
      CALL  STORE   (IALEFT,      1,IA     ,      1)                    COLEASE
C-----STORE APPROACH TO THE RIGHT FOR ENTRY IA OF ENTITY APPRO
C   COLEASE,STORE,IARGHT,APPRO,IA,IARGHT
      CALL  STORE   (IARGHT,      1,IA     ,      2)                    COLEASE
C-----END OF APPROACH LOOP
 1030 CONTINUE
      RETURN
      END                                                               APPLAR
```

```
      SUBROUTINE READAI                                                 COLEASF
C   TASK,READAI
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW         COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / NOATTB / NOATTB( 7)                                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / I,IUSED(20),IZ,J,LTEST,NUM,ZTEMPD(80)
      DIMENSION         IENT2(1)
      EQUIVALENCE       (IARCX,IENT2(1))
  501 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,40H  -  LISTING OF ARCS (FOR PLOTTING ONLY),
     *       //)
  602 FORMAT(12X,35HARC NUMBER -----------------------,I5,/,
     *       12X,35HCENTER X COORDINATE --------------,I5,/,
     *       12X,35HCENTER Y COORDINATE --------------,I5,/,
     *       12X,35HBEGINNING AZIMUTH ----------------,I5,/,
     *       12X,35HSWEEP ANGLE ----------------------,I5,/,
     *       12X,35HRADIUS OF ARC --------------------,I5,/,
     *       12X,35HROTATION FROM BEGINNING AZIMUTH ---)
  603 FORMAT(1H+,47X,9HCLOCKWISE//)
  604 FORMAT(1H+,47X,17HCOUNTER CLOCKWISE//)
  605 FORMAT(12X,23HTOTAL NUMBER OF ARCS = ,I2,///)
  835 FORMAT(18H0NUMBER OF ARCS = ,I3,17H IS LT 0 OR GT 20)
  836 FORMAT(11H0ARC NUMBER,I3,3H = ,I3,17H IS LE 0 OR GT 20)
  837 FORMAT(11H0ARC NUMBER,I3,23H IS USED MORE THAN ONCE)
  838 FORMAT(11H0ARC NUMBER,I3,15H X COORDINATE =,I5,13H IS LT 0 OR G,
     *       6HT 2250)
  839 FORMAT(11H0ARC NUMBER,I3,15H Y COORDINATE =,I5,13H IS LT 0 OR G,
     *       6HT 2250)
  840 FORMAT(11H0ARC NUMBER,I3,10H AZIMUTH =,I4,18H IS LT 0 OR GE 360)
  841 FORMAT(11H0ARC NUMBER,I3,20H NUMBER OF DEGREES =,I4,8H IS LT -,
     *       13H360 OR GT 360)
  842 FORMAT(11H0ARC NUMBER,I3,9H RADIUS =,I6,18H IS LE 0 OR GT 127)
C
C-----SUBROUTINE READAI READ THE ARC INFORMATION AND CHECKS FOR ERRORS
C
C-----READ NUMBER OF ARCS
      READ 501 , NARCS
                     IF ( NARCS . LT .  0 )        GO TO 8350
                     IF ( NARCS . EQ .  0 )        GO TO 1040
                     IF ( NARCS . GT . 20 )        GO TO 8350
                     IF ( NLINE+16 . GT . LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
      DO 1010  IZ = 1 , 20
      IUSED(IZ) = 0
 1010 CONTINUE
      NUM = NOATTB(2)
C-----READ INFORMATION FOR EACH ARC
      DO 1030  I = 1, NARCS
      DO 1020  IZ = I , NUM
      IENT2(IZ) = 0
 1020 CONTINUE
C-----READ ARC INFORMATION
      READ 501 , J,IARCX,IARCY,IARCAZ,IARCSW,IARCR
      LTEST = NLINE + 9
                     IF ( I . EQ . NARCS )         LTEST = LTEST + 4
                     IF ( LTEST . GT . LINES )     CALL  HEADER
      PRINT 602 , J,IARCX,IARCY,IARCAZ,IARCSW,IARCR
                     IF ( IARCSW . GE . 0 )        PRINT 603
                     IF ( IARCSW . LT . 0 )        PRINT 604
      NLINF = NLINE + 9
                     IF ( J . LE .  0 )            GO TO 8360
                     IF ( J . GT . 20 )            GO TO 8360
                     IF ( IUSED(J) . NE . 0 )      GO TO 8370
                     IF ( IARCX  . LT .    0 )     GO TO 8380
                     IF ( IARCX  . GT . 2250 )     GO TO 8380
                     IF ( IAPCY  . LT .    0 )     GO TO 8390
```

```
                IF ( IARCY  . GT . 2250 )    GO TO 8390
                IF ( IARCAZ . LT .    0 )    GO TO 8400
                IF ( IARCAZ . GE .  360 )    GO TO 8400
                IF ( IARCSW . LT . -360 )    GO TO 8410
                IF ( IARCSW . GT . +360 )    GO TO 8410
                IF ( IARCR  . LE .    0 )    GO TO 8420
                IF ( IARCR  . GT .  127 )    GO TO 8420
      LARCS(I) = J
      IUSED(J) = 1
C-----STORE ARC INFORMATION IN ENTRY J OF ENTITY ARC
      IARCSW = IARCSW + 360
C     COLEASE,REPACK,ARC,J
      CALL  REPACK  (       2,J       )                          COLEASE
      IARCSW = IARCSW - 360
C-----END OF ARC LOOP
 1030 CONTINUE
      PRINT 605 , NARCS
      NLINE = NLINE + 4
 1040 CONTINUE
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8350 CONTINUE
      PRINT 835 , NARCS
      STOP  835
 8360 CONTINUE
      PRINT 836 , I,J
      STOP  836
 8370 CONTINUE
      PRINT 837 , J
      STOP  837
 8380 CONTINUE
      PRINT 838 , J,IARCX
      STOP  838
 8390 CONTINUE
      PRINT 839 , J,IARCY
      STOP  839
 8400 CONTINUE
      PRINT 840 , J,IARCAZ
      STOP  840
 8410 CONTINUE
      PRINT 841 , J,IARCSW
      STOP  841
 8420 CONTINUE
      PRINT 842 , J,IARCR
      STOP  842
      END
```

```
      SUBROUTINE READLI                                                  COLEASE
C   TASK,READLI
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2            COLEASE
      COMMON / NOATTB / NOATTB( 7)                                       COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / I,IUSED(100),IZ,J,LTEST,NUM
      DIMENSION         IENT5(1)
      EQUIVALENCE       (ILX1,IENT5(1))
  501 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,41H  -  LISTING OF LINES (FOR PLOTTING ONLY),
     *      //)
  602 FORMAT(12X,35HLINE NUMBER ----------------------,I5,/,
     *         12X,35HSTART X COORDINATE ---------------,I5,/,
     *         12X,35HSTART Y COORDINATE ---------------,I5,/,
     *         12X,35HEND X COORDINATE -----------------,I5,/,
     *         12X,35HEND Y COORDINATE -----------------,I5,//)
  603 FORMAT(12X,24HTOTAL NUMBER OF LINES = ,I2,///)
  843 FORMAT(19H0NUMBER OF LINES = ,I3,18H IS LT 0 OR GT 100)
  844 FORMAT(12H0LINE NUMBER,I3,3H = ,I3,18H IS LE 0 OR GT 100)
  845 FORMAT(12H0LINE NUMBER,I3,23H IS USED MORE THAN ONCE)
  846 FORMAT(12H0LINE NUMBER,I3,25H BEGINNING X COORDINATE =,I5,2H I,
     *         17HS LT 0 OR GT 2250)
  847 FORMAT(12H0LINE NUMBER,I3,25H BEGINNING Y COORDINATE =,I5,2H I,
     *         17HS LT 0 OR GT 2250)
  848 FORMAT(12H0LINE NUMBER,I3,22H ENDING X COORDINATE =,I5,6H IS LT,
     *         13H 0 OR GT 2250)
  849 FORMAT(12H0LINE NUMBER,I3,22H ENDING Y COORDINATE =,I5,6H IS LT,
     *         13H 0 OR GT 2250)
C
C-----SUBROUTINE READLI READS THE LINE INFORMATION AND CHECKS FOR ERRORS
C
C-----READ NUMBER OF LINES
      READ 501 , NLINES
                     IF ( NLINES . LT .   0 )     GO TO 8430
                     IF ( NLINES . EQ .   0 )     GO TO 1040
                     IF ( NLINES . GT . 100 )     GO TO 8430
                     IF ( NLINE+14 . GT . LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
      DO 1010  IZ = 1 , 100
      IUSED(IZ) = 0
 1010 CONTINUE
      NUM = NOATTB(5)                                                    READAI
C-----READ INFORMATION FOR EACH LINE
      DO 1030  I = 1, NLINES
      DO 1020  IZ = 1 , NUM
      IENT5(IZ) = 0
 1020 CONTINUE
C-----READ LINE INFORMATION
      READ 501 , J,ILX1,ILY1,ILX2,ILY2
      LTEST = NLINE + 7
                     IF ( I . EQ . NLINES )       LTEST = LTEST + 4
                     IF ( LTEST . GT . LINES )    CALL  HEADER
      PRINT 602 , J,ILX1,ILY1,ILX2,ILY2
      NLINE = NLINE + 7
                     IF ( J . LE .   0 )          GO TO 8440
                     IF ( J . GT . 100 )          GO TO 8440
                     IF ( IUSED(J) . NE . 0 )     GO TO 8450
                     IF ( ILX1 . LT .    0 )      GO TO 8460
                     IF ( ILX1 . GT . 2250 )      GO TO 8460
                     IF ( ILY1 . LT .    0 )      GO TO 8470
                     IF ( ILY1 . GT . 2250 )      GO TO 8470
                     IF ( ILX2 . LT .    0 )      GO TO 8480
                     IF ( ILX2 . GT . 2250 )      GO TO 8480
                     IF ( ILY2 . LT .    0 )      GO TO 8490
                     IF ( ILY2 . GT . 2250 )      GO TO 8490
      LLINES(I) = J
      IUSED(J) = 1
```

```
C=====STORE LINE INFORMATION IN ENTRY J OF ENTITY LINE
C   COLEASE,REPACK,LINE,J
      CALL  REPACK  (      5,J      )                               COLEASE
C=====END OF LINE LOOP
 1030 CONTINUE
      PRINT 603 , NLINES
      NLINE = NLINE + 4
 1040 CONTINUE
      RETURN
C=====PROCESS INPUT ERRORS AND STOP
 8430 CONTINUE
      PRINT 843 , NLINES
      STOP  843
 8440 CONTINUE
      PRINT 844 , I,J
      STOP  844
 8450 CONTINUE
      PRINT 845 , J
      STOP  845
 8460 CONTINUE
      PRINT 846 , J,ILX1
      STOP  846
 8470 CONTINUE
      PRINT 847 , J,ILY1
      STOP  847
 8480 CONTINUE
      PRINT 848 , J,ILX2
      STOP  848
 8490 CONTINUE
      PRINT 849 , J,ILY2
      STOP  849
      END                                                               READLI
```

```
      SUBROUTINE READSI
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / ZTEMPD / I,IUSED(20),IZ,J,LTEST,ZTEMPD(81)
  501 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,22H  -  LISTING OF SIGHT ,
     *        32HDISTANCE RESTRICTION COORDINATES,//)
  602 FORMAT(12X,35HSIGHT DISTANCE RESTRICTION NUMBER =,I5,/,
     *        12X,35HX COORDINATE ----------------------,I5,/,
     *        12X,35HY COORDINATE ----------------------,I5,//)
  603 FORMAT(12X,25HTOTAL NUMBER OF POINTS = ,I2,///)
  850 FORMAT(41H0NUMBER OF SIGHT DISTANCE RESTRICTIONS = ,I3,8H IS LT 0,
     *        9H OR GT 20)
  851 FORMAT(34H0SIGHT DISTANCE RESTRICTION NUMBER,I3,3H = ,I3,6H IS LE,
     *        11H 0 OR GT 20)
  852 FORMAT(34H0SIGHT DISTANCE RESTRICTION NUMBER,I3,14H IS USED MORE ,
     *        9HTHAN ONCE)
  853 FORMAT(27H0SIGHT DISTANCE RESTRICTION,I3,15H X COORDINATE =,I5,
     *        19H IS LT 0 OR GT 2250)
  854 FORMAT(27H0SIGHT DISTANCE RESTRICTION,I3,15H Y COORDINATE =,I5,
     *        19H IS LT 0 OR GT 2250)
C
C=====SUBROUTINE READSI READS THE SIGHT DISTANCE RESTRICTION
C=====COORDINATE INFORMATION AND CHECKS FOR ERRORS
C
C=====READ NUMBER OF SIGHT DISTANCE RESTRICTION COORDINATES
      READ 501 , NSDRC
                      IF ( NSDRC , LT ,  0 )         GO TO 8500
                      IF ( NSDRC , EQ ,  0 )         GO TO 1030
                      IF ( NSDRC , GT , 20 )         GO TO 8500
                      IF ( NLINE+12 , GT , LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
      DO 1010  IZ = 1 , 20
      IUSED(IZ) = 0
 1010 CONTINUE
C=====READ INFORMATION FOR SIGHT DISTANCE RESTRICTION COORDINATES
      DO 1020  I = 1 , NSDRC
C=====READ SIGHT DISTANCE RESTRICTION COORDINATE INFORMATION
      READ 501 , J,IXSDRC(J),IYSDRC(J)
      LTEST = NLINE + 5
                      IF ( I , EQ , NSDRC )          LTEST = LTEST + 4
                      IF ( LTEST , GT , LINES )      CALL  HEADER
      PRINT 602 , J,IXSDRC(J),IYSDRC(J)
      NLINE = NLINE + 5
                      IF ( J , LE ,  0 )             GO TO 8510
                      IF ( J , GT , 20 )             GO TO 8510
                      IF ( IUSED(J) , NE , 0 )       GO TO 8520
                      IF ( IXSDRC(J) , LT ,    0 ) GO TO 8530
                      IF ( IXSDRC(J) , GT , 2250 ) GO TO 8530
                      IF ( IYSDRC(J) , LT ,    0 ) GO TO 8540
                      IF ( IYSDRC(J) , GT , 2250 ) GO TO 8540
      LSDRC(I) = J
      IUSED(J) = 1
C=====END OF SIGHT DISTANCE RESTRICTION COORDINATE LOOP
 1020 CONTINUE
      PRINT 603 , NSDRC
      NLINE = NLINE + 4
 1030 CONTINUE
      RETURN
C=====PROCESS INPUT ERROR AND STOP
 8500 CONTINUE
      PRINT 850 , NSDRC
      STOP  850
 8510 CONTINUE
      PRINT 851 , I,J
      STOP  851
 8520 CONTINUE
      PRINT 852 , J
      STOP  852
```

```
8530 CONTINUE
     PRINT 853 , J,IXSDRC
     STOP  853
8540 CONTINUE
     PRINT 854 , J,IYSDRC
     STOP  854
     END
```

READSI

```
      SUBROUTINE READOP
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / JPATH(2),JPLOT(2),JSAME(2),R,SA,SI,ZTEMPD(96)
      DIMENSION         NNOPLT(2),NOPT1(2),NPLTI(2),NPRIM(2),NSEPAR(2)
      DATA      NBLANK / 4H             /
      DATA      NNOPLT / 4HNOPL,4HOT    /
      DATA      NOPT1  / 4HOPTI,4HON1   /
      DATA      NPLT   / 4HPLOT         /
      DATA      NPLTI  / 4HPLOT,4HI     /
      DATA      NPRIM  / 4HPRIM,4HARY   /
      DATA      NSAME  / 4HSAME         /
      DATA      NSEPAR / 4HSEPA,4HRATE /
  501 FORMAT(3(2A4,2X),3F10.2,2I5)
  601 FORMAT(8X,5HTABLE,I3,28H  -  LISTING OF OPTIONS AND ,
     *       15HADDITIONAL DATA,//)
  602 FORMAT(12X,A4,A3,15H PATHS SELECTED,/)
  603 FORMAT(12X,43HPLOT SELECTED USING 30 INCH PAPER AND BALL ,
     *       9HPOINT PEN,/)
  604 FORMAT(12X,45HPLOT SELECTED USING 30 INCH PAPER AND INK PEN,/)
  605 FORMAT(12X,16HNO PLOT SELECTED,/)
  606 FORMAT(12X,40HAPPROACH PATHS PLOTTED ON THE SAME FRAME,/)
  607 FORMAT(12X,41HAPPROACH PATHS PLOTTED ON SEPARATE FRAMES,/)
  608 FORMAT(12X,35HAPPROACH SCALE FACTOR FROM INPUT IS,F6.1,8H FEET PE,
     *       6HR INCH,//,12X,39HINTERSECTION SCALE FACTOR FROM INPUT IS,
     *       F6.1,14H FEET PER INCH,/)
  609 FORMAT(12X,47HA STRAIGHT LINE WILL BE USED FOR A PATH WITH A ,
     *       9HRADIUS GT,F7.2,3H FT,/)
  610 FORMAT(12X,46HPROGRAM CHECKS TO SEE IF THE CENTER TO CENTER ,
     *       8HDISTANCE,/,16X,35HBETWEEN VEHICLES BECOMES LESS THAN ,
     *       11HOR EQUAL TO,I3,5H FEET/)
  611 FORMAT(12X,19HPLOT PAPER WIDTH = ,I2,7H INCHES,/)
  855 FORMAT(16H0PATH OPTION = (,2A4,30H) IS NE (        )OR(PRIMARY ),
     *       12HOR(OPTION1 ))
  856 FORMAT(16H0PLOT OPTION = (,2A4,30H) IS NE (        )OR(PLOT    ),
     *       24HOR(PLOTI   )OR(NOPLOT  ))
  857 FORMAT(21H0PATH PLOT OPTION = (,2A4,26H) IS NE (        )OR(SAME ,
     *       16H    )OR(SEPARATE))
  858 FORMAT(18H0CLOSE DISTANCE = ,I3,17H IS LT 6 OR GT 20)
  859 FORMAT(20H0PLOT PAPER WIDTH = ,I3,15H IS NE 12 OR 30)
C
C-----SUBROUTINE READOP READS THE GEOMETRY PROCESSOR OPTIONS AND CHECKS
C-----FOR ERRORS
C
                      IF ( NLINE+7 . GT . LINES )  CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
C-----READ GEOPRO OPTIONS
      READ (5,501,END=1010)  JPATH,JPLOT,JSAME,SA,SI,R,ICLOSE,IPAPER
 1010 CONTINUE
C-----PROCESS PATH OPTION - DEFAULT IS (PRIMARY )
      IF ( JPATH(1).EQ.NBLANK    . AND . JPATH(2).EQ.NBLANK    )
     *                                                   GO TO 1020
      IF ( JPATH(1).EQ.NPRIM(1)  . AND . JPATH(2).EQ.NPRIM(2)  )
     *                                                   GO TO 1030
      IF ( JPATH(1).EQ.NOPT1(1)  . AND . JPATH(2).EQ.NOPT1(2)  )
     *                                                   GO TO 1040
      GO TO 8550
 1020 CONTINUE
      JPATH(1) = NPRIM(1)
      JPATH(2) = NPRIM(2)
 1030 CONTINUE
C-----PATH OPTION IS (PRIMARY )
      IPATH = 1
      GO TO 1050
 1040 CONTINUE
C-----PATH OPTION IS (OPTION1 )
      IPATH = 2
```

```
 1050 CONTINUE
      PRINT 602 , JPATH
      NLINE = NLINE + 2
C-----PROCESS PLOT OPTION - DEFAULT IS (PLOT     )
      IF ( JPLOT(1).EQ.NBLANK     . AND . JPLOT(2).EQ.NBLANK     )
     *                                            GO TO 2010
      IF ( JPLOT(1).EQ.NPLT       . AND . JPLOT(2).EQ.NBLANK     )
     *                                            GO TO 2010
      IF ( JPLOT(1).EQ.NPLTI(1)   . AND . JPLOT(2).EQ.NPLTI(2)   )
     *                                            GO TO 2020
      IF ( JPLOT(1).EQ.NNOPLT(1)  . AND . JPLOT(2).EQ.NNOPLT(2)  )
     *                                            GO TO 2030
      GO TO 8560
 2010 CONTINUE
C-----PLOT OPTION IS (PLOT     )
      IPLOT = 1
      PRINT 603
      GO TO 2040
 2020 CONTINUE
C-----PLOT OPTION IS (PLOTI    )
      IPLOT = 2
      PRINT 604
      GO TO 2040
 2030 CONTINUE
C-----PLOT OPTION IS (NOPLOT   )
      IPLOT = 3
      PRINT 605
 2040 CONTINUE
      NLINE = NLINE + 2
                      IF ( IPLOT . EQ . 3 )       GO TO 4010
                      IF ( NLINE+6 . GT . LINES ) CALL HEADER
C-----PROCESS PATH PLOT OPTION - DEFAULT IS (SEPARATE)
      IF ( JSAME(1).EQ.NBLANK     . AND . JSAME(2).EQ.NBLANK     )
     *                                            GO TO 3020
      IF ( JSAME(1).EQ.NSAME      . AND . JSAME(2).EQ.NBLANK     )
     *                                            GO TO 3010
      IF ( JSAME(1).EQ.NSEPAR(1)  . AND . JSAME(2).EQ.NSEPAR(2)  )
     *                                            GO TO 3020
      GO TO 8570
 3010 CONTINUE
C-----PATH PLOT OPTION IS (SAME     )
      ISAME = 1
      PRINT 606
      GO TO 3030
 3020 CONTINUE
C-----PATH PLOT OPTION IS (SEPARATE)
      ISAME = 2
      PRINT 607
 3030 CONTINUE
      NLINE = NLINE + 2
C-----PROCESS PLOT SCALE FACTOR FOR APPROACH AND INTERSECTION
      PRINT 608 , SA,SI
      NLINE = NLINE + 4
      SCALEA = DBLE(SA)
      SCALEI = DBLE(SI)
 4010 CONTINUE
C-----PROCESS MAXIMUM PATH RADIUS - DEFAULT IS 500.0
                      IF ( R . EQ .   0.0 )       R = 500.0
      R = AMIN1(AMAX1(R,100.0),900.0)
                      IF ( NLINE+2 . GT . LINES  ) CALL  HEADER
      PRINT 609 , R
      NLINE = NLINE + 2
      RADIUS = DBLE(R)
C-----PROCESS CLOSE DISTANCE - DEFAULT IS 8
                      IF ( ICLOSE . EQ .  0 )     ICLOSE = 10
                      IF ( ICLOSE . LT .  6 )     GO TO 8580
                      IF ( ICLOSE . GT . 20 )     GO TO 8580
                      IF ( NLINE+3 . GT . LINES ) CALL  HEADER
      PRINT 610 , ICLOSE
      NLINE = NLINE + 3
                      IF ( IPLOT . EQ . 3 )       GO TO 4020
                      IF ( IPAPER . EQ . 0 )      IPAPER = 30
CA    IPAPER = 12
      IF ( IPAPER.NE.12 . AND . IPAPER.NE.30 )    GO TO 8590
                      IF ( NLINE+2 . GT . LINES ) CALL  HEADER
      PRINT 611 , IPAPER
      NLINE = NLINE + 2
 4020 CONTINUE
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8550 CONTINUE
      PRINT 855 , JPATH
      STOP  855
 8560 CONTINUE
      PRINT 856 , JPLOT
      STOP  856
 8570 CONTINUE
      PRINT 857 , JSAME
      STOP  857
 8580 CONTINUE
      PRINT 858 , ICLOSE
      STOP  858
 8590 CONTINUE
      PRINT 859 , IPAPER
      STOP  859
      END                                                          READOP
```

```
      SUBROUTINE WRITAL                                                 COLEASE
C     TASK,WRITAL
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW         COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2           COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / IARC,IARCN,ILINE,ILINEN,ZTEMPD(101)
  601 FORMAT(20A4)
  602 FORMAT(20I4)
C
C-----SUBROUTINE WRITAL WRITES THE TITLE FOR GEOPRO, THE ARC
C-----INFORMATION, AND THE LINE INFORMATION ONTO TAPE MODELT FOR SIMPRO
C
      REWIND MODELT
C-----WRITE THE TITLE FOR GEOPRO ONTO MODELT
      WRITE (MODELT,601)  ITITLE
C-----WRITE THE ARC INFORMATION ONTO MODELT
      WRITE (MODELT,602)  NARCS
                      IF ( NARCS . LE . 0 )          GO TO 1020
      DO 1010  IARCN = 1 , NARCS
      IARC = LARCS(IARCN)
C     COLEASE,EXTRAC,ARC,IARC
      CALL  EXTRAC  (      2,IARC   )                                   COLEASE
      IARCSW = IARCSW - 360
      WRITE (MODELT,602)  IARC,IARCX,IARCY,IARCAZ,IARCSW,IARCR
 1010 CONTINUE
 1020 CONTINUE
C-----WRITE THE LINE INFORMATION ONTO MODELT
      WRITE (MODELT,602)  NLINES
                      IF ( NLINES . LE . 0 )         GO TO 2020
      DO 2010  ILINEN = 1 , NLINES
      ILINE = LLINES(ILINEN)
C     COLEASE,EXTRAC,LINE,ILINE
      CALL  EXTRAC  (      5,ILINE )                                    COLEASE
      WRITE (MODELT,602)  ILINE,ILX1,ILY1,ILX2,ILY2
 2010 CONTINUE
 2020 CONTINUE
      RETURN
      END                                                               WRITAL
```

```
      SUBROUTINE FNDXYP                                                 COLEASE
C     TASK,FNDXYP
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / DAI,DW,DXI,DYI,I,IDX,IX,IY,LGEOM1,LGEOM2,LGEOM3,
     *                  LGEOM4,LWID,PWID,SA,SI,ZTEMPD(85)
      DOUBLE PRECISION  DAI,DW,DXI,DYI
      DIMENSION         MSG901(20),MSG902(21),SCALEF(11)
      DOUBLE PRECISION  SCALEF
      DATA     MSG901 / 4H NO ,4HSCAL,4HE FA,4HCTOR,4H ON ,4HSCAL,
     *                  4HEF L,4HIST ,4HWILL,4H ALL,4HOW T,4HHE A,
     *                  4HPPRO,4HACH ,4HTO B,4HE PL,4HOTTE,4HD = ,
     *                  4HFNDX,4HYP    /
      DATA     MSG902 / 4H NO ,4HSCAL,4HE FA,4HCTOR,4H ON ,4HSCAL,
     *                  4HEF L,4HIST ,4HWILL,4H ALL,4HOW T,4HHE I,
     *                  4HNTER,4HSECT,4HION ,4HTO B,4HE PL,4HOTTE,
     *                  4HD = ,4HFNDX,4HYP    /
      DATA     NSCALE / 11 /
      DATA     SCALEF /  10.0D+00, 15.0D+00, 20.0D+00, 25.0D+00,
     *                   30.0D+00, 40.0D+00, 50.0D+00, 75.0D+00,
     *                  100.0D+00,200.0D+00,250.0D+00/
  601 FORMAT(12X,35HAPPROACH SCALE FACTOR TO BE USED IS,F6.1,5H FEET,
     *       9H PER INCH,//,12X,36HINTERSECTION SCALE FACTOR TO BE USED,
     *       3H IS,F6.1,14H FEET PER INCH)
  602 FORMAT(/)
C
C-----SUBROUTINE FNDXYP FINDS THE X AND Y COORDINATES FOR A POINT AT THE
C-----MIDDLE AND END OF EACH INBOUND LANE AND AT THE MIDDLE AND START OF
C-----EACH OUTBOUND LANE THAT IS AVAILABLE AT THE INTERSECTION, FINDS
C-----THE BOUNDARIES FOR PLOTTING, AND FINDS THE PLOT SCALE FACTORS
C
C-----PROCESS EACH INBOUND APPROACH
      DO 1040  IAN = 1 , NIBA
      IA = LIBA(IAN)
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                   COLEASE
      DXI = D0P0
C-----PROCESS EACH LANE OF THE INBOUND APPROACH
      DO 1030  ILN = 1 , NLANES
      IL = LLANES(ILN)
C     COLEASE,FIND,LWID,LANE,IL,LWID
      CALL  FIND    (LWID   ,      4,IL     ,      1)                   COLEASE
      DW = DBLE(LWID/2.0)
      DXI = DXI + DW
C     COLEASE,FIND,LGEOM3,LANE,IL,LGEOM(3)
      CALL  FIND    (LGEOM3,       4,IL     ,     16)                   COLEASE
C     COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
      CALL  FIND    (LGEOM4,       4,IL     ,     17)                   COLEASE
      DYI = LGEOM4
      IXAPP(IL) = -1
      IYAPP(IL) = -1
                      IF ( LGEOM3 . EQ . LGEOM4 )          GO TO 1010
C-----FIND THE X AND Y COORDINATES FOR THE END OF THE LANE
      CALL  XROTAI  ( DXI,DYI,IAAZIM,IAPX,IAPY,IXAPP(IL),IYAPP(IL) )
```

```
      DAI = DYI - 5.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 1010 CONTINUE
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND    (LGEOM1,      4,IL     ,     14)                        COLEASE
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                     IF ( LGEOM3 . EQ . LGEOM4 )  GO TO 1020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 1020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE   (IDX    ,      4,IL     ,     19)                       COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 1030 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1040 CONTINUE
C-----PROCESS EACH OUTBOUND APPROACH
      DO 2040  IAN = 1 , NOBA
      IA = LOBA(IAN)
C   COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                       COLEASE
      DXI = D0P0
C-----PROCESS EACH LANE OF OUTBOUND APPROACH
      DO 2030  ILN = 1 , NLANES
      IL = LLANES(ILN)
C   COLEASE,FIND,LWID,LANE,IL,LWID
      CALL  FIND    (LWID   ,      4,IL     ,      1)                       COLEASE
      DW = DBLE(LWID/2.0)
      DXI = DXI + DW
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND    (LGEOM1,       4,IL     ,     14)                       COLEASE
C   COLEASE,FIND,LGEOM2,LANE,IL,LGEOM(2)
      CALL  FIND    (LGEOM2,       4,IL     ,     15)                       COLEASE
      DYI = LGEOM1
      IXAPP(IL) = -1
      IYAPP(IL) = -1
            IF ( LGEOM1 . EQ . LGEOM2 )          GO TO 2010
C-----FIND THE X AND Y COORDINATES FOR THE START OF THE LANE
      CALL  XROTAI  ( DXI,DYI,IAAZIM,IAPX,IAPY,IXAPP(IL),IYAPP(IL) )
      DAI = DYI + 15.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 2010 CONTINUE
C   COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
      CALL  FIND    (LGEOM4,       4,IL     ,     17)                       COLEASE
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                     IF ( LGEOM1 . EQ . LGEOM2 )  GO TO 2020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 2020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE   (IDX    ,      4,IL     ,     19)                       COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 2030 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2040 CONTINUE
C-----ADD 1 FOOT BORDERS FOR APPROACH PLOT BOUNDARIES
      MINXA = MINXA - 1
      MINYA = MINYA - 1
      MAXXA = MAXXA + 1
      MAXYA = MAXYA + 1
C-----ADD 1 FOOT BORDERS FOR INTERSECTION PLOT BOUNDARIES AND ENSURE
C-----THAT AT LEAST THE LAST 20 FEET OF EACH INBOUND LANE AND THE FIRST
C-----20 FEET OF EACH OUTBOUND LANE WILL BE PLOTTED
      MINXI = MINXI - 21
      MINYI = MINYI - 21
      MAXXI = MAXXI + 21
      MAXYI = MAXYI + 21
                     IF ( IPLOT . EQ . 3 )         GO TO 4040
      PWID = IPAPER - 1
                     IF ( SCALEA . LE . D0P0 )     GO TO 3010
C-----CHECK APPROACH PLOT SCALE FACTOR FROM INPUT
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                             GO TO 3030
 3010 CONTINUE
C-----FIND APPROACH PLOT SCALE FACTOR THAT WILL MAKE THE PLOT AS LARGE
C-----AS POSSIBLE ON THE PLOT PAGE
      DO 3020  I = 1 , NSCALE
      SCALEA = SCALEF(I)
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                             GO TO 3030
 3020 CONTINUE
      GO TO 9010
 3030 CONTINUE
                     IF ( SCALEI . LE . D0P0 )     GO TO 4010
C-----CHECK INTERSECTION PLOT SCALE FACTOR FROM INPUT
      XSIZEI = (MAXXI-MINXI)/SCALEI
```

```
      DAI = DYI - 5.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 1010 CONTINUE
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND    (LGEOM1,     4,IL     ,     14)                      COLEASE
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                    IF ( LGEOM3 . EQ . LGEOM4 )  GO TO 1020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 1020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE   (IDX    ,     4,IL     ,     19)                     COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 1030 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1040 CONTINUE
C-----PROCESS EACH OUTBOUND APPROACH
      DO 2040  IAN = 1 , NOBA
      IA = LOBA(IAN)
C   COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                    COLEASE
      DXI = D0P0
C-----PROCESS EACH LANE OF OUTBOUND APPROACH
      DO 2030  ILN = 1 , NLANES
      IL = LLANES(ILN)
C   COLEASE,FIND,LWID,LANE,IL,LWID
      CALL  FIND    (LWID   ,     4,IL     ,      1)                     COLEASE
      DW = DBLE(LWID/2.0)
      DXI = DXI + DW
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND    (LGEOM1,     4,IL     ,     14)                      COLEASE
C   COLEASE,FIND,LGEOM2,LANE,IL,LGEOM(2)
      CALL  FIND    (LGEOM2,     4,IL     ,     15)                      COLEASE
      DYI = LGEOM1
      IXAPP(IL) = -1
      IYAPP(IL) = -1
                IF ( LGEOM1 . EQ . LGEOM2 )          GO TO 2010
C-----FIND THE X AND Y COORDINATES FOR THE START OF THE LANE
      CALL  XROTAI  ( DXI,DYI,IAAZIM,IAPX,IAPY,IXAPP(IL),IYAPP(IL) )
      DAI = DYI + 15.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 2010 CONTINUE
C   COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
      CALL  FIND    (LGEOM4,     4,IL     ,     17)                      COLEASE
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                    IF ( LGEOM1 . EQ . LGEOM2 )  GO TO 2020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 2020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE   (IDX    ,     4,IL     ,     19)                     COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 2030 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2040 CONTINUE
C-----ADD 1 FOOT BORDERS FOR APPROACH PLOT BOUNDARIES
      MINXA = MINXA - 1
      MINYA = MINYA - 1
      MAXXA = MAXXA + 1
      MAXYA = MAXYA + 1
C-----ADD 1 FOOT BORDERS FOR INTERSECTION PLOT BOUNDARIES AND ENSURE
C-----THAT AT LEAST THE LAST 20 FEET OF EACH INBOUND LANE AND THE FIRST
C-----20 FEET OF EACH OUTBOUND LANE WILL BE PLOTTED
      MINXI = MINXI - 21
      MINYI = MINYI - 21
      MAXXI = MAXXI + 21
      MAXYI = MAXYI + 21
                    IF ( IPLOT . EQ . 3 )        GO TO 4040
      PWID = IPAPER - 1
                    IF ( SCALEA . LE . D0P0 )    GO TO 3010
C-----CHECK APPROACH PLOT SCALE FACTOR FROM INPUT
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                           GO TO 3030
 3010 CONTINUE
C-----FIND APPROACH PLOT SCALE FACTOR THAT WILL MAKE THE PLOT AS LARGE
C-----AS POSSIBLE ON THE PLOT PAGE
      DO 3020  I = 1 , NSCALE
      SCALEA = SCALEF(I)
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                           GO TO 3030
 3020 CONTINUE
      GO TO 9010
 3030 CONTINUE
                    IF ( SCALEI . LE . D0P0 )    GO TO 4010
C-----CHECK INTERSECTION PLOT SCALE FACTOR FROM INPUT
      XSIZEI = (MAXXI-MINXI)/SCALEI
```

```
      YSIZEI = (MAXYI-MINYI)/SCALEI
      CSIZEI = XSIZEI/80,0D+00
      IF ( YSIZEI+8,0*CSIZEI,LE,PWID , AND , XSIZEI,LE,PWID )
     *                                                GO TO 4030
 4010 CONTINUE
C-----FIND INTERSECTION SCALE FACTOR THAT WILL MAKE THE PLOT AS LARGE
C-----AS POSSIBLE ON THE PLOT PAGE
      DO 4020  I = 1 , NSCALE
      SCALEI = SCALEF(I)
      XSIZEI = (MAXXI-MINXI)/SCALEI
      YSIZEI = (MAXYI-MINYI)/SCALEI
      CSIZEI = XSIZEI/80,0D+00
      IF ( YSIZEI+8,0*CSIZEI,LE,PWID , AND , XSIZEI,LE,PWID )
     *                                                GO TO 4030
 4020 CONTINUE
      GO TO 9020
 4030 CONTINUE
C-----PRINT APPROACH AND INTERSECTION PLOT SCALE FACTOR TO BE USED
                        IF ( NLINE+5 , GT , LINES )  CALL HEADER
      SA = SCALEA
      SI = SCALEI
      PRINT 601 , SA,SI
      NLINE = NLINE + 3
 4040 CONTINUE
      PRINT 602
      NLINE = NLINE + 2
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9010 CONTINUE
      CALL  ABORTR  ( MSG901,78 )
      STOP  901
 9020 CONTINUE
      CALL  ABORTR  ( MSG902,82 )
      STOP  902
      END                                                               FNDXYP
```

```
      SUBROUTINE FNDSDR                                                 COLEASE
C     TASK,FNDSDR
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / ZTEMPD / DX1,DX2,DY1,XFROM,XINT,XSDR,X1,X2,X3,X4,YFROM,
     *                  YINT,YSDR,Y1,Y2,Y3,Y4,I,IAZIM,IMAXL,INDEX,ISDRC,
     *                  ISDRCN,ISEE,ISTART,ISTOP,ITEST,IXCLAP,IYCLAP,
     *                  JMAXL,MAXSEE,NSDRAP,ZTEMPD(56)
      DOUBLE PRECISION  DUM,DX1,DX2,DY1,XFROM,XINT,XSDR,X1,X2,X3,X4,
     *                  YFROM,YINT,YSDR,Y1,Y2,Y3,Y4
      DIMENSION         MSG903(10),MSG904(17),MSG905(19)
      DOUBLE PRECISION  XBIG
      EQUIVALENCE       (DY1,DUM)
      DATA      MSG903 / 4H APP,4HROAC,4HHES ,4HDO N,4HOT I,4HNTER,
     *                   4HSECT,4H = F,4HNDSD,4HR    /
      DATA      MSG904 / 4H NUM,4HBER ,4HOF S,4HIGHT,4H DIS,4HTANC,
     *                   4HE RE,4HSTRI,4HCTIO,4HNS F,4HOR A,4HPPRO,
     *                   4HACH ,4HIS G,4HT 5 ,4H= FN,4HDSDR /
      DATA      MSG905 / 4H NUM,4HBER ,4HOF E,4HNTRI,4HES F,4HOR S,
     *                   4HIGHT,4H DIS,4HTANC,4HE RE,4HSTRI,4HCTIO,
     *                   4HN EN,4HTITY,4H IS ,4HGT 3,4H0 = ,4HFNDS,
     *                   4HDR   /
      DATA      XBIG   / 2000,0D+00 /
  601 FORMAT(8X,5HTABLE,I3,39H  =  LISTING OF SIGHT DISTANCE RESTRICT,
     *       11HION ENTRIES,//)
  602 FORMAT(12X,32HSIGHT DISTANCE RESTRICTION ENTRY,I3,10H IS NUMBER,
     *       I2,13H FOR APPROACH,I3,/,15X,21HAND INVOLVES APPROACH,I3,/)
  603 FORMAT(15X,8HAPPROACH,I3,5H FROM,I5,3H TO,I5,9H CAN SEE ,
     *       8HAPPROACH,I3,5H FROM,I5,3H TO,I5)
  604 FORMAT(/)
C
C-----SUBROUTINE FNDSDR FINDS THE SIGHT DISTANCE RESTRICTIONS BETWEEN
C-----THE INBOUND APPROACHES
C
                        IF ( NSDRC , LE , 0 )        RETURN
C-----PROCESS EACH INBOUND APPROACH
      DO 3020  IAN = 1 , NIBA
      IA = LIBA(IAN)
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                   COLEASE
      DX1 = D0P0
      IMAXL = 0
C-----FIND THE CENTER OF THE LANES FOR THE APPROACH AND THE MAXIMUM LANE
C-----LENGTH DOWN THE APPROACH
      DO 1010  ILN = 1 , NLANES
      IL = LLANES(ILN)
C     COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (      4,IL     )                                   COLEASE
      DX1 = DX1 + LWID/2,0
                        IF ( LGEOM(3),EQ,LGEOM(4) )  GO TO 1010
      IMAXL = MAX0(IMAXL,LGEOM(4))
 1010 CONTINUE
      IAZIM = IAAZIM
      IXCLAP = IAPX
      IYCLAP = IAPY
      NSDRAP = 0
C-----CHECK AGAINST EACH OTHER INBOUND APPROACH
      DO 3010  JAN = 1 , NIBA
```

```
      DAI = DYI - 5.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 1010 CONTINUE
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND     (LGEOM1,      4,IL     ,      14)                      COLEASE
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                     IF ( LGEOM3 . EQ . LGEOM4 )  GO TO 1020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 1020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE    (IDX    ,      4,IL     ,      19)                      COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 1030 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1040 CONTINUE
C-----PROCESS EACH OUTBOUND APPROACH
      DO 2040  IAN = 1 , NOBA
      IA = LOBA(IAN)
C   COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                       COLEASE
      DXI = D0P0
C-----PROCESS EACH LANE OF OUTBOUND APPROACH
      DO 2030  ILN = 1 , NLANES
      IL = LLANES(ILN)
C   COLEASE,FIND,LWID,LANE,IL,LWID
      CALL  FIND     (LWID   ,      4,IL     ,       1)                      COLEASE
      DW = DBLE(LWID/2.0)
      DXI = DXI + DW
C   COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
      CALL  FIND     (LGEOM1,      4,IL     ,      14)                      COLEASE
C   COLEASE,FIND,LGEOM2,LANE,IL,LGEOM(2)
      CALL  FIND     (LGEOM2,      4,IL     ,      15)                      COLEASE
      DYI = LGEOM1
      IXAPP(IL) = -1
      IYAPP(IL) = -1
            IF ( LGEOM1 . EQ . LGEOM2 )          GO TO 2010
C-----FIND THE X AND Y COORDINATES FOR THE START OF THE LANE
      CALL  XROTAI  ( DXI,DYI,IAAZIM,IAPX,IAPY,IXAPP(IL),IYAPP(IL) )
      DAI = DYI + 15.0
C-----FIND THE X AND Y COORDINATES FOR THE LOCATION OF THE TURN
C-----DIRECTION ARROWS
      CALL  XROTAI  ( DXI,DAI,IAAZIM,IAPX,IAPY,LTDIRX(IL),LTDIRY(IL) )
 2010 CONTINUE
C   COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
      CALL  FIND     (LGEOM4,      4,IL     ,      17)                      COLEASE
```

```
C-----FIND THE BOUNDARIES FOR THE APPROACH PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM4),IAAZIM,IAPX,IAPY,IX,IY )
      MINXA = MIN0(MINXA,IX)
      MAXXA = MAX0(MAXXA,IX)
      MINYA = MIN0(MINYA,IY)
      MAXYA = MAX0(MAXYA,IY)
                     IF ( LGEOM1 . EQ . LGEOM2 )  GO TO 2020
C-----FIND THE BOUNDARIES FOR THE INTERSECTION PLOT
      CALL  XROTAI  ( DXI-DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
      CALL  XROTAI  ( DXI+DW,DFLOAT(LGEOM1),IAAZIM,IAPX,IAPY,IX,IY )
      MINXI = MIN0(MINXI,IX)
      MAXXI = MAX0(MAXXI,IX)
      MINYI = MIN0(MINYI,IY)
      MAXYI = MAX0(MAXYI,IY)
 2020 CONTINUE
C-----FIND THE DISTANCE TO THE CENTER OF THE LANE FROM THE CENTER LINE
C-----OF THE APPROACH
      IDX = DXI + XROUND
C   COLEASE,STORE,IDX,LANE,IL,IDX
      CALL  STORE    (IDX    ,      4,IL     ,      19)                      COLEASE
      DXI = DXI + LWID - DW
C-----END OF LANE LOOP
 2030 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2040 CONTINUE
C-----ADD 1 FOOT BORDERS FOR APPROACH PLOT BOUNDARIES
      MINXA = MINXA - 1
      MINYA = MINYA - 1
      MAXXA = MAXXA + 1
      MAXYA = MAXYA + 1
C-----ADD 1 FOOT BORDERS FOR INTERSECTION PLOT BOUNDARIES AND ENSURE
C-----THAT AT LEAST THE LAST 20 FEET OF EACH INBOUND LANE AND THE FIRST
C-----20 FEET OF EACH OUTBOUND LANE WILL BE PLOTTED
      MINXI = MINXI - 21
      MINYI = MINYI - 21
      MAXXI = MAXXI + 21
      MAXYI = MAXYI + 21
                     IF ( IPLOT . EQ . 3 )          GO TO 4040
      PWID = IPAPER - 1
                     IF ( SCALEA . LE . D0P0 )      GO TO 3010
C-----CHECK APPROACH PLOT SCALE FACTOR FROM INPUT
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                              GO TO 3030
 3010 CONTINUE
C-----FIND APPROACH PLOT SCALE FACTOR THAT WILL MAKE THE PLOT AS LARGE
C-----AS POSSIBLE ON THE PLOT PAGE
      DO 3020  I = 1 , NSCALE
      SCALEA = SCALEF(I)
      XSIZEA = (MAXXA-MINXA)/SCALEA
      YSIZEA = (MAXYA-MINYA)/SCALEA
      CSIZEA = XSIZEA/80.0D+00
      IF ( YSIZEA+8.0*CSIZEA.LE.PWID . AND . XSIZEA.LE.PWID )
     *                                              GO TO 3030
 3020 CONTINUE
      GO TO 9010
 3030 CONTINUE
                     IF ( SCALEI . LE . D0P0 )      GO TO 4010
C-----CHECK INTERSECTION PLOT SCALE FACTOR FROM INPUT
      XSIZEI = (MAXXI-MINXI)/SCALEI
```

```
                 IF ( IAN . EQ . JAN )        GO TO 3010
      JA = LIBA(JAN)
C     COLEASE,EXTRAC,APPRO,JA
      CALL  EXTRAC  (      1,JA     )                                  COLEASE
C-----IF THE APPROACHES GO IN PARALLEL DIRECTIONS THEN THERE IS NO
C-----SIGHT DISTANCE RESTRICTION BETWEEN THESE APPROACHES
                 IF ( IAAZIM.EQ.IAZIM      )   GO TO 3010
                 IF ( IAAZIM.EQ.IAZIM+360 )   GO TO 3010
                 IF ( IAAZIM.EQ.IAZIM-360 )   GO TO 3010
                 IF ( IAAZIM+360.EQ.IAZIM )   GO TO 3010
                 IF ( IAAZIM-360.EQ.IAZIM )   GO TO 3010
                 IF ( IAAZIM.EQ.IAZIM+180 )   GO TO 3010
                 IF ( IAAZIM.EQ.IAZIM-180 )   GO TO 3010
                 IF ( IAAZIM+180.EQ.IAZIM )   GO TO 3010
                 IF ( IAAZIM-180.EQ.IAZIM )   GO TO 3010
      DX2 = D0P0
      JMAXL = 0
C-----FIND THE CENTER OF THE LANES FOR THE APPROACH BEING CHECKED
C-----AGAINST AND THE MAXIMUM LANE LENGTH DOWN THAT APPROACH
      DO 1020  JLN = 1 , NLANES
      JL = LLANES(JLN)
C     COLEASE,EXTRAC,LANE,JL
      CALL  EXTRAC  (      4,JL     )                                  COLEASE
      DX2 = DX2 + LWID/2.0
                 IF ( LGEOM(3).EQ.LGEOM(4) )  GO TO 1020
      JMAXL = MAX0(JMAXL,LGEOM(4))
 1020 CONTINUE
C-----FIND THE INTERSECTION OF THE TWO APPROACHES
      CALL  XROTAX  ( DX2,D0P0,IAAZIM,IAPX,IAPY,X2,Y2 )
      CALL  XROTAX  ( DX2,XBIG,IAAZIM,IAPX,IAPY,X3,Y3 )
      CALL  XROTAX  ( DX1,D0P0,IAZIM,IXCLAP,IYCLAP,X1,Y1 )
      CALL  XROTAX  ( DX1,XBIG,IAZIM,IXCLAP,IYCLAP,X4,Y4 )
      ITEST = LTOL( X1,Y1,X4,Y4,X2,Y2,X3,Y3,XINT,YINT,DUM,DUM )
                 IF ( ITEST . NE . 1 )        GO TO 9030
      X3 = XINT
      Y3 = YINT
C-----FIND THE MAXIMUM DISTANCE DOWN THE OTHER APPROACH THAT CAN BE SEEN
C-----FROM THE CENTER OF EVERY 25 FOOT SECTION DOWN THE APPROACH BEING
C-----PROCESSED
      INDEX = 0
      DY1 = -12.5D+00
 1030 CONTINUE
      DY1 = DY1 + 25.0D+00
      INDEX = INDEX + 1
      CALL  XROTAX  ( DX1,DY1,IAZIM,IXCLAP,IYCLAP,XFROM,YFROM )
      MAXSEE = 0
C-----CHECK EACH SIGHT DISTANCE RESTRICTION COORDINATE WHILE AT THIS
C-----SECTION
      DO 1040  ISDRCN = 1 , NSDRC
      ISDRC = LSDRC(ISDRCN)
      XSDR = IXSDRC(ISDRC)
      YSDR = IYSDRC(ISDRC)
      ISEE = LDOWN( XFROM,YFROM,XSDR,YSDR,X2,Y2,X3,Y3 )
      MAXSEE = MAX0(MAXSEE,ISEE)
 1040 CONTINUE
      ICANSE(INDEX) = MIN0(MAXSEE,JMAXL)
            IF ( DY1+12.6D+00.LT.DFLOAT(IMAXL) ) GO TO 1030
C-----IF YOU CAN SEE THE START OF THE OTHER APPROACH FROM EACH 25 FOOT
C-----SECTION ON THE APPROACH BEING PROCESSED THEN THERE IS NO SIGHT
C-----DISTANCE RESTRICTION BETWEEN THESE APPROACHES
      DO 1050  I = 1 , INDEX
                 IF ( ICANSE(I) . NE . 0 )    GO TO 2010
 1050 CONTINUE
      GO TO 3010
 2010 CONTINUE
C-----THERE IS A SIGHT DISTANCE RESTRICTION
                 IF ( NSDRS . NE . 0 )        GO TO 2020
                 IF ( NLINE+INDEX+8.GT.LINES )CALL HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
 2020 CONTINUE
C-----ADD SIGHT DISTANCE RESTRICTION FOR THE APPROACH BEING PROCESSED
      NSDRAP = NSDRAP + 1
                 IF ( NSDRAP . GT . 5 )       GO TO 9040
      NSDRS = NSDRS + 1
                 IF ( NSDRS . GT . 30 )       GO TO 9050
      INDEX = INDEX + 1
                 IF ( INDEX . GT . 40 )       GO TO 2040
      DO 2030  I = INDEX , 40
      ICANSE(I) = 0
 2030 CONTINUE
 2040 CONTINUE
C-----STORE SIGHT DISTANCE RESTRICTION INFORMATION IN ENTRY NSDRS OF
C-----ENTITY SDR
C     COLEASE,REPACK,SDR,NSDRS
      CALL  REPACK  (      7,NSDRS )                                   COLEASE
C-----STORE INFORMATION ABOUT SIGHT DISTANCE RESTRICTION FOR APPROACH
C     COLEASE,STORE,NSDRAP,APPRO,IA,NSDR
      CALL  STORE   (NSDRAP,      1,IA      ,      13)                 COLEASE
C     COLEASE,STORE,NSDRS,APPRO,IA,ISDRN(NSDRAP)
      CALL  STORE   (NSDRS ,      1,IA      ,      13+NSDRAP)          COLEASE
C     COLEASE,STORE,JA,APPRO,IA,ISDRA(NSDRAP)
      CALL  STORE   (JA    ,      1,IA      ,      18+NSDRAP)          COLEASE
C-----PRINT SIGHT DISTANCE RESTRICTION
      INDEX = INDEX - 1
                 IF ( NLINE+INDEX+5.GT.LINES )CALL HEADER
      PRINT 602 , NSDRS,NSDRAP,IA,JA
      NLINE = NLINE + 3
      ISTART = -25
      ISTOP = 0
      DO 2050  I = 1 , INDEX
      ISTART = ISTART + 25
      ISTOP = MIN0(ISTOP+25,IMAXL)
      PRINT 603 , IA,ISTART,ISTOP,JA,ICANSE(I),JMAXL
      NLINE = NLINE + 1
 2050 CONTINUE
      PRINT 604
      NLINE = NLINE + 2
C-----END OF OTHER APPROACH LOOP
 3010 CONTINUE
C-----END OF APPROACH LOOP
 3020 CONTINUE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9030 CONTINUE
      CALL  ABORTR  ( MSG903,37 )
      STOP  903
 9040 CONTINUE
      CALL  ABORTR  ( MSG904,68 )
      STOP  904
 9050 CONTINUE
      NSDRS = 30
      CALL  ABORTR  ( MSG905,74 )
      STOP  905
      END                                                              FNDSDR
```

```
      FUNCTION   LTOL   ( X1,Y1,X2,Y2,X3,Y3,X4,Y4,XI1,YI1,XI2,YI2 )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / DRWVAR(96),XBA,XBB,XMA,XMB,ZTEMPD(1)
      DOUBLE PRECISION  XBA,XBB,XMA,XMB
      DOUBLE PRECISION  CLOSE,XI1,XI2,X1,X2,X3,X4,YI1,YI2,Y1,Y2,Y3,Y4
      DATA      CLOSE  / 1.000001D+00 /
C
C-----FUNCTION LTOL TESTS IF LINE A FROM (X1,Y1) TO (X2,Y2) INTERSECTS
C-----WITH LINE B FROM (X3,Y3) TO (X4,Y4)  (LTOL=0=NO, LTOL=1=YES, AND
C-----LTOL=2=PARALLEL AND SAME)
C
      LTOL = 0
C-----IF LINE A VERTICAL THEN GO TO 1010
                    IF ( DABS(X2-X1).LE.ZERO )    GO TO 1010
      XMA = (Y2-Y1)/(X2-X1)
      XBA = Y1 - X1*XMA
C-----IF LINE B VERTICAL THEN GO TO 1020
                    IF ( DABS(X4-X3).LE.ZERO )    GO TO 1020
      XMB = (Y4-Y3)/(X4-X3)
      XBB = Y3 - X3*XMB
C-----IF THE SLOPE OF LINE A IS EQUAL TO THE SLOPE OF LINE B THEN LINE A
C-----IS PARALLEL TO LINE B THUS GO TO 2010
                    IF ( DABS(XMA-XMB).LE.ZERO ) GO TO 2010
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XI1 = (XBB-XBA)/(XMA-XMB)
      YI1 = XMA*XI1 + XBA
      GO TO 1030
 1010 CONTINUE
C-----IF LINE B IS ALSO VERTICAL THEN LINE A IS PARALLEL TO LINE B THUS
C-----GO TO 3010
                    IF ( DABS(X4-X3).LE.ZERO )    GO TO 3010
      XMB = (Y4-Y3)/(X4-X3)
      XBB = Y3 - X3*XMB
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XI1 = X1
      YI1 = XMB*XI1 + XBB
      GO TO 1030
 1020 CONTINUE
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XI1 = X3
      YI1 = XMA*XI1 + XBA
 1030 CONTINUE
C-----IF (XI1,YI1) DOES NOT LIE BETWEEN (X1,Y1) AND (X2,Y2) THEN THE
C-----POINT OF INTERSECTION DOES NOT LIE ON LINE A THUS RETURN (LTOL=0)
               IF ( (XI1-X1)*(XI1-X2).GT.ZERO )      RETURN
               IF ( (YI1-Y1)*(YI1-Y2).GT.ZERO )      RETURN
C-----IF (XI1,YI1) DOES NOT LIE BETWEEN (X3,Y3) AND (X4,Y4) THEN THE
C-----POINT OF INTERSECTION DOES NOT LIE ON LINE B THUS RETURN (LTOL=0)
               IF ( (XI1-X3)*(XI1-X4).GT.ZERO )      RETURN
               IF ( (YI1-Y3)*(YI1-Y4).GT.ZERO )      RETURN
C-----LINE A INTERSECTS LINE B
      LTOL = 1
      RETURN
 2010 CONTINUE
C-----LINE A IS PARALLEL TO LINE B THUS FIND THE X AND Y COORDINATES FOR
C-----THE PARTS OF THE LINES THAT OVERLAP
      XI1 = DMAX1( DMIN1(X1,X2),DMIN1(X3,X4) )
      XI2 = DMIN1( DMAX1(X1,X2),DMAX1(X3,X4) )
C-----IF THE MINIMUM X COORDINATE IS GREATER THAN THE MAXIMUM X
C-----COORDINATE THEN RETURN (LTOL=0)
                    IF ( XI1-CLOSE . GT . XI2 )  RETURN
      YI1 = DMAX1( DMIN1(Y1,Y2),DMIN1(Y3,Y4) )
      YI2 = DMIN1( DMAX1(Y1,Y2),DMAX1(Y3,Y4) )
C-----IF THE MINIMUM Y COORDINATE IS GREATER THAN THE MAXIMUM Y
C-----COORDINATE THEN RETURN (LTOL=0)
                    IF ( YI1-CLOSE . GT . YI2 )  RETURN
C-----IF THE PERPENDICULAR DISTANCE BETWEEN THE LINES IS NOT CLOSE THEN
C-----RETURN (LTOL=0) ELSE THE LINES ARE PARALLEL AND THE SAME THUS
C-----RETURN (LTOL=2)
      IF ( DABS(XBA-XBB)*DCOS(DATAN(0.5D+00*(XMA+XMB))) . GT . CLOSE )
     *                                                            RETURN
      LTOL = 2
      RETURN
 3010 CONTINUE
C-----LINE A AND LINE B ARE VERTICAL THUS FIND THE X AND Y COORDINATES
C-----FOR THE PARTS OF THE LINES THAT OVERLAP
      YI1 = DMAX1( DMIN1(Y1,Y2),DMIN1(Y3,Y4) )
      YI2 = DMIN1( DMAX1(Y1,Y2),DMAX1(Y3,Y4) )
C-----IF THE MINIMUM Y COORDINATE IS GREATER THAN THE MAXIMUM Y
C-----COORDINATE THEN RETURN (LTOL=0)
                    IF ( YI1-CLOSE . GT . YI2 )  RETURN
C-----IF THE X INTERCEPT OF THE LINES IS DIFFERENT THEN RETURN (LTOL=0)
C-----ELSE THE LINES ARE PARALLEL AND THE SAME THUS RETURN (LTOL=2)
      IF ( DABS(0.5D+00*(X1+X2)-0.5D+00*(X3+X4)) . GT . CLOSE )
     *                                                            RETURN
      XI1 = 0.25D+00*(X1+X2+X3+X4)
      XI2 = XI1
      LTOL = 2
      RETURN
      END                                                           LTOL
```

```
      FUNCTION   LDOWN  ( X1,Y1,XSDR,YSDR,X2,Y2,X3,Y3 )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / FNDSDR(50),XBA,XBB,XINT,XMA,XMB,YINT,ZTEMPD(43)
      DOUBLE PRECISION  XBA,XBB,XINT,XMA,XMB,YINT
      DOUBLE PRECISION  XSDR,X1,X2,X3,YSDR,Y1,Y2,Y3
C
C-----FUNCTION LDOWN FINDS THE DISTANCE FROM (X2,Y2) TO (XINT,YINT) IF
C-----LINE A FROM (X1,Y1) THROUGH (XSDR,YSDR) INTERSECTS WITH LINE B
C-----FROM (X2,Y2) TO (X3,Y3)  (LDOWN=0=NO INTERSECTION)
C
      LDOWN = 0
C-----IF LINE A VERTICAL THEN GO TO 1010
                    IF ( DABS(XSDR-X1).LE.ZERO ) GO TO 1010
      XMA = (YSDR-Y1)/(XSDR-X1)
      XBA = Y1 - X1*XMA
C-----IF LINE B VERTICAL THEN GO TO 1020
                    IF ( DABS(X3-X2).LE.ZERO )   GO TO 1020
      XMB = (Y3-Y2)/(X3-X2)
      XBB = Y2 - X2*XMB
C-----IF THE SLOPE OF LINE A IS EQUAL TO THE SLOPE OF LINE B THEN LINE A
C-----IS PARALLEL TO LINE B AND THERE IS NO INTERSECTION
                    IF ( DABS(XMA-XMB).LE.ZERO ) RETURN
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XINT = (XBB-XBA)/(XMA-XMB)
      YINT = XMA*XINT + XBA
      GO TO 1030
 1010 CONTINUE
C-----IF LINE B IS ALSO VERTICAL THEN LINE A IS PARALLEL TO LINE B AND
C-----THERE IS NO INTERSECTION
                    IF ( DABS(X3-X2).LE.ZERO )   RETURN
      XMB = (Y3-Y2)/(X3-X2)
      XBB = Y2 - X2*XMB
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XINT = X1
      YINT = XMB*XINT + XBB
      GO TO 1030
 1020 CONTINUE
C-----FIND THE INTERSECTION OF LINE A AND LINE B
      XINT = X2
      YINT = XMA*XINT + XBA
 1030 CONTINUE
C-----IF (XSDR,YSDR) DOES NOT LIE BETWEEN (X1,Y1) AND (XINT,YINT) THEN
C-----THE POINT OF SIGHT DISTANCE RESTRICTION DOES NOT LIE BETWEEN THE
C-----DRIVER AND THE OTHER APPROACH AND THERE IS NO INTERSECTION
            IF ( (XSDR-X1)*(XSDR-XINT).GT.ZERO ) RETURN
            IF ( (YSDR-Y1)*(YSDR-YINT).GT.ZERO ) RETURN
C-----IF (XINT,YINT) DOES NOT LIE BETWEEN (X2,Y2) AND (X3,Y3) THEN THE
C-----POINT OF INTERSECTION DOES NOT LIE ON LINE B
            IF ( (XINT-X2)*(XINT-X3).GT.ZERO )   RETURN
            IF ( (YINT-Y2)*(YINT-Y3).GT.ZERO )   RETURN
C-----FIND THE DISTANCE FROM (X2,Y2) TO (XINT,YINT)
      LDOWN = DSQRT((X2-XINT)**2+(Y2-YINT)**2) + XROUND
      RETURN
      END                                                               LDOWN
```

```
      SUBROUTINE WRITAP                                                 COLEASE
C     TASK,WRITAP
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR     ,     COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST   ,     COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / I,ISDR,ZTEMPD(103)
  601 FORMAT(20I4)
C
C-----SUBROUTINE WRITAP WRITES THE APPROACH INFORMATION ONTO TAPE MODELT
C-----FOR SIMPRO
C
C-----WRITE THE NUMBER AND LIST OF INBOUND APPROACHES ONTO MODELT
      WRITE (MODELT,601)  NIBA
      WRITE (MODELT,601)  (LIBA(I),I=1,NIBA)
C-----WRITE THE NUMBER AND LIST OF OUTBOUND APPROACHES ONTO MODELT
      WRITE (MODELT,601)  NOBA
      WRITE (MODELT,601)  (LOBA(I),I=1,NOBA)
C-----WRITE THE NUMBER OF APPROACHES ONTO MODELT
      WRITE (MODELT,601)  NAP
C-----WRITE THE INFORMATION FOR EACH INBOUND APPROACH ONTO MODELT
      DO 1010  IAN = 1 , NIBA
      IA = LIBA(IAN)
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (    1,IA     )                                     COLEASE
C-----WRITE THE INBOUND APPROACH INFORMATION ONTO MODELT
      WRITE (MODELT,601)  IA,IAAZIM,IAPX,IAPY,ISLIM,NLANES,NSDR,IALEFT,
     *                    IARGHT
      WRITE (MODELT,601)  (LLANES(ILN),ILN=1,NLANES)
                IF ( NSDR . LE . 0 )         GO TO 1010
      WRITE (MODELT,601)  (ISDRN(ISDR),ISDRA(ISDR),ISDR=1,NSDR)
 1010 CONTINUE
C-----WRITE THE INFORMATION FOR EACH OUTBOUND APPROACH ONTO MODELT
      DO 2010  IAN = 1 , NOBA
      IA = LOBA(IAN)
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (    1,IA     )                                     COLEASE
C-----WRITE THE OUTBOUND APPROACH INFORMATION ONTO MODELT
      WRITE (MODELT,601)  IA,IAAZIM,IAPX,IAPY,ISLIM,NLANES,NSDR,IALEFT,
     *                    IARGHT
      WRITE (MODELT,601)  (LLANES(ILN),ILN=1,NLANES)
                IF ( NSDR . LE . 0 )         GO TO 2010
      WRITE (MODELT,601)  (ISDRN(ISDR),ISDRA(ISDR),ISDR=1,NSDR)
 2010 CONTINUE
      RETURN
      END                                                               WRITAP
```

```
      SUBROUTINE INIPLT
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
C;    DIMENSION         IBUF(1024)
C
C-----SUBROUTINE INIPLT INITIALIZES PLOTTING
C
      GO TO ( 1010,2010,3010 ) , IPLOT
 1010 CONTINUE
C-----PLOT OPTION IS (PLOT    )
C¬    IF ( IPAPER , EQ , 12 )
C¬   *CALL  BGNPLT  ( 4LPLOT   )
C¬    IF ( IPAPER , EQ , 30 )
C¬   *CALL  BGNPLT  ( 5LPLOTW  )
C;    CALL  PLOTS   ( IBUF,1024,8 )
      GO TO 2020
 2010 CONTINUE
C-----PLOT OPTION IS (PLOTI   )
C¬    IF ( IPAPER , EQ , 12 )
C¬   *CALL  BGNPLT  ( 5LPLOTI  )
C¬    IF ( IPAPER , EQ , 30 )
C¬   *CALL  BGNPLT  ( 6LPLOTWI )
C;    CALL  PLOTS   ( IBUF,1024,8 )
C;    CALL  NEWPEN  ( 2 )
 2020 CONTINUE
CA    DO 101  KOUNT = 1 , 6
CA    CALL  PLT     ( 0,0,0,0,3 )
CA    CALL  PLT     ( 11,0,0,0,2 )
CA    CALL  PLT     ( 11,0,11,0,2 )
CA    CALL  PLT     ( 0,0,11,0,2 )
CA    CALL  PLT     ( 0,0,0,0,2 )
CA101 CONTINUE
C-----DRAW THE APPROACH PLOT
      CALL  DRWAPR
C-----DRAW THE INTERSECTION PLOT
      CALL  DRWINT
 3010 CONTINUE
      RETURN
      END                                                         INIPLT
```

```
      SUBROUTINE DRWAPR                                                 COLEASE
C   TASK,DRWAPR
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW    ,    COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2           COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / X,XBRDR,X1,X2,Y,YBRDR,Y1,Y2,IARC,IARCN,ILINE,
     *                  ILINEN,ISDRC,ISDRCN,IX1,IX2,JSCALE(4),JTITLE(8),
     *                  NLEFTD,XPAGE,YPAGE,ZTEMPD(66)
      DOUBLE PRECISION  X,XBRDR,X1,X2,Y,YBRDR,Y1,Y2
      DIMENSION         ISCALE(9)
      DATA     ISCALE / 4HSCAL,4HE FA,4HCTOR,4H IS ,4H    ,4H  FE,
     *                  4HET P,4HER I,4HNCH  /
C¬601 FORMAT(20A4)
C
C-----SUBROUTINE DRWAPR DRAWS THE APPROACH PLOT
C
                     IF ( IPLOT , EQ , 3 )          RETURN
C-----SET PLOT PARAMETERS FOR APPROACH PLOT
      SCALE = SCALEA
      XMIN = MINXA
      YMIN = MINYA
      XMAX = MAXXA
      YMAX = MAXYA
C-----FIND APPROACH PLOT BORDERS
      XBRDR = (IPAPER-1,0-XSIZEA)/2,0
      YBRDR = (IPAPER-1,0-YSIZEA-8,0*CSIZEA)/2,0
C-----RE-ORIGIN PLOT SO (XMIN,YMIN) WILL BE (0,0,0,0)
      X0 = XSIZEA + XBRDR
      Y0 = YBRDR + 4,0*CSIZEA
C¬    CALL  PLT     ( SNGL(XBRDR),SNGL(Y0),-3 )
C;    CALL  PLOT    ( SNGL(XBRDR),SNGL(Y0),-3 )
C-----DRAW THE PLOT SCALE FACTOR MESSAGE AT BOTTOM OF PLOT
      XPAGE = XSIZEA/2,0 - 0,5*35*CSIZEA
      YPAGE = -3,0*CSIZEA
C¬          ENCODE  ( 35,601,JSCALE )            ISCALE
C¬    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEA),JSCALE,0,0,35 )
C;    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEA),ISCALE,0,0,35 )
      NLEFTD = DLOG10(SCALE) + 1,0D+00
      XPAGE = XPAGE + (16+0,5*(3-NLEFTD))*CSIZEA
      CALL  NUMBER  ( XPAGE,YPAGE,SNGL(CSIZEA),SNGL(SCALE),0,0,1 )
C-----DRAW THE TITLE FOR GEOPRO AT TOP OF PLOT
      XPAGE = XSIZEA/2,0 - 40,0*CSIZEA
      YPAGE = YSIZEA + 2,0*CSIZEA
C¬          ENCODE  ( 80,601,JTITLE )            ITITLE
C¬    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEA),JTITLE,0,0,80 )
C;    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEA),ITITLE,0,0,80 )
C-----DRAW EACH INBOUND APPROACH
      DO 1060  IAN = 1 , NIBA
      IA = LIBA(IAN)
C   COLEASE,EXTRAC,APPRO,IA
```

```
      CALL  EXTRAC  (      1,IA     )                                  COLEASE
      IX1 = 0
C-----DRAW EACH LANE OF THE INBOUND APPROACH
      DO 1050  ILN = 1 , NLANES
      IL = LLANES(ILN)
C   COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (      4,IL     )                                  COLEASE
      IX2 = IX1 + LWID
                      IF ( LGEOM(1).NE.LGEOM(3) )  GO TO 1010
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(4) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(4) )
      GO TO 1040
 1010 CONTINUE
                      IF ( LGEOM(3).NE.LGEOM(4) )  GO TO 1030
 1020 CONTINUE
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(2) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(2) )
      GO TO 1040
 1030 CONTINUE
C-----DRAW A BOX FROM LGEOM(3) TO LGEOM(4) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(3),LGEOM(4) )
                      IF ( LGEOM(1).NE.LGEOM(2) )  GO TO 1020
 1040 CONTINUE
      IX1 = IX2
C-----END OF LANE LOOP
 1050 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1060 CONTINUE
C-----DRAW EACH OUTBOUND APPROACH
      DO 2060  IAN = 1 , NOBA
      IA = LOBA(IAN)
C   COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                  COLEASE
      IX1 = 0
C-----DRAW EACH LANE OF THE OUTBOUND APPROACH
      DO 2050  ILN = 1 , NLANES
      IL = LLANES(ILN)
C   COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (      4,IL     )                                  COLEASE
      IX2 = IX1 + LWID
                      IF ( LGEOM(1).NE.LGEOM(3) )  GO TO 2010
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(4) FOR THE OUTBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(4) )
      GO TO 2040
 2010 CONTINUE
                      IF ( LGEOM(3).NE.LGEOM(4) )  GO TO 2030
 2020 CONTINUE
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(2) FOR THE OUTBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(2) )
      GO TO 2040
 2030 CONTINUE
C-----DRAW A BOX FROM LGEOM(3) TO LGEOM(4) FOR THE OUTBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(3),LGEOM(4) )
                      IF ( LGEOM(1).NE.LGEOM(2) )  GO TO 2020
 2040 CONTINUE
      IX1 = IX2
C-----END OF LANE LOOP
 2050 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2060 CONTINUE
                      IF ( NARCS . LE . 0 )         GO TO 3020
C-----DRAW EACH ARC
      DO 3010  IARCN = 1 , NARCS
      IARC = LARCS(IARCN)
C   COLEASE,EXTRAC,ARC,IARC
      CALL  EXTRAC  (      2,IARC   )                                  COLEASE
      IARCSW = IARCSW - 360
      CALL  DRWARC  ( IARCX,IARCY,IARCAZ,IARCSW,IARCR )
 3010 CONTINUE
 3020 CONTINUE
                      IF ( NLINES . LE . 0 )        GO TO 4020
C-----DRAW EACH LINE
      DO 4010  ILINEN = 1 , NLINES
      ILINE = LLINES(ILINEN)
C   COLEASE,EXTRAC,LINE,ILINE
      CALL  EXTRAC  (      5,ILINE )                                   COLEASE
      X1 = ILX1
      Y1 = ILY1
      X2 = ILX2
      Y2 = ILY2
      CALL  DRWLIN  ( X1,Y1,X2,Y2 )
 4010 CONTINUE
 4020 CONTINUE
                      IF ( NSDRC . LE . 0 )         GO TO 5020
C-----DRAW EACH SIGHT DISTANCE RESTRICTION COORDINATE
      DO 5010  ISDRCN = 1 , NSDRC
      ISDRC = LSDRC(ISDRCN)
      X = IXSDRC(ISDRC)
      Y = IYSDRC(ISDRC)
C-----IF THE COORDINATES LIE OFF THE PLOT PAGE THEN SKIP THE POINT
                      IF ( X . LT . XMIN )          GO TO 5010
                      IF ( X . GT . XMAX )          GO TO 5010
                      IF ( Y . LT . YMIN )          GO TO 5010
                      IF ( Y . GT . YMAX )          GO TO 5010
C-----DRAW A 5 FOOT STAR AT COORDINATE
      XPAGE = (X-XMIN)/SCALE
      YPAGE = (Y-YMIN)/SCALE
      CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(5.0/SCALE),11,0.0,-1 )
 5010 CONTINUE
 5020 CONTINUE
      RETURN
      END                                                              DRWAPR
```

```
      SUBROUTINE DRWBOX (IX1,     IX2,     IL1,     IL2)                COLEASE
C    TASK,DRWBOX,IX1,IX2,IL1,IL2
      COMMON / APPRO  / IALEFT     ,IARGHT     ,NLANES     ,LLANES( 6),   COLEASE
     *                  IAPX       ,IAPY       ,ISLIM      ,NSDR      ,   COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM     ,NDEGST     ,   COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / ZTEMPD / DRWVAR(46),X1,X2,X3,X4,Y1,Y2,Y3,Y4,ZTEMPD(43)
      DOUBLE PRECISION  X1,X2,X3,X4,Y1,Y2,Y3,Y4
C
C-----SUBROUTINE DRWBOX DRAWS A BOX FROM IL1 TO IL2 FOR A LANE
C
C-----FIND THE COORDINATES OF THE EDGES OF THE BOX FOR THE LANE
      CALL  IROTAX  ( IX1,IL1,IAAZIM,IAPX,IAPY,X1,Y1 )
      CALL  IROTAX  ( IX2,IL1,IAAZIM,IAPX,IAPY,X2,Y2 )
      CALL  IROTAX  ( IX2,IL2,IAAZIM,IAPX,IAPY,X3,Y3 )
      CALL  IROTAX  ( IX1,IL2,IAAZIM,IAPX,IAPY,X4,Y4 )
C-----DRAW THE BOX FOR THE LANE
      CALL  DRWLIN  ( X1,Y1,X2,Y2 )
      CALL  DRWLIN  ( X2,Y2,X3,Y3 )
      CALL  DRWLIN  ( X3,Y3,X4,Y4 )
      CALL  DRWLIN  ( X4,Y4,X1,Y1 )
      RETURN
      END                                                               DRWBOX
```

```
      SUBROUTINE DRWLIN ( X1,Y1,X2,Y2 )
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / ZTEMPD / DRWVAR(72),DIST,DMIN,XDMIN,XINT,XX1,XX2,YDMIN,
     *                  YINT,YY1,YY2,LTEST,XPAGE,YPAGE,ZTEMPD(10)
      DOUBLE PRECISION  DIST,DMIN,XDMIN,XINT,XX1,XX2,YDMIN,YINT,YY1,YY2
      DOUBLE PRECISION  D,X1,X2,Y1,Y2
      EQUIVALENCE       (DIST,D)
C
C-----SUBROUTINE DRWLIN DRAWS A LINE ON THE PLOT PAGE
C
                        IF ( IPLOT . EQ . 3 )         RETURN
      XX1 = X1
      YY1 = Y1
      XX2 = X2
      YY2 = Y2
C-----IF THE START OF THE LINE IS OFF THE PLOT PAGE THEN GO TO 2010
                        IF ( XX1 . LT . XMIN )        GO TO 2010
                        IF ( XX1 . GT . XMAX )        GO TO 2010
                        IF ( YY1 . LT . YMIN )        GO TO 2010
                        IF ( YY1 . GT . YMAX )        GO TO 2010
 1010 CONTINUE
C-----IF THE END OF THE LINE IS OFF THE PLOT PAGE THEN GO TO 3010
                        IF ( XX2 . LT . XMIN )        GO TO 3010
                        IF ( XX2 . GT . XMAX )        GO TO 3010
                        IF ( YY2 . LT . YMIN )        GO TO 3010
                        IF ( YY2 . GT . YMAX )        GO TO 3010
 1020 CONTINUE
C-----MOVE PEN TO THE START OF THE LINE WITH THE PEN UP
      XPAGE = (XX1-XMIN)/SCALE
      YPAGE = (YY1-YMIN)/SCALE
C¬    CALL  PLT     ( XPAGE,YPAGE,3 )
C;    CALL  PLOT    ( XPAGE,YPAGE,3 )
C-----MOVE PEN TO THE END OF THE LINE WITH THE PEN DOWN
      XPAGE = (XX2-XMIN)/SCALE
      YPAGE = (YY2-YMIN)/SCALE
C¬    CALL  PLT     ( XPAGE,YPAGE,2 )
C;    CALL  PLOT    ( XPAGE,YPAGE,2 )
      RETURN
 2010 CONTINUE
C-----THE FIRST POINT IS OFF THE PLOT PAGE THUS FIND THE INTERSECTION
C-----OF THE LINE WITH THE BOUNDARY NEAREST THE FIRST POINT
      DMIN = 1.0D+99
C-----FIND THE INTERSECTION WITH THE BOTTOM EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMIN,YMIN,XMAX,YMIN,XINT,YINT,D,D )
                        IF ( LTEST . NE . 1 )         GO TO 2020
      DIST = DSQRT((XX1-XINT)**2+(YY1-YINT)**2)
                        IF ( DIST . GE . DMIN )       GO TO 2020
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 2020 CONTINUE
C-----FIND THE INTERSECTION WITH THE RIGHT EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMAX,YMIN,XMAX,YMAX,XINT,YINT,D,D )
                        IF ( LTEST . NE . 1 )         GO TO 2030
      DIST = DSQRT((XX1-XINT)**2+(YY1-YINT)**2)
                        IF ( DIST . GE . DMIN )       GO TO 2030
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 2030 CONTINUE
C-----FIND THE INTERSECTION WITH THE TOP EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMAX,YMAX,XMIN,YMAX,XINT,YINT,D,D )
                        IF ( LTEST . NE . 1 )         GO TO 2040
```

```
      DIST = DSQRT((XX1-XINT)**2+(YY1-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 2040
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 2040 CONTINUE
C-----FIND THE INTERSECTION WITH THE LEFT EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMIN,YMAX,XMIN,YMIN,XINT,YINT,D,D )
                      IF ( LTEST . NE . 1 )         GO TO 2050
      DIST = DSQRT((XX1-XINT)**2+(YY1-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 2050
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 2050 CONTINUE
C-----IF THE MINIMUM DISTANCE IS STILL A LARGE NUMBER THEN RETURN
C-----ELSE SET POINT ONE TO THE CLOSEST COORDINATES
                      IF ( DMIN . EQ . 1.0D+99 )    RETURN
      XX1 = XDMIN
      YY1 = YDMIN
      GO TO 1010
 3010 CONTINUE
C-----THE SECOND POINT IS OFF THE PLOT PAGE THUS FIND THE INTERSECTION
C-----OF THE LINE WITH THE BOUNDARY NEAREST THE SECOND POINT
      DMIN = 1.0D+99
C-----FIND THE INTERSECTION WITH THE BOTTOM EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMAX,YMIN,XMAX,YMIN,XINT,YINT,D,D )
                      IF ( LTEST . NE . 1 )         GO TO 3020
      DIST = DSQRT((XX2-XINT)**2+(YY2-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 3020
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 3020 CONTINUE
C-----FIND THE INTERSECTION WITH THE RIGHT EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMAX,YMIN,XMAX,YMAX,XINT,YINT,D,D )
                      IF ( LTEST . NE . 1 )         GO TO 3030
      DIST = DSQRT((XX2-XINT)**2+(YY2-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 3030
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 3030 CONTINUE
C-----FIND THE INTERSECTION WITH THE TOP EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMAX,YMAX,XMIN,YMAX,XINT,YINT,D,D )
                      IF ( LTEST . NE . 1 )         GO TO 3040
      DIST = DSQRT((XX2-XINT)**2+(YY2-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 3040
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 3040 CONTINUE
C-----FIND THE INTERSECTION WITH THE LEFT EDGE
      LTEST = LTOL( XX1,YY1,XX2,YY2,XMIN,YMAX,XMIN,YMIN,XINT,YINT,D,D )
                      IF ( LTEST . NE . 1 )         GO TO 3050
      DIST = DSQRT((XX2-XINT)**2+(YY2-YINT)**2)
                      IF ( DIST . GE . DMIN )       GO TO 3050
      DMIN = DIST
      XDMIN = XINT
      YDMIN = YINT
 3050 CONTINUE
C-----IF THE MINIMUM DISTANCE IS STILL A LARGE NUMBER THEN RETURN
C-----ELSE SET POINT TWO TO THE CLOSEST COORDINATES
                      IF ( DMIN . EQ . 1.0D+99 )    RETURN
      XX2 = XDMIN
      YY2 = YDMIN
      GO TO 1020
      END                                                               DRWLIN
```

```
      SUBROUTINE DRWARC ( IXARC,IYARC,IAZARC,ISWARC,IRARC )
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / DRWVAR(72),X,Y,ADD,ADDAZ,DEG,IADD,IPEN,XPAGE,
     *                  YPAGE,ZTEMPD(22)
      DOUBLE PRECISION  X,Y
C
C-----SUBROUTINE DRWARC DRAWS AN ARC ON THE PLOT PAGE
C
                      IF ( IPLOT . EQ . 3 )         RETURN
C-----THE STEP INCREMENT FOR THE AZIMUTH IS THE MINIMUM OF ONE-TENTH OF
C-----THE TOTAL SWEEP ANGLE AND 5 DEGREES
      ADDAZ = SIGN(AMIN1(IABS(ISWARC)/10.0,5.0),FLOAT(ISWARC))
      ADD = -ADDAZ
      IADD = 0
 1010 CONTINUE
C-----IF FINISHED PLOTTING THE ARC THEN RETURN
                      IF ( IADD.GE.IABS(ISWARC) )  RETURN
      IPEN = 3
 1020 CONTINUE
C-----FIND THE AZIMUTH OF A POINT ON THE ARC
      ADD = ADD + ADDAZ
      IADD = ABS(ADD) + XROUND
                      IF ( IADD.GE.IABS(ISWARC) )  ADD = ISWARC
C-----FIND THE X AND Y COORDINATES OF A POINT ON THE ARC
      DEG = 90 - (IAZARC+ADD)
      X = IXARC + IRARC*DCOS(DEG*RADIAN)
      Y = IYARC + IRARC*DSIN(DEG*RADIAN)
C-----IF THE POINT IS OFF THE PLOT PAGE THEN GO TO 1010
                      IF ( X . LT . XMIN )         GO TO 1010
                      IF ( X . GT . XMAX )         GO TO 1010
                      IF ( Y . LT . YMIN )         GO TO 1010
                      IF ( Y . GT . YMAX )         GO TO 1010
C-----MOVE TO THE POINT WITH THE PEN UP (IPEN=3) OR DOWN (IPEN=2)
      XPAGE = (X-XMIN)/SCALE
      YPAGE = (Y-YMIN)/SCALE
C-    CALL  PLT     ( XPAGE,YPAGE,IPEN )
C;    CALL  PLOT    ( XPAGE,YPAGE,IPEN )
      IPEN = 2
C-----IF FINISHED PLOTTING THE ARC THEN RETURN
                      IF ( IADD.GE.IABS(ISWARC) )  RETURN
      GO TO 1020
      END                                                               DRWARC
```

```
      SUBROUTINE DRWINT                                                 COLEASE
C    TASK,DRWINT
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW    ,    COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2           COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / X,XBRDR,X1,X2,Y,YBRDR,Y1,Y2,IAL,IAR,IARC,IARCN,
     *                  IAS,ILINE,ILINEN,ISDRC,ISDRCN,IX1,IX2,JSCALE(4),
     *                  JTITLE(8),KA,KAN,KL,KLN,NLEFTD,XPAGE,YPAGE,
     *                  ZTEMPD(59)
      DOUBLE PRECISION  X,XBRDR,X1,X2,Y,YBRDR,Y1,Y2
      DIMENSION         ISCALE(9)
      DATA      ISCALE / 4HSCAL,4HE FA,4HCTOR,4H IS ,4H    ,4H  FE,
     *                  4HET P,4HER I,4HNCH  /
C¬601 FORMAT(20A4)
C
C-----SUBROUTINE DRWINT DRAWS THE INTERSECTION PLOT
C-----(MAY NOT USE /INDEX/ BECAUSE CALLED BY FNDPTH)
C
                        IF ( IPLOT . EQ . 3 )       RETURN
C-----SET PLOT PARAMETERS FOR INTERSECTION PLOT
      SCALE = SCALEI
      XMIN = MINXI
      YMIN = MINYI
      XMAX = MAXXI
      YMAX = MAXYI
C-----RE-ORIGIN THE PLOT PAST THE LAST PLOT PAGE
C¬    CALL  PLT     ( 0,0,0,0,999 )
C;    CALL  PLOT    ( SNGL(X0+4.0),SNGL(-Y0),-3 )
CA    DO 101  KOUNT = 1 , 6
CA    CALL  PLT     ( 0,0,0,0,3 )
CA    CALL  PLT     ( 11,0,0,0,2 )
CA    CALL  PLT     ( 11,0,11,0,2 )
CA    CALL  PLT     ( 0,0,11,0,2 )
CA    CALL  PLT     ( 0,0,0,0,2 )
CA101 CONTINUE
C-----FIND THE INTERSECTION PLOT BORDERS
      XBRDR = (IPAPER-1.0-XSIZEI)/2.0
      YBRDR = (IPAPER-1.0-YSIZEI-8.0*CSIZEI)/2.0
C-----RE-ORIGIN THE PLOT SO (XMIN,YMIN) WILL BE (0,0,0,0)
      X0 = XSIZEI + XBRDR
      Y0 = YBRDR + 4.0*CSIZEI
C¬    CALL  PLT     ( SNGL(XBRDR),SNGL(Y0),-3 )
C;    CALL  PLOT    ( SNGL(XBRDR),SNGL(Y0),-3 )
C-----DRAW THE PLOT SCALE FACTOR MESSAGE AT THE BOTTOM OF THE PLOT
      XPAGE = XSIZEI/2.0 - 0.5*35*CSIZEI
      YPAGE = -3.0*CSIZEI
C¬          ENCODE  ( 35,601,JSCALE )            ISCALE
C;    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEI),ISCALE,0,0,35 )
C¬    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEI),JSCALE,0,0,35 )
      NLEFTD = DLOG10(SCALE) + 1.0D+00
      XPAGE = XPAGE + (16+0.5*(3-NLEFTD))*CSIZEI
      CALL  NUMBER  ( XPAGE,YPAGE,SNGL(CSIZEI),SNGL(SCALE),0,0,1 )
C-----DRAW THE TITLE FOR GEOPRO AT THE TOP OF THE PLOT
      XPAGE = XSIZEI/2.0 - 40.0*CSIZEI
      YPAGE = YSIZET + 2.0*CSIZEI
C¬          ENCODE  ( 80,601,JTITLE )            ITITLE
C¬    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEI),JTITLE,0,0,80 )
C;    CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(CSIZEI),ITITLE,0,0,80 )
C-----DRAW EACH INBOUND APPROACH
      DO 1060  KAN = 1 , NIBA
      KA = LIBA(KAN)
C   COLEASE,EXTRAC,APPRO,KA
      CALL  EXTRAC  (      1,KA     )                                   COLEASE
      IX1 = 0
C-----DRAW EACH LANE OF THE INBOUND APPROACH
      DO 1050  KLN = 1 , NLANES
      KL = LLANES(KLN)
C   COLEASE,EXTRAC,LANE,KL
      CALL  EXTRAC  (      4,KL     )                                   COLEASE
      IX2 = IX1 + LWID
                        IF ( LGEOM(1).NE.LGEOM(3) )  GO TO 1010
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(4) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(4) )
      GO TO 1040
 1010 CONTINUE
                        IF ( LGEOM(3).NE.LGEOM(4) )  GO TO 1030
 1020 CONTINUE
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(2) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(2) )
      GO TO 1040
 1030 CONTINUE
C-----DRAW A BOX FROM LGEOM(3) TO LGEOM(4) FOR THE INBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(3),LGEOM(4) )
                        IF ( LGEOM(1).NE.LGEOM(2) )  GO TO 1020
 1040 CONTINUE
      IX1 = IX2
C-----DRAW THE LANE TURN CODE ARROWS FOR THE INBOUND LANE
      IAL = IAAZIM - 90
      IAS = IAAZIM
      IAR = IAAZIM + 90
                        IF ( IAND(LTURN,8) , NE , 0 )CALL DRWUTA ( KL )
                        IF ( IAND(LTURN,4) , NE , 0 )CALL DRWARR ( IAL,KL )
                        IF ( IAND(LTURN,2) , NE , 0 )CALL DRWARR ( IAS,KL )
                        IF ( IAND(LTURN,1) , NE , 0 )CALL DRWARR ( IAR,KL )
C-----END OF LANE LOOP
 1050 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1060 CONTINUE
C-----DRAW EACH OUTBOUND APPROACH
      DO 2060  KAN = 1 , NOBA
      KA = LOBA(KAN)
C   COLEASE,EXTRAC,APPRO,KA
      CALL  EXTRAC  (      1,KA     )                                   COLEASE
      IX1 = 0
C-----DRAW EACH LANE OF THE OUTBOUND APPROACH
      DO 2050  KLN = 1 , NLANES
      KL = LLANES(KLN)
C   COLEASE,EXTRAC,LANE,KL
      CALL  EXTRAC  (      4,KL     )                                   COLEASE
      IX2 = IX1 + LWID
                        IF ( LGEOM(1).NE.LGEOM(3) )  GO TO 2010
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(4) FOR THE OUTBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(4) )
      GO TO 2040
 2010 CONTINUE
                        IF ( LGEOM(3).NE.LGEOM(4) )  GO TO 2030
 2020 CONTINUE
C-----DRAW A BOX FROM LGEOM(1) TO LGEOM(2) FOR THE OUTBOUND LANE
      CALL  DRWBOX  ( IX1,IX2,LGEOM(1),LGEOM(2) )
      GO TO 2040
 2030 CONTINUE
C-----DRAW A BOX FROM LGEOM(3) TO LGEOM(4) FOR THE OUTBOUND LANE
```

```
      CALL  DRWBOX   ( IX1,IX2,LGEOM(3),LGEOM(4) )
                     IF ( LGEOM(1).NE.LGEOM(2) )  GO TO 2020
 2040 CONTINUE
      IX1 = IX2
C-----DRAW THE LANE TURN CODE ARROWS FOR THE OUTBOUND LANE
      IAL = IAAZIM - 90
      IAS = IAAZIM
      IAR = IAAZIM + 90
                     IF ( IAND(LTURN,8) . NE . 0 )CALL DRWUTA ( KL )
                     IF ( IAND(LTURN,4) . NE . 0 )CALL DRWARR ( IAL,KL )
                     IF ( IAND(LTURN,2) . NE . 0 )CALL DRWARR ( IAS,KL )
                     IF ( IAND(LTURN,1) . NE . 0 )CALL DRWARR ( IAR,KL )
C-----END OF LANE LOOP
 2050 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2060 CONTINUE
                     IF ( NARCS . LE . 0 )          GO TO 3020
C-----DRAW EACH ARC
      DO 3010  IARCN = 1 , NARCS
      IARC = LARCS(IARCN)
C   COLEASE,EXTRAC,ARC,IARC
      CALL  EXTRAC  (      2,IARC   )                                     COLEASE
      IARCSW = IARCSW - 360
      CALL  DRWARC  ( IARCX,IARCY,IARCAZ,IARCSW,IARCR )
 3010 CONTINUE
 3020 CONTINUE
                     IF ( NLINES . LE . 0 )         GO TO 4020
C-----DRAW EACH LINE
      DO 4010  ILINEN = 1 , NLINES
      ILINE = LLINES(ILINEN)
C   COLEASE,EXTRAC,LINE,ILINE
      CALL  EXTRAC  (      5,ILINE )                                      COLEASE
      X1 = ILX1
      Y1 = ILY1
      X2 = ILX2
      Y2 = ILY2
      CALL  DRWLIN   ( X1,Y1,X2,Y2 )
 4010 CONTINUE
 4020 CONTINUE
                     IF ( NSDRC . LE . 0 )          GO TO 5020
C-----DRAW EACH SIGHT DISTANCE RESTRICTION COORDINATE
      DO 5010  ISDRCN = 1 , NSDRC
      ISDRC = LSDRC(ISDRCN)
      X = IXSDRC(ISDRC)
      Y = IYSDRC(ISDRC)
C-----IF THE COORDINATES LIE OFF THE PLOT PAGE THEN SKIP THE POINT
                     IF ( X . LT . XMIN )           GO TO 5010
                     IF ( X . GT . XMAX )           GO TO 5010
                     IF ( Y . LT . YMIN )           GO TO 5010
                     IF ( Y . GT . YMAX )           GO TO 5010
C-----DRAW A 5 FOOT STAR AT THE COORDINATE
      XPAGE = (X-XMIN)/SCALE
      YPAGE = (Y-YMIN)/SCALE
      CALL  SYMBOL  ( XPAGE,YPAGE,SNGL(5.0/SCALE),11,0.0,-1 )
 5010 CONTINUE
 5020 CONTINUE
      RETURN
      END                                                                DRWINT
```

```
      SUBROUTINE DRWUTA (ILANE)                                           COLEASE
C   TASK,DRWUTA,ILANE
      COMMON / APPRO  / IALEFT     ,IARGHT     ,NLANES     ,LLANES( 6),   COLEASE
     *                  IAPX       ,IAPY       ,ISLIM      ,NSDR      ,   COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM     ,NDEGST     ,    COLEASE
     *                  NDEGUT                                            COLEASE
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / DRWVAR(46),UX1,UX2,UX3,UX4,UX5,UX6,UY1,UY2,UY3,
     *                  UY4,UY5,UY6,ICX,ICY,ZTEMPD(33)
      DOUBLE PRECISION  UX1,UX2,UX3,UX4,UX5,UX6,UY1,UY2,UY3,UY4,UY5,UY6
      DOUBLE PRECISION  D1P5,D2P0,D2P5,D3P0
      DATA       D1P5    / 1.5D+00 /
      DATA       D2P0    / 2.0D+00 /
      DATA       D2P5    / 2.5D+00 /
      DATA       D3P0    / 3.0D+00 /
C
C-----SUBROUTINE DRWUTA DRAWS A U-TURN ARROW FOR A LANE
C
      ICX = LTDIRX(ILANE)
      ICY = LTDIRY(ILANE)
C-----FIND THE COORDINATES OF THE U-TURN ARROW
      CALL  XROTAX   ( D2P0,-D2P0,IAAZIM,ICX,ICY,UX1,UY1 )
      CALL  XROTAX   ( D2P0,D0P0,IAAZIM,ICX,ICY,UX2,UY2 )
      CALL  XROTAX   ( -D2P0,D0P0,IAAZIM,ICX,ICY,UX3,UY3 )
      CALL  XROTAX   ( -D2P0,-D3P0,IAAZIM,ICX,ICY,UX4,UY4 )
      CALL  XROTAX   ( -D2P5,-D2P0,IAAZIM,ICX,ICY,UX5,UY5 )
      CALL  XROTAX   ( -D1P5,-D2P0,IAAZIM,ICX,ICY,UX6,UY6 )
C-----DRAW A U-TURN ARROW FOR THE LANE
      CALL  DRWLIN   ( UX1,UY1,UX2,UY2 )
      CALL  DRWARC   ( ICX,ICY,IAAZIM+90,-180,2 )
      CALL  DRWLIN   ( UX3,UY3,UX4,UY4 )
      CALL  DRWLIN   ( UX4,UY4,UX5,UY5 )
      CALL  DRWLIN   ( UX4,UY4,UX6,UY6 )
      RETURN
      END                                                                DRWUTA
```

```
      SUBROUTINE DRWARR ( IANGLE,ILANE )
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / DRWVAR(46),XBOT,XLEFT,XRGHT,XTOP,YBOT,YLEFT,
     *                  YRGHT,YTOP,ICX,ICY,ZTEMPD(41)
      DOUBLE PRECISION  XBOT,XLEFT,XRGHT,XTOP,YBOT,YLEFT,YRGHT,YTOP
      DOUBLE PRECISION  D0P5,D2P5,D3P5
      DATA      D0P5    / 0.5D+00 /
      DATA      D2P5    / 2.5D+00 /
      DATA      D3P5    / 3.5D+00 /
C
C-----SUBROUTINE DRWARR DRAWS AN ARROW POINTING IN THE IANGLE DIRECTION
C
      ICX = LTDIRX(ILANE)
      ICY = LTDIRY(ILANE)
C-----FIND THE COORDINATES OF THE ARROW POINTING IN THE IANGLE DIRECTION
      CALL  XROTAX  ( D0P0,-D3P5,IANGLE,ICX,ICY,XBOT,YBOT )
      CALL  XROTAX  ( D0P0,D3P5,IANGLE,ICX,ICY,XTOP,YTOP )
      CALL  XROTAX  ( -D0P5,D2P5,IANGLE,ICX,ICY,XLEFT,YLEFT )
      CALL  XROTAX  ( D0P5,D2P5,IANGLE,ICX,ICY,XRGHT,YRGHT )
C-----DRAW THE ARROW POINTING IN THE IANGLE DIRECTION
      CALL  DRWLIN  ( XBOT,YBOT,XTOP,YTOP )
      CALL  DRWLIN  ( XTOP,YTOP,XLEFT,YLEFT )
      CALL  DRWLIN  ( XTOP,YTOP,XRGHT,YRGHT )
      RETURN
      END                                                               DRWARR
```

```
      SUBROUTINE FNDPTH                                                 COLEASE
C     TASK,FNDPTH
      COMMON / NOATTB / NOATTB( 7)                                      COLEASE
      COMMON / PATH   / IGEOCP(60),IXL   ( 2),IYL   ( 2),JXL   ( 2),    COLEASE
     *                  JYL   ( 2),IXA   ( 2),IYA   ( 2),LL1        ,   COLEASE
     *                  LA1         ,LA2         ,LL2         ,IIA  ,   COLEASE
     *                  IIL         ,IOA         ,IOL         ,IOPT ,   COLEASE
     *                  ILCH        ,IBA   ( 2),IDA   ( 2),IRA   ( 2),  COLEASE
     *                  IPTURN      ,LENP        ,LIBL        ,LOBL ,   COLEASE
     *                  LIMP        ,NGEOCP                             COLEASE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      DIMENSION         IENT6(1),MSG906(9)
      EQUIVALENCE       (IGEOCP(1),IENT6(1))
      DATA      MSG906  / 4H NUM,4HBER ,4HOF P,4HATHS,4H IS ,4HLE 0,
     *                  4H = F,4HNDPT,4HH     /
C
C-----SUBROUTINE FNDPTH FINDS THE INTERSECTION PATHS WITHIN THE
C-----INTERSECTION
C
      NUM = NOATTB(6)
      DO 1010  IZ = 1 , NUM
      IENT6(IZ) = 0
 1010 CONTINUE
C-----PROCESS EACH INBOUND APPROACH
      DO 2040  IAN = 1 , NIBA
                       IF ( IAN.EQ.1.OR.ISAME.EQ.2 )CALL DRWINT
      IA = LIBA(IAN)
C     COLEASE,FIND,JAZIM,APPRO,IA,IAAZIM
      CALL  FIND    (JAZIM ,      1,IA     ,      24)                   COLEASE
C     COLEASE,FIND,NLANEI,APPRO,IA,NLANES
      CALL  FIND    (NLANEI,      1,IA     ,      3)                    COLEASE
C-----PROCESS EACH LANE OF THE INBOUND APPROACH
      DO 2030  ILN = 1 , NLANEI
C     COLEASE,FIND,IL,APPRO,IA,LLANES(ILN)
      CALL  FIND    (IL    ,      1,IA     ,      3+ILN    )            COLEASE
C-----PROCESS EACH OUTBOUND APPROACH
      DO 2020  JAN = 1 , NOBA
      JA = LOBA(JAN)
C     COLEASE,FIND,KAZIM,APPRO,JA,IAAZIM
      CALL  FIND    (KAZIM ,      1,JA     ,      24)                   COLEASE
C     COLEASE,FIND,NLANEJ,APPRO,JA,NLANES
      CALL  FIND    (NLANEJ,      1,JA     ,      3)                    COLEASE
C-----PROCESS EACH LANE OF THE OUTBOUND APPROACH
      DO 2010  JLN = 1 , NLANEJ
C     COLEASE,FIND,JL,APPRO,JA,LLANES(JLN)
      CALL  FIND    (JL    ,      1,JA     ,      3+JLN    )            COLEASE
C-----CALCULATE AN INTERSECTION PATH WITHIN THE INTERSECTION AND CHECK
C-----ITS LEGALITY
      CALL  CALPTH
C-----IF THE PATH COULD NOT BE CALCULATED THEN GO TO THE NEXT OUTBOUND
C-----LANE
                       IF ( IFLAG , NE , 0 )          GO TO 2010
C-----IF THE PATH OPTION IS PRIMARY AND THE PATH OPTION CALCULATED FOR
C-----THE PATH IS NOT PRIMARY THEN GO TO THE NEXT OUTBOUND LANE
                 IF ( IPATH.EQ.1 . AND . JOPT.NE.0 )  GO TO 2010
C-----ADD THE INTERSECTION PATH FOR THE INBOUND LANE
      CALL  ADDPTH
                       IF ( IPLOT . EQ . 3 )          GO TO 2010
C-----DRAW THE INTERSECTION PATH OF THE PLOT PAGE
      CALL  DRWPTH
```

E M

```
C-----END OF OUTBOUND LANE LOOP
 2010 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 2020 CONTINUE
C-----END OF INBOUND LANE LOOP
 2030 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 2040 CONTINUE
                    IF ( NPATHS . LE . 0 )         GO TO 9060
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9060 CONTINUE
      CALL  ABORTR  ( MSG906,33 )
      STOP  906
      END                                                          FNDPTH
```

```
      SUBROUTINE CALPTH
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / IANGLE,ILNI,ILNO,ITURN,JTURN,KANGLE,LAZIM,LN,
     *                  LNI,LNJ,LNN,MAZIM,MTURN,NDEGST,NDEGUT,ZTEMPD(90)
      DIMENSION         MSG907(19),MSG908(19)
      DATA      MSG907 / 4H PAT,4HH TU,4HRN C,4HODE ,4HDOES,4H NOT,
     *                  4H MAT,4HCH A,4HNY T,4HURN ,4HCODE,4H FOR,
     *                  4H INB,4HOUND,4H APP,4HROAC,4HH - ,4HCALP,
     *                  4HATH  /
      DATA      MSG908 / 4H PAT,4HH TU,4HRN C,4HODE ,4HDOES,4H NOT,
     *                  4H MAT,4HCH A,4HNY T,4HURN ,4HCODE,4H FOR,
     *                  4H OUT,4HBOUN,4HD AP,4HPROA,4HCH -,4H CAL,
     *                  4HPTH  /
C
C-----SUBROUTINE CALPTH CALCULATES AN INTERSECTION PATH WITHIN THE
C-----INTERSECTION AND CHECKS ITS LEGALITY
C
      IFLAG = 1
C-----IF THE INBOUND LANE IS NOT AVAILABLE AT THE INTERSECTION THEN
C-----RETURN WITH IFLAG EQUAL 1
                    IF ( IXAPP(IL) . LT . 0 )     RETURN
                    IF ( IYAPP(IL) . LT . 0 )     RETURN
C-----IF THE OUTBOUND LANE IS NOT AVAILABLE AT THE INTERSECTION THEN
C-----RETURN WITH IFLAG EQUAL 1
                    IF ( IXAPP(JL) . LT . 0 )     RETURN
                    IF ( IYAPP(JL) . LT . 0 )     RETURN
      IFLAG = 0
C-----ROTATE THE COORDINATES OF THE INBOUND LANE AND THE OUTBOUND LANE
C-----SO THAT THE INBOUND LANE IS POINTING NORTH (0 AZIMUTH)
      CALL  IROTX   ( IXAPP(IL),IYAPP(IL),-JAZIM,XI,YI )
      CALL  IROTX   ( IXAPP(JL),IYAPP(JL),-JAZIM,XO,YO )
C-----FIND THE PARAMETERS FOR CALCULATING THE INTERSECTION PATH
      ADX = DABS( XI-XO )
      ADY = DABS( YI-YO )
C   COLEASE,FIND,NDEGST,APPRO,IA,NDEGST
      CALL  FIND    (NDEGST,     1,IA    ,     25)                  COLEASE
C   COLEASE,FIND,NDEGUT,APPRO,IA,NDEGUT
      CALL  FIND    (NDEGUT,     1,IA    ,     26)                  COLEASE
C   COLEASE,FIND,ITURN,LANE,IL,LTURN
      CALL  FIND    (ITURN ,     4,IL    ,     13)                  COLEASE
C   COLEASE,FIND,JTURN,LANE,JL,LTURN
      CALL  FIND    (JTURN ,     4,JL    ,     13)                  COLEASE
      LAZIM = JAZIM + 180
      MAZIM = KAZIM
                    IF ( LAZIM . GE . 360 )       LAZIM = LAZIM - 360
                    IF ( MAZIM . LT . LAZIM )     MAZIM = MAZIM + 360
      IANGLE = MAZIM - LAZIM
                    IF ( IANGLE . LT . 180 )      JANGLE = 180 - IANGLE
                    IF ( IANGLE . GE . 180 )      JANGLE = IANGLE - 180
                    IF ( JANGLE . EQ .   0 )      GO TO 1010
                    IF ( JANGLE . EQ . 180 )      GO TO 1020
                    IF ( XO - XI )                2010 , 2010 , 3010
 1010 CONTINUE
C-----CALCULATE A STRAIGHT PATH
      KTURN = 2
             IF ( XO.LT.XI . AND . ADX.GT.ZERO )  CALL  STRLFT
             IF ( XO.EQ.XI .  OR . ADX.LE.ZERO )  CALL  STRSTR
             IF ( XO.GT.XI . AND . ADX.GT.ZERO )  CALL  STRRGH
                    IF ( RA2 . GT . RADIUS )      CALL  STRSTR
      GO TO 4010
```

```
 1020 CONTINUE
C-----CALCULATE A U-TURN PATH
      KTURN = 8
                    IF ( XI . GE . XO )            CALL  UTURNL
                    IF ( XI . LT . XO )            CALL  UTURNR
      GO TO 4010
 2010 CONTINUE
C-----CALCULATE A LEFT TURN PATH
      KTURN = 4
                    IF ( JANGLE - 90 )             2020 , 2030 , 2030
 2020 CONTINUE
C-----LEFT TURN IS LESS THAN 90 DEGREES
                    IF ( JANGLE . LE . NDEGST )  KTURN = 2
      RC = ADX / ( 1.0-DCOS(JANGLE*RADIAN) )
      YC = RC*DSIN(JANGLE*RADIAN)
                    IF ( ADY . GE . YC )           CALL  LTLTGE
                    IF ( ADY . LT . YC )           CALL  LTLTLT
                    IF ( RA2 . GT . RADIUS )       CALL  STRSTR
      GO TO 4010
 2030 CONTINUE
C-----LEFT TURN IS GREATER THAN OR EQUAL 90 DEGREES
                    IF ( JANGLE.GE.180-NDEGUT )  KTURN = 8
      KANGLE = 180 - JANGLE
      RC = ADX / ( 1.0+DCOS(KANGLE*RADIAN) )
      YC = RC*DSIN(KANGLE*RADIAN)
             IF ( ADY.GE.YC . AND . YO.GE.YI )     CALL  LTGEGE
             IF ( ADY.LT.YC .  OR . YO.LT.YI )     CALL  LTGELT
      GO TO 4010
 3010 CONTINUE
C-----CALCULATE A RIGHT TURN PATH
      KTURN = 1
                    IF ( JANGLE - 90 )             3020 , 3030 , 3030
 3020 CONTINUE
C-----RIGHT TURN IS LESS THAN 90 DEGREES
                    IF ( JANGLE . LE . NDEGST )  KTURN = 2
      RC = ADX / ( 1.0-DCOS(JANGLE*RADIAN) )
      YC = RC*DSIN(JANGLE*RADIAN)
                    IF ( ADY . GE . YC )           CALL  RTLTGE
                    IF ( ADY . LT . YC )           CALL  RTLTLT
                    IF ( RA2 . GT . RADIUS )       CALL  STRSTR
      GO TO 4010
 3030 CONTINUE
C-----RIGHT TURN IS GREATER THAN OR EQUAL TO 90 DEGREES
                    IF ( JANGLE.GE.180-NDEGUT )  KTURN = 8
      KANGLE = 180 - JANGLE
      RC = ADX / ( 1.0+DCOS(KANGLE*RADIAN) )
      YC = RC*DSIN(KANGLE*RADIAN)
             IF ( ADY.GE.YC . AND . YO.GE.YI )     CALL  RTGEGE
             IF ( ADY.LT.YC .  OR . YO.LT.YI )     CALL  RTGELT
 4010 CONTINUE
C-----IF THE INTERSECTION PATH COULD NOT BE CALCULATED THEN RETURN
                    IF ( IFLAG . NE . 0 )          RETURN
C-----IF THE TURN CODE OF THE PATH DOES NOT MATCH THE TURN CODE OF THE
C-----INBOUND LANE AND THE OUTBOUND LANE THEN RETURN WITH IFLAG EQUAL 1
                    IF ( IAND(ITURN,KTURN).EQ.0 )IFLAG = 1
                    IF ( IAND(JTURN,KTURN).EQ.0 )IFLAG = 1
                    IF ( IFLAG . NE . 0 )          RETURN
C-----CHECK THE LANE CHANGE OPTION AND THE PATH OPTION
      JOPT = 0
      JLCH = 0
C-----IF THE PATH IS A U-TURN THEN RETURN AND DO NOT CHECK THE LANE
C-----CHANGE OPTION OR THE PATH OPTION
                    IF ( KTURN . EQ . 8 )          RETURN
C-----IF THE PATH IS A RIGHT TURN THEN GO TO 4060
                    IF ( KTURN . EQ . 1 )          GO TO 4060
C-----FIND THE LANE NUMBER OF THE FIRST INBOUND LANE WITH A TURN CODE
C-----THAT MATCHES THE TURN CODE FOR THE PATH (INSIDE TO OUTSIDE)
      DO 4020  LNI = 1 , NLANEI
C     COLEASE,FIND,LN,APPRO,IA,LLANES(LNI)
      CALL  FIND     (LN     ,       1,IA     ,       3+LNI    )            COLEASE
C     COLEASE,FIND,MTURN,LANE,LN,LTURN
      CALL  FIND     (MTURN ,       4,LN     ,       13)                    COLEASE
                    IF ( IAND(MTURN,KTURN).NE.0 )GO TO 4030
 4020 CONTINUE
      GO TO 9070
 4030 CONTINUE
C-----FIND THE LANE NUMBER OF THE FIRST OUTBOUND LANE WITH A TURN CODE
C-----THAT MATCHES THE TURN CODE FOR THE PATH (INSIDE TO OUTSIDE)
      DO 4040  LNJ = 1 , NLANEJ
C     COLEASE,FIND,LN,APPRO,JA,LLANES(LNJ)
      CALL  FIND     (LN     ,       1,JA     ,       3+LNJ    )            COLEASE
C     COLEASE,FIND,MTURN,LANE,LN,LTURN
      CALL  FIND     (MTURN ,       4,LN     ,       13)                    COLEASE
                    IF ( IAND(MTURN,KTURN).NE.0 )GO TO 4050
 4040 CONTINUE
      GO TO 9080
 4050 CONTINUE
C-----IF NOT THE SAME RELATIVE LANE NUMBER THEN THERE IS A LANE CHANGE
      ILNI = ILN - LNI
      ILNO = JLN - LNJ
                    IF ( ILNO . NE . ILNI )       JLCH = 1
C-----IF LANE 1 OF THE INBOUND APPROACH THEN GO TO 5010 AND CHECK THE
C-----PATH OPTION
                    IF ( ILN . EQ . 1 )           GO TO 5010
C-----IF NOT THE LAST LANE OF THE INBOUND APPROACH THEN GO TO 5010 AND
C-----CHECK THE PATH OPTION
                    IF ( ILN . NE . NLANEI )      GO TO 5010
 4060 CONTINUE
C-----FIND THE LANE NUMBER OF THE FIRST INBOUND LANE WITH A TURN CODE
C-----THAT MATCHES THE TURN CODE FOR THE PATH (OUTSIDE TO INSIDE)
      DO 4070  LNN = 1 , NLANEI
      LNI = NLANEI - LNN + 1
C     COLEASE,FIND,LN,APPRO,IA,LLANES(LNI)
      CALL  FIND     (LN     ,       1,IA     ,       3+LNI    )            COLEASE
C     COLEASE,FIND,MTURN,LANE,LN,LTURN
      CALL  FIND     (MTURN ,       4,LN     ,       13)                    COLEASE
                    IF ( IAND(MTURN,KTURN).NE.0 )GO TO 4080
 4070 CONTINUE
      GO TO 9070
 4080 CONTINUE
C-----FIND THE LANE NUMBER OF THE FIRST OUTBOUND LANE WITH A TURN CODE
C-----THAT MATCHES THE TURN CODE FOR THE PATH (OUTSIDE TO INSIDE)
      DO 4090  LNN = 1 , NLANEJ
      LNJ = NLANEJ - LNN + 1
C     COLEASE,FIND,LN,APPRO,JA,LLANES(LNJ)
      CALL  FIND     (LN     ,       1,JA     ,       3+LNJ    )            COLEASE
C     COLEASE,FIND,MTURN,LANE,LN,LTURN
      CALL  FIND     (MTURN ,       4,LN     ,       13)                    COLEASE
                    IF ( IAND(MTURN,KTURN).NE.0 )GO TO 4100
 4090 CONTINUE
      GO TO 9080
 4100 CONTINUE
C-----IF NOT THE SAME RELATIVE LANE NUMBER THEN THERE IS A LANE CHANGE
      ILNI = ILN - LNI
      ILNO = JLN - LNJ
                    IF ( ILNO . NE . ILNI )       JLCH = 1
 5010 CONTINUE
C-----IF NOT THE SAME RELATIVE LANE NUMBER THEN THE PATH IS OPTION1
                    IF ( ILNO . NE . ILNI )       JOPT = 1
C-----IF MORE THAN 1 LANE CHANGED THEN THE PATH IS ILLEGAL
                    IF ( ILNO . LT . ILNI-1 )     IFLAG = 1
                    IF ( ILNO . GT . ILNI+1 )     IFLAG = 1
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9070 CONTINUE
      CALL  ABORTR   ( MSG907,74 )
      STOP  907
 9080 CONTINUE
      CALL  ABORTR   ( MSG908,75 )
      STOP  908
      END                                                                   CALPTH
```

```
      SUBROUTINE STRLFT
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      COMMON / ZTEMPD / CALPTH(16),ANGLE,ZTEMPD(87)
      DOUBLE PRECISION  ANGLE
C
C-----SUBROUTINE STRLFT CALCULATES AN INTERSECTION PATH THAT IS A
C-----STRAIGHT THROUGH MOVEMENT THAT IS A REVERSE CIRCULAR CURVE THAT
C-----VEERS LEFT (EXACTLY Ø DEGREES)
C
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE SECTION 2 (ARC 1) AS A REVERSE CIRCULAR CURVE
      RA2 = ( ADX**2+ADY**2 )/( 4.Ø*ADX )
      XC2 = XI - RA2
      YC2 = YI
      ANGLE = DATAN(ADY/(2.Ø*RA2-ADX)) / RADIAN
      JANGLE = DMAX1( 1.ØD+ØØ,ANGLE+XROUND )
      L2 = ANGLE*RA2*RADIAN + XROUND
      JB2 = 9Ø
      JD2 = -JANGLE
C-----CALCULATE SECTION 3 (ARC 2) AS A REVERSE CIRCULAR CURVE
      RA3 = RA2
      XC3 = XO + RA3
      YC3 = YO
      L3 = L2
      JB3 = 27Ø - JANGLE
      JD3 = JANGLE
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
      RETURN
      END                                                               STRLFT
```

```
      SUBROUTINE STRSTR
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      COMMON / ZTEMPD / CALPTH(16),ZTEMPD(89)
C
C-----SUBROUTINE STRSTR CALCULATES AN INTERSECTION PATH THAT IS A
C-----STRAIGHT THROUGH MOVEMENT THAT GOES STRAIGHT FROM THE INBOUND LANE
C-----TO THE OUTBOUND LANE
C
C-----CALCULATE SECTION 1 (LINE 1) FROM THE INBOUND LANE TO THE OUTBOUND
C-----LANE
      X11 = XI
      Y11 = YI
      L1 = DSQRT(ADX**2+ADY**2) + XROUND
      X12 = XO
      Y12 = YO
C-----SECTION 2 (ARC 1) IS NOT USED
      CALL  ZEROP2
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----SET A HIGH MAXIMUM SPEED FOR THE INTERSECTION PATH SO THAT THE
C-----SPEED LIMIT OF THE INBOUND AND THE OUTBOUND APPROACH WILL GOVERN
      JSPEED = 999
      RETURN
      END                                                               STRSTR
```

```
      SUBROUTINE STRRGH
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),ANGLE,ZTEMPD(87)
      DOUBLE PRECISION  ANGLE
C
C-----SUBROUTINE STRRGH CALCULATES AN INTERSECTION PATH AS A STRAIGHT
C-----STRAIGHT THROUGH MOVEMENT THAT IS A REVERSE CIRCULAR CURVE THAT
C-----VEERS RIGHT (EXACTLY 0 DEGREES)
C
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE SECTION 2 (ARC 1) AS A REVERSE CIRCULAR CURVE
      RA2 = ( ADX**2+ADY**2 )/( 4.0*ADX )
      XC2 = XI + RA2
      YC2 = YI
      ANGLE = DATAN(ADY/(2.0*RA2-ADX)) / RADIAN
      JANGLE = DMAX1( 1.0D+00,ANGLE+XROUND )
      L2 = ANGLE*RA2*RADIAN + XROUND
      JB2 = 270
      JD2 = JANGLE
C-----CALCULATE SECTION 3 (ARC 2) AS A REVERSE CIRCULAR CURVE
      RA3 = RA2
      XC3 = XO - RA3
      YC3 = YO
      L3 = L2
      JB3 = 90 + JANGLE
      JD3 = -JANGLE
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
      RETURN
      END                                                         STRRGH
```

```
      SUBROUTINE UTURNL
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),ZTEMPD(89)
C
C-----SUBROUTINE UTURNL CALCULATES AN INTERSECTION PATH THAT IS A U-TURN
C-----THAT GOES LEFT (EXACTLY 180 DEGREES)
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      L1 = ADY + XROUND
      X12 = XI
      Y12 = YI + ADY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE END OF SECTION 1
C-----(LINE 1) TO THE START OF SECTION 4 (LINE 2)
      RA2 = ADX / 2.0
      XC2 = XI - RA2
      YC2 = YI
                      IF ( YO . GT . YI )           YC2 = YO
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----(ARC 1) TO THE OUTBOUND LANE
      X41 = XO
      Y41 = YO + ADY
      L4 = ADY + XROUND
      X42 = XO
      Y42 = YO
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE INBOUND LANE IS ABOVE THE OUTBOUND LANE THEN SECTION 1
C-----(LINE 1) IS NOT USED
                      IF ( YI . GE . YO )           CALL  ZEROP1
C-----IF THE OUTBOUND LANE IS ABOVE THE INBOUND LANE THEN SECTION 4
C-----(LINE 2) IS NOT USED
                      IF ( YO . GE . YI )           CALL  ZEROP4
      RETURN
      END                                                         UTURNL
```

```
      SUBROUTINE UTURNR
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),ZTEMPD(89)
C
C-----SUBROUTINE UTURNR CALCULATES AN INTERSECTION PATH THAT IS A U-TURN
C-----THAT GOES RIGHT (EXACTLY 180 DEGREES)
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      L1 = ADY + XROUND
      X12 = XI
      Y12 = YI + ADY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE END OF SECTION 1
C-----(LINE 1) TO THE START OF SECTION 4 (LINE 4)
      RA2 = ADX / 2.0
      XC2 = XI + RA2
      YC2 = YI
                      IF ( YO . GT . YI )           YC2 = YO
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 270
      JD2 = JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----(ARC 1) TO THE OUTBOUND LANE
      X41 = XO
      Y41 = YO + ADY
      L4 = ADY + XROUND
      X42 = XO
      Y42 = YO
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE INBOUND LANE IS ABOVE THE OUTBOUND LANE THEN SECTION 1
C-----(LINE 1) IS NOT USED
                      IF ( YI . GE . YO )           CALL  ZEROP1
C-----IF THE OUTBOUND LANE IS ABOVE THE INBOUND LANE THEN SECTION 4
C-----(LINE 2) IS NOT USED
                      IF ( YO . GE . YI )           CALL  ZEROP4
      RETURN
      END                                                               UTURNR
```

```
      SUBROUTINE LTLTGE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),DY,ZTEMPD(87)
      DOUBLE PRECISION  DY
C
C-----SUBROUTINE LTLTGE CALCULATES AN INTERSECTION PATH THAT IS A LEFT
C-----TURN LT 90 DEGREES AND ADY GE YC WITH RADIUS RC
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      DY = ADY - YC
      L1 = DY + XROUND
      X12 = XI
      Y12 = YI + DY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC WITH RADIUS RC FROM THE END
C-----OF SECTION 1 (LINE 1) TO THE OUTBOUND LANE
      RA2 = RC
      XC2 = XI - RA2
      YC2 = YI + DY
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 1 (LINE 1) IS LE 0 THEN SECTION 1 IS NOT
C-----USED
                      IF ( L1 . LE . 0 )            CALL  ZEROP1
      RETURN
      END                                                               LTLTGE
```

```
      SUBROUTINE LTLTLT
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION   XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN  / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION   PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD  / CALPTH(16),A,ANGLE2,ANGLE3,B,C,COSJA,DY,RADICL,
     *                   SINJA,KANGLE,KANGL2,KANGL3,ZTEMPD(68)
      DOUBLE PRECISION   A,ANGLE2,ANGLE3,B,C,COSJA,DY,RADICL,SINJA
      DOUBLE PRECISION   DTAN
C
C-----SUBROUTINE LTLTLT CALCULATES AN INTERSECTION PATH THAT IS A LEFT
C-----TURN LT 90 DEGREES AND ADY LT YC
C
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----(ARC 1) TO THE START OF THE OUTBOUND LANE
      X42 = XO
      Y42 = YO
      DY = YC - ADY
      L4 = DY + XROUND
      KANGLE = 90 - JANGLE
      X41 = XO + DY*DCOS(KANGLE*RADIAN)
      Y41 = YO - DY*DSIN(KANGLE*RADIAN)
C-----IF THE START OF SECTION 4 (LINE 2) IS TO THE RIGHT OR BELOW THE
C-----INBOUND LANE THEN GO TO 1010 AND CALCULATE A REVERSE CURVE
                      IF ( X41 . GE . XI )         GO TO 1010
                      IF ( Y41 . LE . YI )         GO TO 1010
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE INBOUND LANE TO THE
C-----START OF SECTION 4 (LINE 2)
      RA2 = XI-X41 + (Y41-YI)/DTAN(JANGLE*RADIAN)
      XC2 = XI - RA2
      YC2 = YI
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -JANGLE
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 4 (LINE 2) IS LE 0 THEN SECTION 4 IS NOT
C-----USED
                      IF ( L4 . LE . 0 )           CALL  ZEROP4
      RETURN
 1010 CONTINUE
C-----CALCULATE A REVERSE CURVE
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE INBOUND LANE TO THE
C-----START OF SECTION 3 (ARC 2)
      SINJA = DSIN(JANGLE*RADIAN)
      COSJA = DCOS(JANGLE*RADIAN)
      A = 2.0 - 2.0*COSJA
      B = 2.0*ADX*(1.0+COSJA) - 2.0*ADY*SINJA
      C = ADX**2 + ADY**2
      C = -C
      RADICL = B**2 - 4.0*A*C
C-----IF RADICL IS LT 0.0 THEN THE PATH CAN NOT BE CALCULATED
                      IF ( RADICL . LT . D0P0 )    GO TO 2010
      RA2 = (-B+DSQRT(RADICL))/(2.0*A)
      XC2 = XI - RA2
      YC2 = YI
      ANGLE2 = DATAN((RA2*SINJA+ADY)/(RA2+RA2*COSJA-ADX))/RADIAN
      KANGL2 = DMAX1( 1.0D+00,ANGLE2+XROUND )
      L2 = ANGLE2*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -KANGL2
C-----CALCULATE SECTION 3 (ARC 2) AS AN ARC FROM THE END OF SECTION 2
C-----(ARC 2) TO THE OUTBOUND LANE
      RA3 = RA2
      XC3 = XO + RA3*COSJA
      YC3 = YO + RA3*SINJA
      ANGLE3 = ANGLE2 - JANGLE
      KANGL3 = DMAX1( 1.0D+00,ANGLE3+XROUND )
      L3 = ANGLE3*RA3*RADIAN + XROUND
      JB3 = 270 - JANGLE - KANGL3
      JD3 = KANGL3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
      RETURN
 2010 CONTINUE
      IFLAG = 1
      RETURN
      END
```

419

```
      SUBROUTINE LTGEGE
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION   XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),DY,ZTEMPD(87)
      DOUBLE PRECISION  DY
C
C-----SUBROUTINE LTGEGE CALCULATES AN INTERSECTION PATH THAT IS A LEFT
C-----TURN GE 90 DEGREES AND ADY GE YC WITH RADIUS RC
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      DY = ADY - YC
      L1 = DY + XROUND
      X12 = XI
      Y12 = YI + DY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC WITH RADIUS RC FROM THE END
C-----OF SECTION 1 (LINE 1) TO THE OUTBOUND LANE
      RA2 = RC
      XC2 = XI - RA2
      YC2 = YI + DY
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 1 (LINE 1) IS LE 0 THEN SECTION 1 IS NOT
C-----USED
                      IF ( L1 . LE . 0 )           CALL  ZEROP1
      RETURN
      END                                                            LTGEGE
```

```
      SUBROUTINE LTGELT
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION   XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),DY,KANGLE,ZTEMPD(86)
      DOUBLE PRECISION  DY
      DOUBLE PRECISION  DTAN
C
C-----SUBROUTINE LTGELT CALCULATES AN INTERSECTION PATH THAT IS A LEFT
C-----TURN GE 90 DEGREES AND ADY LT YC
C
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----TO THE OUTBOUND LANE
      X42 = XO
      Y42 = YO
      DY = YI + YC - YO
      L4 = DY + XROUND
      KANGLE = JANGLE - 90
      X41 = XO + DY*DCOS(KANGLE*RADIAN)
      Y41 = YO + DY*DSIN(KANGLE*RADIAN)
C-----IF THE START OF SECTION 4 (LINE 2) IS TO THE RIGHT OR BELOW THE
C-----INBOUND LANE THEN THE PATH CAN NOT BE CALCULATED
                      IF ( X41 . GE . XI )         GO TO 2010
                      IF ( Y41 . LE . YI )         GO TO 2010
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE INBOUND LANE TO THE
C-----START OF SECTION 4 (LINE 2)
      RA2 = XI - X41
                      IF ( JANGLE . EQ . 90 )      GO TO 1010
      KANGLE = 180 - JANGLE
      RA2 = RA2 - (Y41-YI)/DTAN(KANGLE*RADIAN)
 1010 CONTINUE
      XC2 = XI - RA2
      YC2 = YI
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 90
      JD2 = -JANGLE
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 4 (LINE 2) IS LE 0 THEN SECTION 4 IS NOT
C-----USED
                      IF ( L4 . LE . 0 )           CALL  ZEROP4
      RETURN
 2010 CONTINUE
      IFLAG = 1
      RETURN
      END
```

```
      SUBROUTINE RTLTGE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      COMMON / ZTEMPD / CALPTH(16),DY,ZTEMPD(87)
      DOUBLE PRECISION  DY
C
C-----SUBROUTINE RTLTGE CALCULATES AN INTERSECTION PATH THAT IS A RIGHT
C-----TURN LT 9Ø DEGREES AND ADY GE YC WITH RADIUS RC
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      DY = ADY - YC
      L1 = DY + XROUND
      X12 = XI
      Y12 = YI + DY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC WITH RADIUS RC FROM THE END
C-----OF SECTION 1 TO THE OUTBOUND LANE
      RA2 = RC
      XC2 = XI + RA2
      YC2 = YI + DY
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 27Ø
      JD2 = JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 1 (LINE 1) IS LE Ø THEN SECTION 1 IS NOT
C-----USED
                      IF ( L1 . LE . Ø )            CALL  ZEROP1
      RETURN
      END                                                         RTLTGE
```

```
      SUBROUTINE RTLTLT
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      COMMON / ZTEMPD / CALPTH(16),A,ANGLE2,ANGLE3,B,C,COSJA,DY,RADICL,
     *                  SINJA,KANGLE,KANGL2,KANGL3,ZTEMPD(68)
      DOUBLE PRECISION  A,ANGLE2,ANGLE3,B,C,COSJA,DY,RADICL,SINJA
      DOUBLE PRECISION  DTAN
C
C-----SUBROUTINE RTLTLT CALCULATES AN INTERSECTION PATH THAT IS A RIGHT
C-----TURN LT 9Ø DEGREES AND ADY LT YC
C
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----(ARC 1) TO THE OUTBOUND LANE
      X42 = XO
      Y42 = YO
      DY = YC - ADY
      L4 = DY + XROUND
      KANGLE = 9Ø - JANGLE
      X41 = XO - DY*DCOS(KANGLE*RADIAN)
      Y41 = YO - DY*DSIN(KANGLE*RADIAN)
C-----IF THE START OF SECTION 4 (LINE 2) IS TO THE LEFT OR BELOW THE
C-----INBOUND LANE THEN GO TO 1Ø1Ø AND CALCULATE REVERSE CURVES
                      IF ( X41 . LE . XI )          GO TO 1Ø1Ø
                      IF ( Y41 . LE . YI )          GO TO 1Ø1Ø
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE INBOUND LANE TO THE
C-----START OF SECTION 4 (LINE 2)
      RA2 = X41 - XI + (Y41-YI)/DTAN(JANGLE*RADIAN)
      XC2 = XI + RA2
      YC2 = YI
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 27Ø
      JD2 = JANGLE
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 4 (LINE 2) IS LE Ø THEN SECTION 4 IS NOT
C-----USED
                      IF ( L4 . LE . Ø )            CALL  ZEROP4
      RETURN
 1Ø1Ø CONTINUE
C-----CALCULATE REVERSE CURVES
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE SECTION 2 (ARC 1) AS A REVERSE CURVE FROM THE INBOUND
C-----LANE TO THE START OF SECTION 3 (ARC 2)
      SINJA = DSIN(JANGLE*RADIAN)
      COSJA = DCOS(JANGLE*RADIAN)
      A = 2.Ø - 2.Ø*COSJA
      B = 2.Ø*ADX*(1.Ø+COSJA) - 2.Ø*ADY*SINJA
      C = ADX**2 + ADY**2
      C = -C
      RADICL = B**2 - 4.Ø*A*C
C-----IF RADICL LT Ø.Ø THEN THE REVERSE CURVE CAN NOT BE CALCULATED
                      IF ( RADICL . LT . DØPØ )     GO TO 2Ø1Ø
      RA2 = (-B+DSQRT(RADICL))/(2.Ø*A)
      XC2 = XI + RA2
      YC2 = YI
      ANGLE2 = DATAN((RA2*SINJA+ADY)/(RA2+RA2*COSJA-ADX))/RADIAN
      KANGL2 = DMAX1( 1.ØD+ØØ,ANGLE2+XROUND )
      L2 = ANGLE2*RA2*RADIAN + XROUND
      JB2 = 27Ø
```

```
      JD2 = KANGL2
C-----CALCULATE SECTION 3 (ARC 2) AS A REVERSE CURVE FROM THE END OF
C-----SECTION 2 (ARC 1) TO THE OUTBOUND LANE
      RA3 = RA2
      XC3 = XO - RA3*COSJA
      YC3 = YO + RA3*SINJA
      ANGLE3 = ANGLE2 - JANGLE
      KANGL3 = DMAX1( 1,ØD+ØØ,ANGLE3+XROUND )
      L3 = ANGLE3*RA3*RADIAN + XROUND
      JB3 = 9Ø + JANGLE + KANGL3
      JD3 = -KANGL3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
      RETURN
 2Ø1Ø CONTINUE
      IFLAG = 1
      RETURN
      END                                                              RTLTLT
```

```
      SUBROUTINE RTGEGE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,DØPØ
      COMMON / ZTEMPD / CALPTH(16),DY,ZTEMPD(87)
      DOUBLE PRECISION  DY
C
C-----SUBROUTINE RTGEGE CALCULATES AN INTERSECTION PATH THAT IS A RIGHT
C-----TURN GE 9Ø DEGREES AND ADY GE YC WITH RADIUS RC
C
C-----CALCULATE SECTION 1 (LINE 1) AS A LINE FROM THE INBOUND LANE TO
C-----THE START OF SECTION 2 (ARC 1)
      X11 = XI
      Y11 = YI
      DY = ADY - YC
      L1 = DY + XROUND
      X12 = XI
      Y12 = YI + DY
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC WITH RADIUS RC FROM THE END
C-----OF SECTION 1 (LINE 1) TO THE OUTBOUND LANE
      RA2 = RC
      XC2 = XI + RA2
      YC2 = YI + DY
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 27Ø
      JD2 = JANGLE
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----SECTION 4 (LINE 2) IS NOT USED
      CALL  ZEROP4
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 1 (LINE 1) IS LE Ø THEN SECTION 1 IS NOT
C-----USED
                    IF ( L1 . LE . Ø )            CALL  ZEROP1
      RETURN
      END                                                              RTGEGE
```

```
      SUBROUTINE RTGELT
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALPTH(16),DY,KANGLE,ZTEMPD(86)
      DOUBLE PRECISION  DY
      DOUBLE PRECISION  DTAN
C
C-----SUBROUTINE RTGELT CALCULATES AN INTERSECTION PATH THAT IS A RIGHT
C-----TURN GE 90 DEGREES AND ADY LT YC
C
C-----CALCULATE SECTION 4 (LINE 2) AS A LINE FROM THE END OF SECTION 2
C-----(ARC 1) TO THE OUTBOUND LANE
      X42 = XO
      Y42 = YO
      DY = YI + YC - YO
      L4 = DY + XROUND
      KANGLE = JANGLE - 90
      X41 = XO - DY*DCOS(KANGLE*RADIAN)
      Y41 = YO + DY*DSIN(KANGLE*RADIAN)
C-----IF THE START OF SECTION 4 (LINE 2) IS TO THE LEFT OR BELOW THE
C-----INBOUND LANE THEN THE PATH CAN NOT BE CALCULATED
                 IF ( X41 . LE . XI )          GO TO 2010
                 IF ( Y41 . LE . YI )          GO TO 2010
C-----SECTION 3 (ARC 2) IS NOT USED
      CALL  ZEROP3
C-----CALCULATE SECTION 2 (ARC 1) AS AN ARC FROM THE INBOUND LANE TO THE
C-----START OF SECTION 4 (LINE 2)
      RA2 = X41 - XI
                 IF ( JANGLE . EQ . 90 )       GO TO 1010
      KANGLE = 180 - JANGLE
      RA2 = RA2 - (Y41-YI)/DTAN(KANGLE*RADIAN)
 1010 CONTINUE
      XC2 = XI + RA2
      YC2 = YI
      L2 = JANGLE*RA2*RADIAN + XROUND
      JB2 = 270
      JD2 = JANGLE
C-----SECTION 1 (LINE 1) IS NOT USED
      CALL  ZEROP1
C-----CALCULATE THE MAXIMUM VELOCITY FOR THE INTERSECTION PATH BASED ON
C-----THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE PATH
      JSPEED = MAXVEL( RA2 )
C-----IF THE LENGTH OF SECTION 4 (LINE 2) IS LE 0 THEN SECTION 4 IS NOT
C-----USED
                 IF ( L4 . LE . 0 )            CALL  ZEROP4
      RETURN
 2010 CONTINUE
      IFLAG = 1
      RETURN
      END                                                          RTGELT
```

```
      SUBROUTINE ZEROP1
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALVAL(38),ZTEMPD(67)
C
C-----SUBROUTINE ZEROP1 ZEROES OUT THE PARAMETERS FOR SECTION 1 OF THE
C-----INTERSECTION PATH (LINE 1)
C
      X11 = D0P0
      Y11 = D0P0
      L1 = 0
      X12 = D0P0
      Y12 = D0P0
      RETURN
      END                                                          ZEROP1
```

```
      SUBROUTINE ZEROP2
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALVAL(38),ZTEMPD(67)
C
C-----SUBROUTINE ZEROP2 ZEROES OUT THE PARAMETERS FOR SECTION 2 OF THE
C-----INTERSECTION PATH (ARC 1)
C
      XC2 = D0P0
      YC2 = D0P0
      RA2 = D0P0
      L2 = 0
      JB2 = 0
      JD2 = 0
      RETURN
      END                                                         ZEROP2
```

```
      SUBROUTINE ZEROP3
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CALVAL(38),ZTEMPD(67)
C
C-----SUBROUTINE ZEROP3 ZEROES OUT THE PARAMETERS FOR SECTION 3 OF THE
C-----INTERSECTION PATH (ARC 2)
C
      XC3 = D0P0
      YC3 = D0P0
      RA3 = D0P0
      L3 = 0
      JB3 = 0
      JD3 = 0
      RETURN
      END
```

```
      SUBROUTINE ZEROP4
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION   XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN /  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION   PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD /  CALVAL(38),ZTEMPD(67)
C
C-----SUBROUTINE ZEROP4 ZEROES OUT THE PARAMETERS FOR SECTION 4 OF THE
C-----INTERSECTION PATH (LINE 2)
C
      X41 = D0P0
      Y41 = D0P0
      L4 = 0
      X42 = D0P0
      Y42 = D0P0
      RETURN
      END
```

```
      FUNCTION    MAXVEL ( R )
      COMMON / DATA    / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                   L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                   IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION   XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                   YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / RADIAN /  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION   PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD /  CALVAL(38),A,B,C,VELMPH,ZTEMPD(59)
      DOUBLE PRECISION   A,B,C,VELMPH
      DOUBLE PRECISION   AL,AP,BL,BP,CP,R
      DATA      AL       / +0.190D+00 /
      DATA      AP       / +0.49671329D+00 /
      DATA      BL       / -0.001D+00 /
      DATA      BP       / -0.01403629D+00 /
      DATA      CP       / +0.00013951D+00 /
C
C-----SUBROUTINE MAXVEL FINDS THE MAXIMUM VELOCITY FOR AN INTERSECTION
C-----PATH BASED ON THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF
C-----THE INTERSECTION PATH
C
                        IF ( R . LT . D0P0 )          GO TO 2010
C-----FIND THE MAXIMUM VELOCITY USING THE LINEAR EQUATION FOR MAXIMUM
C-----SAFE SIDE FRICTION
      A = 1.0D+00
      B = -15.0*R*BL
      C = -15.0*R*AL
      VELMPH = (-B+DSQRT(B**2-4.0*A*C))/(2.0*A)
C-----IF THE MAXIMUM VELOCITY IS GT 46.7 THEN THE LINEAR EQUATION FOR
C-----MAXIMUM SAFE SIDE FRICTION WAS VALID AND GO TO 1010
                        IF ( VELMPH . GT . 46.7D+00 )GO TO 1010
C-----CALCULATE THE MAXIMUM VELOCITY USING THE PARABOLIC EQUATION FOR
C-----MAXIMUM SAFE SIDE FRICTION
      A = 1.0-15.0*R*CP
      B = -15.0*R*BP
      C = -15.0*R*AP
      VELMPH = (-B+DSQRT(B**2-4.0*A*C))/(2.0*A)
 1010 CONTINUE
C-----CONVERT THE MAXIMUM VELOCITY FROM MPH TO FPS
      MAXVEL = FPSMPH*VELMPH + XROUND
      RETURN
 2010 CONTINUE
      IFLAG = 1
      RETURN
      END
```

```
      SUBROUTINE ADDPTH                                                 COLEASE
C     TASK,ADDPTH
      COMMON / PATH   / IGEOCP(60),IXL   ( 2),IYL   ( 2),JXL   ( 2),    COLEASE
     *                  JYL   ( 2),IXA   ( 2),IYA   ( 2),LL1     ,      COLEASE
     *                  LA1      ,LA2      ,LL2      ,IIA      ,        COLEASE
     *                  IIL      ,IOA      ,IOL      ,IOPT     ,        COLEASE
     *                  ILCH     ,IBA   ( 2),IDA   ( 2),IRA   ( 2),     COLEASE
     *                  IPTURN   ,LENP     ,LIBL     ,LOBL     ,        COLEASE
     *                  LIMP     ,NGEOCP                                COLEASE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / JSLIM,KSLIM,NPINT,ZTEMPD(102)
      DIMENSION         MSG909(9),MSG910(11)
      DATA      MSG909 / 4H NUM,4HBER ,4HOF P,4HATHS,4H IS ,4HGT 1,
     *                  4H25 -,4H ADD,4HPTH  /
      DATA      MSG910 / 4H NUM,4HBER ,4HOF P,4HATHS,4H FRO,4HM LA,
     *                  4HNE I,4HS GT,4H 7 -,4H ADD,4HPTH  /
C
C-----SUBROUTINE ADDPTH ADDS INTERSECTION PATHS FOR A LANE
C
      NPATHS = NPATHS + 1
                    IF ( NPATHS . GT . 125 )     GO TO 9090
C-----SET UP INDEXES FOR THE INTERSECTION PATHS
      IIA = IA
      IIL = ILN
      LIBL = IL
      IOA = JA
      IOL = JLN
      LOBL = JL
C-----STORE PARAMETERS FOR SECTION 1 (LINE 1) OF THE INTERSECTION PATH
      CALL  XROTI   ( X11,Y11,JAZIM,IXL(1),IYL(1) )
      LL1 = L1
      CALL  XROTI   ( X12,Y12,JAZIM,JXL(1),JYL(1) )
C-----STORE PARAMETERS FOR SECTION 2 (ARC 1) OF THE INTERSECTION PATH
      CALL  XROTI   ( XC2,YC2,JAZIM,IXA(1),IYA(1) )
      LA1 = L2
      IRA(1) = RA2 + XROUND
      CALL  AJAZIM  ( JAZIM,JB2,IBA(1),JD2,IDA(1),L2 )
C-----STORE PARAMETERS FOR SECTION 3 (ARC 2) OF THE INTERSECTION PATH
      CALL  XROTI   ( XC3,YC3,JAZIM,IXA(2),IYA(2) )
      LA2 = L3
      IRA(2) = RA3 + XROUND
      CALL  AJAZIM  ( JAZIM,JB3,IBA(2),JD3,IDA(2),L3 )
C-----STORE PARAMETERS FOR SECTION 4 (LINE 2) OF THE INTERSECTION PATH
      CALL  XROTI   ( X41,Y41,JAZIM,IXL(2),IYL(2) )
      LL2 = L4
      CALL  XROTI   ( X42,Y42,JAZIM,JXL(2),JYL(2) )
C-----STORE OTHER PARAMETERS FOR THE INTERSECTION PATH
      LENP = L1 + L2 + L3 + L4
      IPTURN = KTURN
C     COLEASE,FIND,JSLIM,APPRO,IA,ISLIM
      CALL  FIND    (JSLIM ,     1,IA     ,     12)                     COLEASE
C     COLEASE,FIND,KSLIM,APPRO,JA,ISLIM
      CALL  FIND    (KSLIM ,     1,JA     ,     12)                     COLEASE
      LIMP = MIN0(JSPEED,JSLIM,KSLIM)
      IOPT = JOPT
      ILCH = JLCH
C-----BIAS THE INTERSECTION PATH PARAMETERS
      IXA(1) = IXA(1) + 900
      IXA(2) = IXA(2) + 900
      IYA(1) = IYA(1) + 900
      IYA(2) = IYA(2) + 900
      IDA(1) = IDA(1) + 360
      IDA(2) = IDA(2) + 360
C-----STORE THE INTERSECTION PATH IN ENTRY NPATHS OF ENTITY PATH
C     COLEASE,REPACK,PATH,NPATHS
      CALL  REPACK  (      6,NPATHS)                                    COLEASE
C-----UN-BIAS THE INTERSECTION PATH PARAMETERS
      IXA(1) = IXA(1) - 900
      IXA(2) = IXA(2) - 900
      IYA(1) = IYA(1) - 900
      IYA(2) = IYA(2) - 900
      IDA(1) = IDA(1) - 360
      IDA(2) = IDA(2) - 360
C-----ADD THE INTERSECTION PATH FOR THE INBOUND LANE
C     COLEASE,FIND,NPINT,LANE,IL,NPINT
      CALL  FIND    (NPINT ,     4,IL     ,     5)                      COLEASE
      NPINT = NPINT + 1
                    IF ( NPINT . GT . 7 )        GO TO 9100
C     COLEASE,STORE,NPINT,LANE,IL,NPINT
      CALL  STORE   (NPINT ,     4,IL     ,     5)                      COLEASE
C     COLEASE,STORE,NPATHS,LANE,IL,LINTP(NPINT)
      CALL  STORE   (NPATHS,     4,IL     ,     5+NPINT )               COLEASE
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9090 CONTINUE
      NPATHS = 125
      CALL  ABORTR  ( MSG909,35 )
      STOP  909
 9100 CONTINUE
      CALL  ABORTR  ( MSG910,43 )
      STOP  910
      END                                                               ADDPTH
```

```
      SUBROUTINE AJAZIM ( JAZIML,JB2OR3,IBAL,JD2OR3,IDAL,L2OR3 )
      COMMON / ZTEMPD / ADDPTH(4),ZTEMPD(101)
C
C-----SUBROUTINE AJAZIM ADDS JAZIML TO JB2OR3 AND MAKES IT FALL IN THE
C-----RANGE FROM 0 TO 359 DEGREES AND SETS IDAL TO JD2OR3 WHEN THE
C-----LENGTH OF THE ARC (L2OR3) IS GT 0
C
C-----ADD JAZIML TO JB2OR3 AND MAKE IT FALL IN THE RANGE FROM 0 TO 359
C-----DEGREES
      IBAL = JAZIML + JB2OR3
 1010 CONTINUE
                      IF ( IBAL . LT .   0 )        IBAL = IBAL + 360
                      IF ( IBAL . GE . 360 )        IBAL = IBAL - 360
                      IF ( IBAL . LT .   0 )        GO TO 1010
                      IF ( IBAL . GE . 360 )        GO TO 1010
C-----SET IDAL TO JD2OR3
      IDAL = JD2OR3
C-----IF THE LENGTH OF THE ARC (L2OR3) IS GT 0 THEN RETURN
                      IF ( L2OR3 . GT . 0 )         RETURN
C-----SET IBAL AND IDAL TO 0 AND RETURN
      IBAL = 0
      IDAL = 0
      RETURN
      END                                                               AJAZIM
```

```
      SUBROUTINE DRWPTH                                                 COLEASE
C     TASK,DRWPTH
      COMMON / PATH   / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2), COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1       , COLEASE
     *                  LA1       ,LA2       ,LL2       ,IIA       ,    COLEASE
     *                  IIL       ,IDA       ,IOL       ,IOPT      ,    COLEASE
     *                  ILCH      ,IBA    ( 2),IDA    ( 2),IRA    ( 2), COLEASE
     *                  IPTURN    ,LENP      ,LIBL      ,LOBL      ,    COLEASE
     *                  LIMP      ,NGEOCP                               COLEASE
      COMMON / DATA   / XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42,JANGLE,L1,
     *                  L2,L3,L4,JB2,JD2,JB3,JD3,KTURN,JSPEED,JOPT,
     *                  IFLAG,JAZIM,KAZIM,JLCH
      DOUBLE PRECISION  XI,YI,XO,YO,ADX,ADY,RC,YC,X11,Y11,X12,Y12,XC2,
     *                  YC2,RA2,XC3,YC3,RA3,X41,Y41,X42,Y42
      COMMON / ZTEMPD / X1,X2,Y1,Y2,ZTEMPD(97)
      DOUBLE PRECISION  X1,X2,Y1,Y2
C
C-----SUBROUTINE DRWPTH DRAWS AN INTERSECTION PATH ON THE PLOT PAGE
C
                      IF ( LL1 . LE . 0 )           GO TO 1010
C-----DRAW SECTION 1 (LINE 1) OF THE INTERSECTION PATH ON THE PLOT PAGE
      CALL  XROTX   ( X11,Y11,JAZIM,X1,Y1 )
      CALL  XROTX   ( X12,Y12,JAZIM,X2,Y2 )
      CALL  DRWLIN  ( X1,Y1,X2,Y2 )
 1010 CONTINUE
                      IF ( LA1 . LE . 0 )           GO TO 2010
C-----DRAW SECTION 2 (ARC 1) OF THE INTERSECTION PATH ON THE PLOT PAGE
      CALL  DRWARC  ( IXA(1),IYA(1),IBA(1),IDA(1),IRA(1) )
 2010 CONTINUE
                      IF ( LA2 . LE . 0 )           GO TO 3010
C-----DRAW SECTION 3 (ARC 2) OF THE INTERSECTION PATH ON THE PLOT PAGE
      CALL  DRWARC  ( IXA(2),IYA(2),IBA(2),IDA(2),IRA(2) )
 3010 CONTINUE
                      IF ( LL2 . LE . 0 )           GO TO 4010
C-----DRAW SECTION 4 (LINE 2) OF THE INTERSECTION PATH ON THE PLOT PAGE
      CALL  XROTX   ( X41,Y41,JAZIM,X1,Y1 )
      CALL  XROTX   ( X42,Y42,JAZIM,X2,Y2 )
      CALL  DRWLIN  ( X1,Y1,X2,Y2 )
 4010 CONTINUE
      RETURN
      END                                                               DRWPTH
```

```
      SUBROUTINE CHKPTH                                                 COLEASE
C     TASK,CHKPTH
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / ZTEMPD / IPINT,IPTURN,ITEST,JPINT,JTEST,ZTEMPD(100)
      DIMENSION         ITURNC(4),MSG911(12),MSG912(15)
      DATA     ITURNC / 1HR,1HS,1HL,1HU /
      DATA     MSG911 / 4H NO ,4HPATH,4H INT,4HO TH,4HE IN,4HTERS,
     *                  4HECTI,4HON F,4HOR L,4HANE ,4H- CH,4HKPTH/
      DATA     MSG912 / 4H PAT,4HH HA,4HS NO,4HT GE,4HNERA,4HTED ,
     *                  4HFOR ,4HEACH,4H TUR,4HN CO,4HDE F,4HOR L,
     *                  4HANE ,4H- CH,4HKPTH/
  911 FORMAT(17H0INBOUND APPROACH,I2,2H =,I3,23H - NO INTERSECTION PATH,
     *       25HS WERE GENERATED FOR LANE,I2)
  912 FORMAT(17H0INBOUND APPROACH,I2,2H =,I3,23H - NO INTERSECTION PATH,
     *       19H GENERATED FOR LANE,I2,21H WITH A TURN CODE = (,A1,1H))
C
C-----SUBROUTINE CHKPTH CHECKS EACH INBOUND LANE THAT IS AVAILABLE AT
C-----THE INTERSECTION TO SEE IF AN INTERSECTION PATH WAS CALCULATED FOR
C-----EACH TURNING MOVEMENT SPECIFIED FOR THE INBOUND LANE
C
C-----PROCESS EACH INBOUND APPROACH
      DO 1040  IAN = 1 , NIBA
      IA = LIBA(IAN)
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (     1,IA     )                                    COLEASE
C-----PROCESS EACH LANE OF THE INBOUND APPROACH
      DO 1030  ILN = 1 , NLANES
      IL = LLANES(ILN)
C-----IF THE INBOUND LANE IS NOT AVAILABLE AT THE INTERSECTION THEN
C-----PROCESS THE NEXT INBOUND LANE
                      IF ( IXAPP(IL) . LT . 0 )    GO TO 1030
                      IF ( IYAPP(IL) . LT . 0 )    GO TO 1030
C     COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (     4,IL     )                                    COLEASE
                      IF ( NPINT . LE . 0 )        GO TO 9110
C-----TEST THE INBOUND LANE FOR EACH TURN CODE POSSIBLE
      DO 1020  ITEST = 1 , 4
      JTEST = LSHIFT(1,ITEST-1)
C-----IF THE INBOUND LANE DID NOT HAVE THE TURN CODE SELECTED THEN
C-----PROCESS THE NEXT TURN CODE POSSIBLE
            IF ( IAND(JTEST,LTURN) . EQ . 0 )      GO TO 1020
C-----CHECK EACH INTERSECTION PATH FROM THIS INBOUND LANE TO SEE IF AT
C-----LEAST ONE OF THE INTERSECTION PATHS HAS THE TEST TURN CODE
      DO 1010  IPINT = 1 , NPINT
      JPINT = LINTP(IPINT)
C     COLEASE,FIND,IPTURN,PATH,JPINT,IPTURN
      CALL  FIND    (IPTURN,     6,JPINT ,     89)                      COLEASE
C-----IF THE TURN CODES MATCH THEN PROCESS THE NEXT TURN CODE POSSIBLE
            IF ( IAND(IPTURN,JTEST) . NE . 0 )     GO TO 1020
C-----END OF INTERSECTION PATH LOOP
 1010 CONTINUE
      GO TO 9120
C-----END OF TEST TURN CODE LOOP
 1020 CONTINUE
C-----END OF INBOUND LANE LOOP
 1030 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 1040 CONTINUE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9110 CONTINUE
      PRINT 911 , IAN,IA,ILN
      CALL  ABORTR  ( MSG911,48 )
      STOP  911
 9120 CONTINUE
      PRINT 912 , IAN,IA,ILN,ITURNC(ITEST)
      CALL  ABORTR  ( MSG912,60 )
      STOP  912
      END
```

```
      SUBROUTINE WRITLA                                                 COLEASE
C     TASK,WRITLA
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / I,ILANE,ISDRS,NUMLAN,ZTEMPD(101)
  601 FORMAT(20I4)
C
C-----SUBROUTINE WRITLA WRITES THE LANE INFORMATION AND THE SIGHT
C-----DISTANCE RESTRICTION INFORMATION ONTO TAPE MODELT FOR SIMPRO
C
      NUMLAN = NIBL + NOBL
      WRITE (MODELT,601)  NUMLAN
C-----WRITE THE INFORMATION FOR EACH LANE
      DO 1010  ILANE = 1 , NUMLAN
C     COLEASE,EXTRAC,LANE,ILANE
      CALL  EXTRAC  (      4,ILANE )                                    COLEASE
                    IF ( LTYPE  . EQ . 2 )         LTURN = 0
      WRITE (MODELT,601)  LWID,LTURN,NPINT,NLL,NLR,ISNA,LGEOM,IDX,IBLN
                    IF ( NPINT  . LE . 0 )         GO TO 1010
      WRITE (MODELT,601)  (LINTP(I),I=1,NPINT)
 1010 CONTINUE
      WRITE (MODELT,601)  NSDRS
                    IF ( NSDRS  . LE . 0 )         GO TO 2020
C-----WRITE THE INFORMATION FOR EACH SIGHT DISTANCE RESTRICTION
      DO 2010  ISDRS = 1 , NSDRS
C     COLEASE,EXTRAC,SDR,ISDRS
      CALL  EXTRAC  (      7,ISDRS )                                    COLEASE
      WRITE (MODELT,601)  ICANSE
 2010 CONTINUE
 2020 CONTINUE
      RETURN
      END                                                               WRITLA
```

```
      SUBROUTINE FNDCON                                                 COLEASE
C     TASK,FNDCON
      COMMON / PATH   / IGEOCP(60),IXL     ( 2),IYL     ( 2),JXL    ( 2),   COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA     ( 2),LL1       ,    COLEASE
     *                  LA1        ,LA2        ,LL2        ,IIA        ,    COLEASE
     *                  IIL        ,IOA        ,IOL        ,IOPT       ,    COLEASE
     *                  ILCH       ,IBA     ( 2),IDA    ( 2),IRA    ( 2),   COLEASE
     *                  IPTURN     ,LENP       ,LIBL       ,LOBL       ,    COLEASE
     *                  LIMP       ,NGEOCP                                  COLEASE
      COMMON / GEOCP  / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / ZTEMPD / IBAND,IFS,IZ,JCLOSE,MIBL,MLCH,MOA,MOBL,MPTHP1,
     *                  MPTURN,NC,NPM1,ZTEMPD(93)
      DIMENSION         JGEOCP(1),MSG913(11)
      EQUIVALENCE       (MXL(1,1),JGEOCP(1))
      DATA      MSG913 / 4H TOT,4HAL N,4HUMBE,4HR OF,4H CON,4HFLIC,
     *                   4HTS I,4HS LE,4H 0 -,4H FND,4HCON  /
C
C-----SUBROUTINE FNDCON FINDS THE INTERSECTION CONFLICTS BETWEEN THE
C-----INTERSECTION PATHS
C
      NPM1 = NPATHS - 1
C-----CHECK EACH INTERSECTION PATH EXCEPT THE LAST
      DO 7010  MPTH = 1 , NPM1
C     COLEASE,EXTRAC,PATH,MPTH
      CALL  EXTRAC  (      6,MPTH   )                                   COLEASE
C-----UN-BIAS THE INTERSECTION PATH ATTRIBUTES
      IXA(1) = IXA(1) - 900
      IXA(2) = IXA(2) - 900
      IYA(1) = IYA(1) - 900
      IYA(2) = IYA(2) - 900
      IDA(1) = IDA(1) - 360
      IDA(2) = IDA(2) - 360
      DO 1010  IZ = 1 , 94
      JGEOCP(IZ) = 0
 1010 CONTINUE
C-----SET THE INTERSECTION PATH AS THE MAIN INTERSECTION PATH IN THE
C-----BAND
      MXL(1,1) = IXL(1)
      MXL(2,1) = IXL(2)
      MYL(1,1) = IYL(1)
      MYL(2,1) = IYL(2)
      NXL(1,1) = JXL(1)
      NXL(2,1) = JXL(2)
      NYL(1,1) = JYL(1)
      NYL(2,1) = JYL(2)
      MXA(1,1) = IXA(1)
      MXA(2,1) = IXA(2)
      MYA(1,1) = IYA(1)
      MYA(2,1) = IYA(2)
      MLL(1) = LL1
      MAL(1) = LA1
      MAL(2) = LA2
      MLL(2) = LL2
      MBA(1,1) = IBA(1)
      MBA(2,1) = IBA(2)
      MDA(1,1) = IDA(1)
      MDA(2,1) = IDA(2)
      MRA(1,1) = IRA(1)
      MRA(2,1) = IRA(2)
      MIA = IIA
      MIBL = LIBL
      MOA = IOA
      MOBL = LOBL
```

```
      MLCH = ILCH
      MPTURN = IPTURN
C-----BUILD A BAND 1 FOOT TO THE LEFT AND TO THE RIGHT OF THE MAIN
C-----INTERSECTION PATH
      CALL  BAND     ( 2,1,-1 )
      CALL  BAND     ( 3,1,+1 )
      JCLOSE = -1
      MPTHP1 = MPTH + 1
C-----CHECK AGAINST EACH INTERSECTION PATH THAT HAS A HIGHER NUMBER
C-----THAN THE INTERSECTION PATH BEING CHECKED
      DO 6010  NPTH = MPTHP1 , NPATHS
C    COLEASE,EXTRAC,PATH,NPTH
      CALL  EXTRAC  (      6,NPTH  )                                    COLEASE
C-----UN-BIAS THE INTERSECTION PATH ATTRIBUTES
      IXA(1) = IXA(1) - 900
      IXA(2) = IXA(2) - 900
      IYA(1) = IYA(1) - 900
      IYA(2) = IYA(2) - 900
      IDA(1) = IDA(1) - 360
      IDA(2) = IDA(2) - 360
C-----IF THE INTERSECTION PATHS ORIGINATE FROM THE SAME INBOUND APPROACH
C-----AND THE SAME INBOUND LANE THEN SKIP THIS INTERSECTION PATH
            IF ( MIA.EQ.IIA.AND.MIBL.EQ.LIBL )   GO TO 6010
C-----IF EITHER OF THE INTERSECTION PATHS CHANGES LANES THEN SKIP THE
C-----NEXT TEST
                    IF ( MLCH , NE , 0 )           GO TO 1020
                    IF ( ILCH , NE , 0 )           GO TO 1020
C-----IF THE INTERSECTION PATHS ORIGINATE FROM THE SAME INBOUND APPROACH
C-----AND GO TO DIFFERENT OUTBOUND LANES THEN SKIP THIS INTERSECTION
C-----PATH
            IF ( MIA.EQ.IIA.AND.MOBL.NE.LOBL )   GO TO 6010
 1020 CONTINUE
C-----IF EITHER OF THE INTERSECTION PATHS IS A STRAIGHT THROUGH MOVEMENT
C-----OR A RIGHT TURN THEN GO TO 1030 AND BUILD THE 7 FOOT BANDS
                    IF ( MPTURN , LE , 2 )         GO TO 1030
                    IF ( IPTURN , LE , 2 )         GO TO 1030
C-----IF THE INTERSECTION PATHS GO TO THE SAME OUTBOUND APPROACH BUT GO
C-----TO DIFFERENT OUTBOUND LANES THEN GO TO 1030 AND BUILD THE 7 FOOT
C-----BANDS
            IF ( MOA.EQ.IOA.AND.MOBL.NE.LOBL )   GO TO 1030
C-----BOTH INTERSECTION PATHS ARE U-TURN OR LEFT TURNS THUS IF THE
C-----ICLOSE BANDS ARE ALREADY BUILT THEN GO TO 1050 ELSE BUILD THE
C-----ICLOSE BANDS
                    IF ( JCLOSE , EQ , ICLOSE )  GO TO 1050
      JCLOSE = ICLOSE
      GO TO 1040
 1030 CONTINUE
C-----ONE OF THE INTERSECTION PATHS IS A STRAIGHT THROUGH MOVEMENT OR
C-----A RIGHT TURN THUS IF THE 7 FOOT BANDS ARE ALREADY BUILT THEN GO
C-----TO 1050 ELSE BUILD THE 7 FOOT BANDS
                    IF ( JCLOSE , EQ , 7 )         GO TO 1050
      JCLOSE = 7
 1040 CONTINUE
      CALL  BAND     ( 4,JCLOSE,-1 )
      CALL  BAND     ( 5,JCLOSE,+1 )
 1050 CONTINUE
      NC = 0
C-----CHECK EACH BAND OF THE INTERSECTION PATH STARTING WITH THE MAIN
C-----INTERSECTION PATH, THEN THE 1 FOOT BANDS, AND FINALLY THE ICLOSE
C-----BANDS
      DO 5010  IBAND = 1 , 5
C-----CHECK THE FIRST AND SECOND LINE AND ARC
      DO 4010  IFS = 1 , 2
                    IF ( MLL(IFS) , EQ , 0 )       GO TO 3010
                    IF ( LL1 , EQ , 0 )            GO TO 2010
C-----CHECK BAND IBAND OF LINE IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH LINE 1 OF THE OTHER INTERSECTION PATH
      CALL  CLTOLC   ( IFS,IBAND,1,NC )
 2010 CONTINUE
                    IF ( LA1 , EQ , 0 )            GO TO 2020
C-----CHECK BAND IBAND OF LINE IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH ARC 1 OF THE OTHER INTERSECTION PATH
      CALL  CLTOAC   ( IFS,IBAND,1,NC )
 2020 CONTINUE
                    IF ( LA2 , EQ , 0 )            GO TO 2030
C-----CHECK BAND IBAND OF LINE IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH ARC 2 OF THE OTHER INTERSECTION PATH
      CALL  CLTOAC   ( IFS,IBAND,2,NC )
 2030 CONTINUE
                    IF ( LL2 , EQ , 0 )            GO TO 3010
C-----CHECK BAND IBAND OF LINE IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH LINE 2 OF THE OTHER INTERSECTION PATH
      CALL  CLTOLC   ( IFS,IBAND,2,NC )
 3010 CONTINUE
                    IF ( MAL(IFS) , EQ , 0 )       GO TO 4010
                    IF ( LL1 , EQ , 0 )            GO TO 3020
C-----CHECK BAND IBAND OF ARC IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH LINE 1 OF THE OTHER INTERSECTION PATH
      CALL  CATOLC   ( IFS,IBAND,1,NC )
 3020 CONTINUE
                    IF ( LA1 , EQ , 0 )            GO TO 3030
C-----CHECK BAND IBAND OF ARC IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH ARC 1 OF THE OTHER INTERSECTION PATH
      CALL  CATOAC   ( IFS,IBAND,1,NC )
 3030 CONTINUE
                    IF ( LA2 , EQ , 0 )            GO TO 3040
C-----CHECK BAND IBAND OF ARC IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH ARC 2 OF THE OTHER INTERSECTION PATH
      CALL  CATOAC   ( IFS,IBAND,2,NC )
 3040 CONTINUE
                    IF ( LL2 , EQ , 0 )            GO TO 4010
C-----CHECK BAND IBAND OF ARC IFS OF THE INTERSECTION PATH FOR
C-----CONFLICTS WITH LINE 2 OF THE OTHER INTERSECTION PATH
      CALL  CATOLC   ( IFS,IBAND,2,NC )
C-----END OF FIRST OR SECOND ARC OR LINE LOOP
 4010 CONTINUE
C-----IF A CONFLICT WAS DETECTED THEN GO TO THE NEXT INTERSECTION PATH
                    IF ( NC , NE , 0 )             GO TO 5020
C-----END OF BAND LOOP
 5010 CONTINUE
 5020 CONTINUE
C-----END OF OTHER INTERSECTION PATH LOOP
 6010 CONTINUE
C-----END OF INTERSECTION PATH LOOP
 7010 CONTINUE
                    IF ( NCONFS , LE , 0 )         GO TO 9130
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9130 CONTINUE
      CALL  ABORTR   ( MSG913,43 )
      STOP  913
      END                                                               ENDCON
```

```
      SUBROUTINE BAND   ( IB,IDIST,ILR )
      COMMON / GEOCP   / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                   NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                   MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(12),BEARX,BEARY,IAZ1,IAZ2,ZTEMPD(87)
      DOUBLE PRECISION  BEARX,BEARY
      DOUBLE PRECISION  AZIM36
C
C-----SUBROUTINE BAND BUILDS A BAND IDIST DISTANCE FROM THE MAIN
C-----INTERSECTION PATH EITHER LEFT OR RIGHT OF THE MAIN INTERSECTION
C-----PATH DEPENDING UPON ILR
C
                    IF ( MLL(1) . LE . 0 )        GO TO 1010
C-----BUILD A BAND FOR SECTION 1 (LINE 1) OF THE INTERSECTION PATH
      BEARX = NXL(1,1) - MXL(1,1)
      BEARY = NYL(1,1) - MYL(1,1)
      IAZ1 = AZIM36( BEARY,BEARX ) +  ILR*90 + XROUND
      CALL  XROTAI  ( D0P0,DFLOAT(IDIST),IAZ1,MXL(1,1),MYL(1,1),
     *                MXL(1,IB),MYL(1,IB) )
      CALL  XROTAI  ( D0P0,DFLOAT(IDIST),IAZ1,NXL(1,1),NYL(1,1),
     *                NXL(1,IB),NYL(1,IB) )
 1010 CONTINUE
                    IF ( MAL(1) . LE . 0 )        GO TO 2010
C-----BUILD A BAND FOR SECTION 2 (ARC 1) OF THE INTERSECTION PATH
      MXA(1,IB) = MXA(1,1)
      MYA(1,IB) = MYA(1,1)
      MBA(1,IB) = MBA(1,1)
      MDA(1,IB) = MDA(1,1)
      MRA(1,IB) = MRA(1,1) - ILR*(ISIGN(1,MDA(1,IB))*IDIST) + XROUND
 2010 CONTINUE
                    IF ( MAL(2) . LE . 0 )        GO TO 3010
C-----BUILD A BAND FOR SECTION 3 (ARC 2) OF THE INTERSECTION PATH
      MXA(2,IB) = MXA(2,1)
      MYA(2,IB) = MYA(2,1)
      MBA(2,IB) = MBA(2,1)
      MDA(2,IB) = MDA(2,1)
      MRA(2,IB) = MRA(2,1) - ILR*(ISIGN(1,MDA(2,IB))*IDIST) + XROUND
 3010 CONTINUE
                    IF ( MLL(2) . LE . 0 )        RETURN
C-----BUILD A BAND FOR SECTION 4 (LINE 2) OF THE INTERSECTION PATH
      BEARX = NXL(2,1) - MXL(2,1)
      BEARY = NYL(2,1) - MYL(2,1)
      IAZ2 = AZIM36( BEARY,BEARX ) +  ILR*90 + XROUND
      CALL  XROTAI  ( D0P0,DFLOAT(IDIST),IAZ2,MXL(2,1),MYL(2,1),
     *                MXL(2,IB),MYL(2,IB) )
      CALL  XROTAI  ( D0P0,DFLOAT(IDIST),IAZ2,NXL(2,1),NYL(2,1),
     *                NXL(2,IB),NYL(2,IB) )
      RETURN
      END                                                           BAND
```

```
      SUBROUTINE CLTOLC (IFS,    IBAND,   JFS,     NC)                COLEASE
C     TASK,CLTOLC,IFS,IBAND,JFS,NC
      COMMON / PATH    / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),  COLEASE
     *                   JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1       ,  COLEASE
     *                   LA1       ,LA2       ,LL2       ,IIA       ,  COLEASE
     *                   IIL       ,IOA       ,IOL       ,IOPT      ,  COLEASE
     *                   ILCH      ,IBA    ( 2),IDA    ( 2),IRA    ( 2),  COLEASE
     *                   IPTURN    ,LENP      ,LIBL      ,LOBL      ,  COLEASE
     *                   LIMP      ,NGEOCP                           COLEASE
      COMMON / GEOCP   / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                   NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                   MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(12),AZ1,AZ2,X1,X2,X3,X4,Y1,Y2,Y3,Y4,IL1,
     *                  IL2,ITEST,ZTEMPD(70)
      DOUBLE PRECISION  AZ1,AZ2,X1,X2,X3,X4,Y1,Y2,Y3,Y4
      DOUBLE PRECISION  AZIM36
C
C-----SUBROUTINE CLTOLC CHECKS FOR INTERSECTION CONFLICTS BETWEEN THE
C-----LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
      X1 = MXL(IFS,IBAND)
      Y1 = MYL(IFS,IBAND)
      X2 = NXL(IFS,IBAND)
      Y2 = NYL(IFS,IBAND)
      X3 = IXL(JFS)
      Y3 = IYL(JFS)
      X4 = JXL(JFS)
      Y4 = JYL(JFS)
C-----TEST IF LINE A FROM (X1,Y1) TO (X2,Y2) FOR THE INTERSECTION PATH
C-----BEING CHECKED INTERSECTS WITH LINE B FROM (X3,Y3) TO (X4,Y4) FOR
C-----THE INTERSECTION PATH BEING CHECKED AGAINST
      ITEST = LTOL( X1,Y1,X2,Y2,X3,Y3,X4,Y4,XINT1,YINT1,XINT2,YINT2 )
                    IF ( ITEST . EQ . 0 )        RETURN
C-----FIND THE PARAMETERS FOR THE FIRST INTERSECTION CONFLICT
      IL1 = DSQRT((XINT1-MXL(IFS,IBAND))**2+(YINT1-MYL(IFS,IBAND))**2) +
     *      XROUND
      AZ1 = AZIM36( Y2-Y1,X2-X1 )
                    IF ( IFS . EQ . 1 )          GO TO 1010
      IL1 = IL1 + MLL(1) + MAL(1) + MAL(2)
 1010 CONTINUE
      IL2 = DSQRT((XINT1-IXL(JFS))**2+(YINT1-IYL(JFS))**2) + XROUND
      AZ2 = AZIM36( Y4-Y3,X4-X3 )
                    IF ( JFS . EQ . 1 )          GO TO 1020
      IL2 = IL2 + LL1 + LA1 + LA2
 1020 CONTINUE
C-----ADD THE INTERSECTION CONFLICT BETWEEN THE INTERSECTION PATHS
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ1,NPTH,IIA,IL2,AZ2,NC )
C-----IF THERE WAS ONLY ONE INTERSECTION CONFLICT BETWEEN LINE A AND
C-----LINE B THEN RETURN ELSE FIND THE PARAMETERS FOR THE INTERSECTION
C-----CONFLICT
                    IF ( ITEST . EQ . 1 )        RETURN
      IL1 = DSQRT((XINT2-MXL(IFS,IBAND))**2+(YINT2-MYL(IFS,IBAND))**2) +
     *      XROUND
                    IF ( IFS . EQ . 1 )          GO TO 2010
      IL1 = IL1 + MLL(1) + MAL(1) + MAL(2)
 2010 CONTINUE
      IL2 = DSQRT((XINT2-IXL(JFS))**2+(YINT2-IYL(JFS))**2) + XROUND
                    IF ( JFS . EQ . 1 )          GO TO 2020
      IL2 = IL2 + LL1 + LA1 + LA2
 2020 CONTINUE
C-----ADD THE INTERSECTION CONFLICT BETWEEN THE INTERSECTION PATHS
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ1,NPTH,IIA,IL2,AZ2,NC )
      RETURN
      END                                                           CLTOLC
```

```
      SUBROUTINE ADDCON (INP,    INA,    INL,    AI,    JNP,    JNA,      COLEASE
     *                   JNL,    AJ,     NC)                              COLEASE
C   TASK,ADDCON,INP,INA,INL,AI,JNP,JNA,JNL,AJ,NC
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,      COLEASE
     *                  ICONI ( 2),IDUMCO                                 COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(86),IANGLE,ICON,KP,LP,MGEOCP,ZTEMPD(14)
      DIMENSION         MSG914(12),MSG915(12)
      DOUBLE PRECISION  AI,AJ
      DATA     MSG914 / 4H TOT,4HAL N,4HUMBE,4HR OF,4H CON,4HFLIC,
     *                  4HTS I,4HS GT,4H 100,4H0 - ,4HADDC,4HON   /
      DATA     MSG915 / 4H NUM,4HBER ,4HOF C,4HONFL,4HICTS,4H FOR,
     *                  4H PAT,4HH IS,4H GT ,4H60 -,4H ADD,4HCON  /
  915 FORMAT(29H0NUMBER OF CONFLICTS FOR PATH,I4,8H OR PATH,I4,
     *       18H IS GT 60 - ADDCON)
C
C-----SUBROUTINE ADDCON ADDS INTERSECTION CONFLICTS BETWEEN TWO
C-----INTERSECTION PATHS
C
      IANGLE = AJ - AI + XROUND
 1010 CONTINUE
                    IF ( IANGLE . LE . 0 )       IANGLE = IANGLE + 360
                    IF ( IANGLE . GE . 360 )     IANGLE = IANGLE - 360
                    IF ( IANGLE . LT . 0 )       GO TO 1010
                    IF ( IANGLE . GT . 360 )     GO TO 1010
                    IF ( NCONFS . LE . 0 )       GO TO 2020
C-----CHECK TO SEE IF THERE IS ALREADY AN INTERSECTION CONFLICT BETWEEN
C-----THESE TWO INTERSECTION PATHS THAT ARE CLOSE TOGETHER
      DO 2010  ICON = 1 , NCONFS
C   COLEASE,EXTRAC,CONFLT,ICON
      CALL  EXTRAC  (     3,ICON  )                                       COLEASE
      KP = -1
                    IF ( ICONP(1) . EQ . INP )   KP = 1
                    IF ( ICONP(2) . EQ . INP )   KP = 2
C-----IF THE INTERSECTION CONFLICT DOES NOT INVOLVE INTERSECTION PATH
C-----INP THEN GO TO 2010 AND SKIP TO THE NEXT INTERSECTION CONFLICT
                    IF ( KP . LE . 0 )           GO TO 2010
      LP = 3 - KP
C-----IF THE INTERSECTION CONFLICT DOES NOT INVOLVE INTERSECTION PATH
C-----JNP THEN GO TO 2010 AND SKIP TO THE NEXT INTERSECTION CONFLICT
                    IF ( ICONP(LP) . NE . JNP )  GO TO 2010
C-----IF THE DISTANCES TO THE INTERSECTION CONFLICT ARE GT ICLOSE THEN
C-----GO TO 2010 AND SKIP TO THE NEXT INTERSECTION CONFLICT
            IF ( IABS(ICOND(KP)-INL).GT.ICLOSE ) GO TO 2010
            IF ( IABS(ICOND(LP)-JNL).GT.ICLOSE ) GO TO 2010
C-----AVERAGE THE INTERSECTION CONFLICTS AND RE-STORE
      ICOND(KP) = 0.5*(ICOND(KP)+INL) + XROUND
      ICOND(LP) = 0.5*(ICOND(LP)+JNL) + XROUND
                    IF ( KP . EQ . 2 )           IANGLE = 360 - IANGLE
                    IF ( IANGLE . EQ . 360 )     IANGLE = 0
      ICONAN = 0.5*(ICONAN+IANGLE) + XROUND
C   COLEASE,REPACK,CONFLT,ICON
      CALL  REPACK  (     3,ICON  )                                       COLEASE
C-----RETURN WITHOUT ADDING THE INTERSECTION CONFLICT
      RETURN
 2010 CONTINUE
 2020 CONTINUE
      NC = NC + 1
      NCONFS = NCONFS + 1
                    IF ( NCONFS . GT . 1000 )    GO TO 9140
C-----ADD INTERSECTION CONFLICT FOR INTERSECTION PATH BEING CHECKED
C-----(INP)
C   COLEASE,FIND,MGEOCP,PATH,INP,NGEOCP
      CALL  FIND    (MGEOCP,      6,INP    ,      94)                     COLEASE
      MGEOCP = MGEOCP + 1
                    IF ( MGEOCP . GT . 60 )      GO TO 9150
C   COLEASE,STORE,NCONFS,PATH,INP,IGEOCP(MGEOCP)
      CALL  STORE   (NCONFS,      6,INP    ,       0+MGEOCP)              COLEASE
C   COLEASE,STORE,MGEOCP,PATH,INP,NGEOCP
      CALL  STORE   (MGEOCP,      6,INP    ,      94)                     COLEASE
C-----ADD INTERSECTION CONFLICT FOR INTERSECTION PATH BEING CHECKED
C-----AGAINST (JNP)
C   COLEASE,FIND,MGEOCP,PATH,JNP,NGEOCP
      CALL  FIND    (MGEOCP,      6,JNP    ,      94)                     COLEASE
      MGEOCP = MGEOCP + 1
                    IF ( MGEOCP . GT . 60 )      GO TO 9150
C   COLEASE,STORE,NCONFS,PATH,JNP,IGEOCP(MGEOCP)
      CALL  STORE   (NCONFS,      6,JNP    ,       0+MGEOCP)              COLEASE
C   COLEASE,STORE,MGEOCP,PATH,JNP,NGEOCP
      CALL  STORE   (MGEOCP,      6,JNP    ,      94)                     COLEASE
C-----SET PARAMETERS FOR INTERSECTION CONFLICT NCONFS
      ICONP(1) = INP
      ICONP(2) = JNP
      ICONA(1) = INA
      ICONA(2) = JNA
      ICOND(1) = INL
      ICOND(2) = JNL
      ICONAN = IANGLE
      ICONI(1) = 0
      ICONI(2) = 0
C-----STORE INTERSECTION CONFLICT PARAMETERS IN ENTRY NCONFS OF ENTITY
C-----CONFLT
C   COLEASE,REPACK,CONFLT,NCONFS
      CALL  REPACK  (     3,NCONFS)                                       COLEASE
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9140 CONTINUE
      NCONFS = 1000
      CALL  ABORTR  ( MSG914,46 )
      STOP  914
 9150 CONTINUE
      PRINT 915 , INP,JNP
      CALL  ABORTR  ( MSG915,47 )
      STOP  915
      END                                                                 ADDCON
```

```
      SUBROUTINE CLTOAC (IFS,    IBAND,  JFS,     NC)                   COLEASE
C     TASK,CLTOAC,IFS,IBAND,JFS,NC
      COMMON / PATH    / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),  COLEASE
     *                   JYL    ( 2),IXA   ( 2),IYA    ( 2),LL1         ,  COLEASE
     *                   LA1          ,LA2        ,LL2         ,IIA      ,  COLEASE
     *                   IIL          ,IOA        ,IOL         ,IOPT     ,  COLEASE
     *                   ILCH         ,IBA   ( 2),IDA    ( 2),IRA    ( 2),  COLEASE
     *                   IPTURN       ,LENP       ,LIBL        ,LOBL     ,  COLEASE
     *                   LIMP         ,NGEOCP                              COLEASE
      COMMON / GEOCP   / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                   NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                   MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(12),A,B,C,RADICL,X,XB,XM,ZTEMPD(79)
      DOUBLE PRECISION  A,B,C,RADICL,X,XB,XM
C
C-----SUBROUTINE CLTOAC CHECKS FOR INTERSECTION CONFLICTS BETWEEN THE
C-----LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
C-----IF THE LINE IS VERTICAL THEN GO TO 1050
      IF ( IABS(NXL(IFS,IBAND)-MXL(IFS,IBAND)) . LE . 0 )
     *                                         GO TO 1050
C-----FIND THE SLOPE AND THE Y INTERCEPT OF THE LINE
      XM = DFLOAT(NYL(IFS,IBAND)-MYL(IFS,IBAND)) /
     *     DFLOAT(NXL(IFS,IBAND)-MXL(IFS,IBAND))
      XB = MYL(IFS,IBAND) - MXL(IFS,IBAND)*XM
C-----FIND THE POINT(S) OF INTERSECTION BETWEEN THE LINE AND THE ARC
      A = 1.0 + XM**2
      B = -2.0*IXA(JFS) + 2.0*XM*XB - 2.0*IYA(JFS)*XM
      C = IXA(JFS)**2+IYA(JFS)**2+XB**2-IRA(JFS)**2-2.0*IYA(JFS)*XB
      RADICL = B**2 - 4.0*A*C
                       IF ( DABS(RADICL).LE.ZERO )  GO TO 1010
                       IF ( RADICL )                2010 , 1010 , 1030
 1010 CONTINUE
C-----FIND 1 POINT OF INTERSECTION BETWEEN THE LINE AND THE ARC
      XINT1 = -B/(2.0*A)
      YINT1 = XM*XINT1 + XB
 1020 CONTINUE
C-----ADD 1 POINT OF INTERSECTION BETWEEN THE LINE AND THE ARC
      CALL  ADDLA   ( IFS,IBAND,JFS,NC,1 )
      RETURN
 1030 CONTINUE
C-----FIND 2 POINTS OF INTERSECTION BETWEEN THE LINE AND THE ARC
      XINT1 = (-B+DSQRT(RADICL))/(2.0*A)
      YINT1 = XM*XINT1 + XB
      XINT2 = (-B-DSQRT(RADICL))/(2.0*A)
      YINT2 = XM*XINT2 + XB
 1040 CONTINUE
C-----ADD 2 POINTS OF INTERSECTION BETWEEN THE LINE AND THE ARC
      CALL  ADDLA   ( IFS,IBAND,JFS,NC,2 )
      RETURN
 1050 CONTINUE
C-----FIND THE INTERSECTION BETWEEN THE VERTICAL LINE AND THE ARC
      X = DBLE(0.5*(MXL(IFS,IBAND)+NXL(IFS,IBAND)))
      A = 1.0D+00
      B = DBLE(-2.0*IYA(JFS))
      C = IYA(JFS)**2 + (X-IXA(JFS))**2 - IRA(JFS)**2
      RADICL = B**2  - 4.0*A*C
                       IF ( DABS(RADICL).LE.ZERO )  GO TO 1060
                       IF ( RADICL )                2010 , 1060 , 1070
 1060 CONTINUE
C-----FIND 1 POINT OF INTERSECTION BETWEEN THE VERTICAL LINE AND THE ARC
      XINT1 = X
      YINT1 = -B/(2.0*A)
      GO TO 1020
 1070 CONTINUE
C-----FIND 2 POINTS OF INTERSECTION BETWEEN THE VERTICAL LINE AND THE
C-----ARC
      XINT1 = X
      YINT1 = (-B+DSQRT(RADICL))/(2.0*A)
      XINT2 = X
      YINT2 = (-B-DSQRT(RADICL))/(2.0*A)
      GO TO 1040
 2010 CONTINUE
      RETURN
      END                                                              CLT AC
```

```
      SUBROUTINE ADDLA  (IFS,    IBAND,  JFS,     NC,      NUM)            COLEASE
C     TASK,ADDLA,IFS,IBAND,JFS,NC,NUM
      COMMON / PATH   / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),    COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1         ,   COLEASE
     *                  LA1         ,LA2        ,LL2        ,IIA        ,   COLEASE
     *                  IIL         ,IOA        ,IOL        ,IOPT       ,   COLEASE
     *                  ILCH        ,IBA    ( 2),IDA    ( 2),IRA    ( 2),   COLEASE
     *                  IPTURN      ,LENP       ,LIBL       ,LOBL       ,   COLEASE
     *                  LIMP        ,NGEOCP                                 COLEASE
      COMMON / GEOCP  / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(44),AZIM1,AZIM2,AZ11,AZ12,AZ21,AZ22,
     *                  BEARX,BEARY,DA1,DA2,X,XBEAR,YBEAR,IL1,IL2,
     *                  ITEST1,ITEST2,JTEST1,JTEST2,NUMPTS,ZTEMPD(28)
      DOUBLE PRECISION  AZIM1,AZIM2,AZ11,AZ12,AZ21,AZ22,BEARX,BEARY,DA1,
     *                  DA2,X,XBEAR,YBEAR
      DOUBLE PRECISION  AZIM36
C
C-----SUBROUTINE ADDLA ADDS INTERSECTION CONFLICTS BETWEEN THE LINE
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
      NUMPTS = NUM
 1010 CONTINUE
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE LINE
      ITEST1 = ICHKL( MXL(IFS,IBAND),MYL(IFS,IBAND),NXL(IFS,IBAND),
     *         NYL(IFS,IBAND),XINT1,YINT1 )
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE ARC
      BEARX = NXL(IFS,IBAND) - MXL(IFS,IBAND)
      BEARY = NYL(IFS,IBAND) - MYL(IFS,IBAND)
      AZ11 = AZIM36( BEARY,BEARX )
      XBEAR = XINT1 - IXA(JFS)
      YBEAR = YINT1 - IYA(JFS)
      AZIM1 = AZIM36( YBEAR,XBEAR )
      AZ12 = AZIM1 + ISIGN(90,IDA(JFS))
      ITEST2 = ICHKA( AZIM1,IBA(JFS),IDA(JFS),DA1 )
      JTEST1 = 1
      JTEST2 = 1
                    IF ( NUMPTS . EQ . 1 )        GO TO 1020
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE LINE
      JTEST1 = ICHKL( MXL(IFS,IBAND),MYL(IFS,IBAND),NXL(IFS,IBAND),
     *         NYL(IFS,IBAND),XINT2,YINT2 )
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE ARC
      BEARX = NXL(IFS,IBAND) - MXL(IFS,IBAND)
      BEARY = NYL(IFS,IBAND) - MYL(IFS,IBAND)
      AZ21 = AZIM36( BEARY,BEARX )
      XBEAR = XINT2 - IXA(JFS)
      YBEAR = YINT2 - IYA(JFS)
      AZIM2 = AZIM36( YBEAR,XBEAR )
      AZ22 = AZIM2 + ISIGN(90,IDA(JFS))
      JTEST2 = ICHKA( AZIM2,IBA(JFS),IDA(JFS),DA2 )
 1020 CONTINUE
C-----IF NEITHER POINT OF INTERSECTION LIES ON BOTH THE LINE AND THE ARC
C-----THEN RETURN
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )        RETURN
C-----IF ONLY THE FIRST POINT OF INTERSECTION LIES ON BOTH THE LINE AND
C-----THE ARC THEN ADD THE FIRST POINT OF INTERSECTION
      IF ( (ITEST1.EQ.0.AND.ITEST2.EQ.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )        GO TO 2010
C-----IF ONLY THE SECOND POINT OF INTERSECTION LIES ON BOTH THE LINE AND
C-----THE ARC THEN ADD THE SECOND POINT OF INTERSECTION
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.EQ.0.AND.JTEST2.EQ.0) )        GO TO 3010
C-----IF THIS IS NOT THE MAIN INTERSECTION PATH THEN GO TO 4010
                    IF ( IBAND . NE . 1 )           GO TO 4010
C-----IF THE DISTANCE BETWEEN THE 2 POINTS OF CONFLICT ON THE MAIN
C-----INTERSECTION PATH IS LE ICLOSE THEN GO TO 4010
      X = DSQRT((XINT1-XINT2)**2+(YINT1-YINT2)**2)
                    IF ( X.LE.DFLOAT(ICLOSE) )    GO TO 4010
 2010 CONTINUE
C-----ADD FIRST POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = DSQRT((XINT1-MXL(IFS,IBAND))**2+(YINT1-MYL(IFS,IBAND))**2) +
     *      XROUND
                    IF ( IFS . EQ . 1 )             GO TO 2020
      IL1 = IL1 + MLL(1) + MAL(1) + MAL(2)
 2020 CONTINUE
      IL2 = IRA(JFS)*DABS(DA1)*RADIAN + LL1 + XROUND
                    IF ( JFS .EQ . 1 )              GO TO 2030
      IL2 = IL2 + LA1
 2030 CONTINUE
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ11,NPTH,IIA,IL2,AZ12,NC )
C-----IF THE SECOND POINT OF INTERSECTION DOES NOT LIE ON THE LINE OR
C-----THE ARC THEN RETURN
      IF ( JTEST1.NE.0 . OR . JTEST2.NE.0 )        RETURN
 3010 CONTINUE
C-----ADD THE SECOND POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = DSQRT((XINT2-MXL(IFS,IBAND))**2+(YINT2-MYL(IFS,IBAND))**2) +
     *      XROUND
                    IF ( IFS . EQ . 1 )             GO TO 3020
      IL1 = IL1 + MLL(1) + MAL(1) + MAL(2)
 3020 CONTINUE
      IL2 = IRA(JFS)*DABS(DA2)*RADIAN + LL1 + XROUND
                    IF ( JFS .EQ . 1 )              GO TO 3030
      IL2 = IL2 + LA1
 3030 CONTINUE
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ21,NPTH,IIA,IL2,AZ22,NC )
      RETURN
 4010 CONTINUE
C-----COMBINE THE 2 POINTS OF INTERSECTION AND CHECK AGAIN
      XINT1 = 0.5*(XINT1+XINT2)
      YINT1 = 0.5*(YINT1+YINT2)
      NUMPTS = 1
      GO TO 1010
      END                                                                ADDLA
```

```
      FUNCTION    ICHKL   ( IX1,IY1,IX2,IY2,XINT,YINT )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  XINT,YINT
      COMMON / ZTEMPD / CONVAR(86),ZTEMPD(19)
C
C-----FUNCTION ICHKL CHECKS TO SEE IF (XINT,YINT) LIES BETWEEN (IX1,IY1)
C-----AND (IX2,IY2)  (ICHKL=0=YES AND ICHKL=1=NO)
C
      ICHKL = 1
            IF ( (XINT-IX1)*(XINT-IX2).GT.ZERO ) RETURN
            IF ( (YINT-IY1)*(YINT-IY2).GT.ZERO ) RETURN
      ICHKL = 0
      RETURN
      END                                                               ICHKL
```

```
      FUNCTION    ICHKA   ( AZIM,NBA,NDA,DA )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(86),BZIM,ZTEMPD(17)
      DOUBLE PRECISION  BZIM
      DIMENSION         MSG916(13)
      DOUBLE PRECISION  AZIM,DA
      DATA      MSG916 / 4H SWE,4HEP A,4HNGLE,4H FOR,4H ARC,4H POR,
     *                   4HTION,4H OF ,4HPATH,4H IS ,4HEQ 0,4H = I,
     *                   4HCHKA /
C
C-----FUNCTION ICHKA CHECKS TO SEE IF AZIM LIES BETWEEN NBA AND NBA+NDA
C-----AND RETURNS DA
C
      ICHKA = 1
      DA = D0P0
      BZIM = AZIM
                     IF ( NDA )                    1010 , 9160 , 2010
 1010 CONTINUE
C-----NDA NEGATIVE
C-----IF BZIM IS VERY CLOSE TO NBA THEN RETURN WITH ICHKA=0
            IF ( DABS(BZIM-NBA) . LE . XROUND)   GO TO 3010
C-----MAKE BZIM LT NBA
                     IF ( BZIM.LT.DFLOAT(NBA) )   GO TO 1020
      BZIM = BZIM - 360.0
      GO TO 1010
 1020 CONTINUE
      DA = BZIM - NBA
C-----IF DA IS VERY CLOSE TO NDA THEN RETURN WITH ICHKA=0
                     IF ( DABS(DA-NDA).LE.XROUND )GO TO 3010
C-----IF DA IS GE NDA THEN RETURN WITH ICHKA=0
                     IF ( DA . GE . DFLOAT(NDA) ) GO TO 3010
      RETURN
 2010 CONTINUE
C-----NDA IS POSITIVE
C-----IF BZIM IS VERY CLOSE TO NBA THEN RETURN WITH ICHKA=0
            IF ( DABS(BZIM-NBA) . LE . XROUND)   GO TO 3010
C-----MAKE BZIM GT NBA
                     IF ( BZIM.GT.DFLOAT(NBA) )   GO TO 2020
      BZIM = BZIM + 360.0
      GO TO 2010
 2020 CONTINUE
      DA = BZIM - NBA
C-----IF DA IS VERY CLOSE TO NDA THEN RETURN WITH ICHKA=0
                     IF ( DABS(DA-NDA).LE.XROUND )GO TO 3010
C-----IF DA LE NDA THEN RETURN WITH ICHKA=0
                     IF ( DA . LE . DFLOAT(NDA) ) GO TO 3010
      RETURN
 3010 CONTINUE
      ICHKA = 0
      RETURN
 9160 CONTINUE
      CALL  ABORTR  ( MSG916,52 )
      STOP  916
      END                                                               ICHKA
```

```
      SUBROUTINE CATOLC (IFS,    IBAND,   JFS,     NC)                  COLEASE
C    TASK,CATOLC,IFS,IBAND,JFS,NC
      COMMON / PATH    / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),    COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1         ,  COLEASE
     *                  LA1        ,LA2        ,LL2        ,IIA         ,  COLEASE
     *                  IIL        ,IOA        ,IOL        ,IOPT        ,  COLEASE
     *                  ILCH       ,IBA    ( 2),IDA    ( 2),IRA    ( 2),  COLEASE
     *                  IPTURN     ,LENP       ,LIBL       ,LOBL        ,  COLEASE
     *                  LIMP       ,NGEOCP                                COLEASE
      COMMON / GEOCP   / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(12),A,B,C,RADICL,X,XB,XM,ZTEMPD(79)
      DOUBLE PRECISION  A,B,C,RADICL,X,XB,XM
C
C-----SUBROUTINE CATOLC CHECKS FOR INTERSECTION CONFLICTS BETWEEN THE
C-----ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
C-----IF THE LINE IS VERTICAL THEN GO TO 1050
            IF ( IABS(JXL(JFS)-IXL(JFS)).LE.0 )  GO TO 1050
C-----FIND THE SLOPE AND THE Y INTERCEPT OF THE LINE
      XM = DFLOAT(JYL(JFS)-IYL(JFS))/DFLOAT(JXL(JFS)-IXL(JFS))
      XB = IYL(JFS) - IXL(JFS)*XM
C-----FIND THE POINT(S) OF INTERSECTION BETWEEN THE ARC AND THE LINE
      A = 1.0 + XM**2
      B = -2.0*MXA(IFS,IBAND) + 2.0*XM*XB - 2.0*MYA(IFS,IBAND)*XM
      C = MXA(IFS,IBAND)**2 + MYA(IFS,IBAND)**2 + XB**2 -
     *    MRA(IFS,IBAND)**2 - 2.0*MYA(IFS,IBAND)*XB
      RADICL = B**2 - 4.0*A*C
                    IF ( DABS(RADICL).LE.ZERO )  GO TO 1010
                    IF ( RADICL )                2010 , 1010 , 1030
 1010 CONTINUE
C-----FIND 1 POINT OF INTERSECTION BETWEEN THE ARC AND THE LINE
      XINT1 = -B/(2.0*A)
      YINT1 = XM*XINT1 + XB
 1020 CONTINUE
C-----ADD 1 POINT OF INTERSECTION BETWEEN THE ARC AND THE LINE
      CALL  ADDAL   ( IFS,IBAND,JFS,NC,1 )
      RETURN
 1030 CONTINUE
C-----FIND 2 POINTS OF INTERSECTION BETWEEN THE ARC AND THE LINE
      XINT1 = (-B+DSQRT(RADICL))/(2.0*A)
      YINT1 = XM*XINT1 + XB
      XINT2 = (-B-DSQRT(RADICL))/(2.0*A)
      YINT2 = XM*XINT2 + XB
 1040 CONTINUE
C-----ADD 2 POINTS OF INTERSECTION BETWEEN THE ARC AND THE LINE
      CALL  ADDAL   ( IFS,IBAND,JFS,NC,2 )
      RETURN
 1050 CONTINUE
C-----FIND THE INTERSECTION BETWEEN THE ARC AND THE VERTICAL LINE
      X = DBLE(0.5*(IXL(JFS)+JXL(JFS)))
      A = 1.0D+00
      B = DBLE(-2.0*MYA(IFS,IBAND))
      C = MYA(IFS,IBAND)**2 + (X-MXA(IFS,IBAND))**2 - MRA(IFS,IBAND)**2
      RADICL = B**2 - 4.0*A*C
                    IF ( DABS(RADICL).LE.ZERO )  GO TO 1060
                    IF ( RADICL )                2010 , 1060 , 1070
 1060 CONTINUE
C-----FIND 1 POINT OF INTERSECTION BETWEEN THE ARC AND THE VERTICAL LINE
      XINT1 = X
      YINT1 = -B/(2.0*A)
      GO TO 1020
 1070 CONTINUE
C-----FIND 2 POINTS OF INTERSECTION BETWEEN THE ARC AND THE VERTICAL
C-----LINE
      XINT1 = X
      YINT1 = (-B+DSQRT(RADICL))/(2.0*A)
      XINT2 = X
      YINT2 = (-B-DSQRT(RADICL))/(2.0*A)
      GO TO 1040
 2010 CONTINUE
      RETURN
      END                                                              CATOLC
```

```
      SUBROUTINE ADDAL  (IFS,    IBAND,  JFS,     NC,      NUM)                COLEASE
C     TASK,ADDAL,IFS,IBAND,JFS,NC,NUM
      COMMON / PATH    / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),       COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1        ,       COLEASE
     *                  LA1        ,LA2        ,LL2        ,IIA        ,       COLEASE
     *                  IIL        ,IOA        ,IOL        ,IOPT       ,       COLEASE
     *                  ILCH       ,IBA    ( 2),IDA    ( 2),IRA    ( 2),       COLEASE
     *                  IPTURN     ,LENP       ,LIBL       ,LOBL       ,       COLEASE
     *                  LIMP       ,NGEOCP                                     COLEASE
      COMMON / GEOCP  / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(44),AZIM1,AZIM2,AZ11,AZ12,AZ21,AZ22,
     *                  BEARX,BEARY,DA1,DA2,X,XBEAR,YBEAR,IL1,IL2,
     *                  ITEST1,ITEST2,JTEST1,JTEST2,NUMPTS,ZTEMPD(28)
      DOUBLE PRECISION  AZIM1,AZIM2,AZ11,AZ12,AZ21,AZ22,BEARX,BEARY,DA1,
     *                  DA2,X,XBEAR,YBEAR
      DOUBLE PRECISION  AZIM36
C
C-----SUBROUTINE ADDAL ADDS INTERSECTION CONFLICTS BETWEEN THE ARC
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
      NUMPTS = NUM
 1010 CONTINUE
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE ARC
      BEARX = JXL(JFS) - IXL(JFS)
      BEARY = JYL(JFS) - IYL(JFS)
      AZ12 = AZIM36( BEARY,BEARX )
      XBEAR = XINT1 - MXA(IFS,IBAND)
      YBEAR = YINT1 - MYA(IFS,IBAND)
      AZIM1 = AZIM36( YBEAR,XBEAR )
      AZ11 = AZIM1 + ISIGN(90,MDA(IFS,IBAND))
      ITEST1 = ICHKA( AZIM1,MBA(IFS,IBAND),MDA(IFS,IBAND),DA1 )
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE LINE
      ITEST2 = ICHKL( IXL(JFS),IYL(JFS),JXL(JFS),JYL(JFS),XINT1,YINT1 )
      JTEST1 = 1
      JTEST2 = 1
                    IF ( NUMPTS . EQ . 1 )         GO TO 1020
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE ARC
      BEARX = JXL(JFS) - IXL(JFS)
      BEARY = JYL(JFS) - IYL(JFS)
      AZ22 = AZIM36( BEARY,BEARX )
      XBEAR = XINT2 - MXA(IFS,IBAND)
      YBEAR = YINT2 - MYA(IFS,IBAND)
      AZIM2 = AZIM36( YBEAR,XBEAR )
      AZ21 = AZIM2 + ISIGN(90,MDA(IFS,IBAND))
      JTEST1 = ICHKA( AZIM2,MBA(IFS,IBAND),MDA(IFS,IBAND),DA2 )
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE LINE
      JTEST2 = ICHKL( IXL(JFS),IYL(JFS),JXL(JFS),JYL(JFS),XINT2,YINT2 )
 1020 CONTINUE
C-----IF NEITHER POINT OF INTERSECTION LIES ON BOTH THE ARC AND THE LINE
C-----THEN RETURN
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )         RETURN
C-----IF ONLY THE FIRST POINT OF INTERSECTION LIES ON BOTH THE ARC AND
C-----THE LINE THEN ADD THE FIRST POINT OF INTERSECTION
      IF ( (ITEST1.EQ.0.AND.ITEST2.EQ.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )         GO TO 2010
C-----IF ONLY THE SECOND POINT OF INTERSECTION LIES ON BOTH THE ARC AND
C-----THE LINE THEN ADD THE SECOND POINT OF INTERSECTION
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.EQ.0.AND.JTEST2.EQ.0) )         GO TO 3010
C-----IF THIS IS NOT THE MAIN INTERSECTION PATH THEN GO TO 4010
                    IF ( IBAND . NE . 1 )          GO TO 4010
C-----IF THE DISTANCE BETWEEN THE 2 POINTS OF CONFLICT ON THE MAIN
C-----INTERSECTION PATH IS LE ICLOSE THEN GO TO 4010
      X = DSQRT((XINT1-XINT2)**2+(YINT1-YINT2)**2)
                    IF ( X.LE.DFLOAT(ICLOSE) )    GO TO 4010
 2010 CONTINUE
C-----ADD FIRST POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = MRA(IFS,1)*DABS(DA1)*RADIAN + MLL(1) + XROUND
                    IF ( IFS .EQ . 1 )             GO TO 2020
      IL1 = IL1 + MAL(1)
 2020 CONTINUE
      IL2 = DSQRT((XINT1-IXL(JFS))**2+(YINT1-IYL(JFS))**2) + XROUND
                    IF ( JFS . EQ . 1 )            GO TO 2030
      IL2 = IL2 + LL1 + LA1 + LA2
 2030 CONTINUE
C-----IF THE SECOND POINT OF INTERSECTION DOES NOT LIE ON THE ARC OR
C-----THE LINE THEN RETURN
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ11,NPTH,IIA,IL2,AZ12,NC )
      IF ( JTEST1.NE.0 . OR . JTEST2.NE.0 )       RETURN
 3010 CONTINUE
C-----ADD THE SECOND POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = MRA(IFS,1)*DABS(DA2)*RADIAN + MLL(1) + XROUND
                    IF ( IFS . EQ . 1 )            GO TO 3020
      IL1 = IL1 + MAL(1)
 3020 CONTINUE
      IL2 = DSQRT((XINT2-IXL(JFS))**2+(YINT2-IYL(JFS))**2) + XROUND
                    IF ( JFS . EQ . 1 )            GO TO 3030
      IL2 = IL2 + LL1 + LA1 + LA2
 3030 CONTINUE
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ21,NPTH,IIA,IL2,AZ22,NC )
      RETURN
 4010 CONTINUE
C-----COMBINE THE 2 POINTS OF INTERSECTION AND CHECK AGAIN
      XINT1 = 0.5*(XINT1+XINT2)
      YINT1 = 0.5*(YINT1+YINT2)
      NUMPTS = 1
      GO TO 1010
      END                                                                ADDAL
```

```
      SUBROUTINE CATOAC (IFS,    IBAND,  JFS,     NC)                   COLEASE
C     TASK,CATOAC,IFS,IBAND,JFS,NC
      COMMON / PATH     / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),   COLEASE
     *                    JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1        ,   COLEASE
     *                    LA1        ,LA2        ,LL2        ,IIA        ,   COLEASE
     *                    IIL        ,IOA        ,IOL        ,IOPT       ,   COLEASE
     *                    ILCH       ,IBA    ( 2),IDA    ( 2),IRA    ( 2),   COLEASE
     *                    IPTURN     ,LENP       ,LIBL       ,LOBL       ,   COLEASE
     *                    LIMP       ,NGEOCP                                COLEASE
      COMMON / GEOCP    / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                    NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                    MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION    XINT1,YINT1,XINT2,YINT2
      COMMON / RADIAN /   PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION    PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD /   CONVAR(12),A,B,C,RADICL,R1,R1SQ,R2,R2SQ,X1,X2,
     *                    X2X1SQ,Y1,Y1SQ,Y2,Y2SQ,Y2Y1SQ,ZTEMPD(61)
      DOUBLE PRECISION    A,B,C,RADICL,R1,R1SQ,R2,R2SQ,X1,X2,X2X1SQ,Y1,
     *                    Y1SQ,Y2,Y2SQ,Y2Y1SQ
      DIMENSION           MSG917(8)
      DOUBLE PRECISION    XVAL
      DATA       MSG917 / 4H CIR,4HCLES,4H ARE,4H IDE,4HNTIC,4HAL -,
     *                    4H CAT,4HOAC  /
C
C-----SUBROUTINE CATOAC CHECKS FOR CONFLICTS BETWEEN THE ARC PORTION OF
C-----THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE
C-----INTERSECTION PATH BEING CHECKED AGAINST
C
      R1 = MRA(IFS,IBAND)
                     IF ( R1 . LE . ZERO )          RETURN
      R2 = IRA(JFS)
                     IF ( R2 . LE . ZERO )          RETURN
      X1 = MXA(IFS,IBAND)
      X2 = IXA(JFS)
      Y1 = MYA(IFS,IBAND)
      Y2 = IYA(JFS)
      X2X1SQ = (X2-X1)**2
      Y2Y1SQ = (Y2-Y1)**2
      Y1SQ = Y1**2
      Y2SQ = Y2**2
      R1SQ = R1**2
      R2SQ = R2**2
C-----CALCULATE THE POINT(S) OF INTERSECTION OF THE TWO ARCS
      A = 4.0*(X2X1SQ+Y2Y1SQ)
      B = 4.0*(Y2-Y1)*(R2SQ-R1SQ+Y1SQ-Y2SQ) -
     *    4.0*X2X1SQ*(Y1+Y2)
      C = ((R2SQ-R1SQ)-(Y2SQ-Y1SQ))**2 +
     *    X2X1SQ*(-2.0*R2SQ-2.0*R1SQ+2.0*Y1SQ+2.0*Y2SQ+X2X1SQ)
                     IF ( A . EQ . D0P0 )           GO TO 4010
      RADICL = B**2 - 4.0*A*C
                     IF ( DABS(RADICL).LE.ZERO )  GO TO 1010
                     IF ( RADICL )                5010 , 1010 , 2010
 1010 CONTINUE
C-----ONE Y COORDINATE FOR THE POINT(S) OF INTERSECTION
      YINT1 = -B/(2.0*A)
      YINT2 = YINT1
      RADICL = R1SQ - (YINT1-Y1)**2
                     IF ( DABS(RADICL).LE.ZERO )  GO TO 1020
                     IF ( RADICL )                5010 , 1020 , 1040
 1020 CONTINUE
C-----ONE X COORDINATE FOR ONE Y COORDINATE FOR THE POINT OF
C-----INTERSECTION
      XINT1 = X1
 1030 CONTINUE
C-----ADD 1 POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      CALL  ADDAA   ( IFS,IBAND,JFS,NC,1 )
      RETURN
 1040 CONTINUE
C-----POSSIBLY TWO X COORDINATES FOR ONE Y COORDINATE FOR THE POINTS OF
C-----INTERSECTION
      XINT1 = XVAL( X1,Y1,R1,X2,Y2,R2,RADICL,YINT1,+1,IFLAG )
                     IF ( IFLAG . NE . 0 )          GO TO 1050
      XINT2 = XVAL( X1,Y1,R1,X2,Y2,R2,RADICL,YINT2,-1,IFLAG )
                     IF ( IFLAG . NE . 0 )          GO TO 1030
                IF ( DABS(XINT1-XINT2) . LE . ZERO ) GO TO 1030
      GO TO 3010
 1050 CONTINUE
C-----THE FIRST X COORDINATE DOES NOT LIE ON EITHER ARC OF A CIRCLE THUS
C-----CHECK THE SECOND X COORDINATE
      XINT1 = XVAL( X1,Y1,R1,X2,Y2,R2,RADICL,YINT1,-1,IFLAG )
                     IF ( IFLAG . NE . 0 )          GO TO 5010
      GO TO 1030
 2010 CONTINUE
C-----TWO Y COORDINATES FOR THE POINT(S) OF INTERSECTION
      YINT1 = (-B+DSQRT(RADICL))/(2.0*A)
      YINT2 = (-B-DSQRT(RADICL))/(2.0*A)
      RADICL = R1SQ - (YINT1-Y1)**2
                     IF ( DABS(RADICL).LE.ZERO )  RADICL = D0P0
                     IF ( RADICL . LT . D0P0 )    GO TO 5010
C-----FIRST X COORDINATE FOR TWO Y COORDINATES FOR THE POINTS OF
C-----INTERSECTION
      XINT1 = XVAL( X1,Y1,R1,X2,Y2,R2,RADICL,YINT1,+1,IFLAG )
                     IF ( IFLAG . NE . 0 )        GO TO 5010
      RADICL = R1SQ - (YINT2-Y1)**2
                     IF ( DABS(RADICL).LE.ZERO )  RADICL = D0P0
                     IF ( RADICL . LT . D0P0 )    GO TO 5010
C-----SECOND X COORDINATE FOR TWO Y COORDINATES FOR THE POINTS OF
C-----INTERSECTION
      XINT2 = XVAL( X1,Y1,R1,X2,Y2,R2,RADICL,YINT2,+1,IFLAG )
                     IF ( IFLAG . NE . 0 )        GO TO 5010
 3010 CONTINUE
C-----ADD TWO POINTS OF INTERSECTION AS INTERSECTION CONFLICTS
      CALL  ADDAA   ( IFS,IBAND,JFS,NC,2 )
      RETURN
 4010 CONTINUE
C-----BOTH OF THE ARCS HAVE THE SAME CENTER COORDINATES
                     IF ( DABS(R1-R2).GT.ZERO )   GO TO 5010
      GO TO 9170
 5010 CONTINUE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9170 CONTINUE
      CALL  ABORTR  ( MSG917,31 )
      STOP  917
      END                                                            CATOAC
```

```
      DOUBLE PRECISION
     *FUNCTION   XVAL    ( X1,Y1,R1,X2,Y2,R2,RADICL,YVAL,IISIGN,IFLAG )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(44),RA,RB,ZTEMPD(57)
      DOUBLE PRECISION  RA,RB
      DOUBLE PRECISION  RADICL,R1,R2,X1,X2,YVAL,Y1,Y2
C
C-----FUNCTION XVAL FINDS THE X COORDINATE OF THE INTERSECTION OF TWO
C-----ARCS FOR A GIVEN YVAL COORDINATE (IFLAG=0=OK AND IFLAG=1=NOT ON
C-----EITHER ARC OF CIRCLE
C
      IFLAG = 1
C-----FIND ONE OF THE VALUES FOR XVAL AND CHECK IF IT IS ON BOTH ARCS
      XVAL = X1 + IISIGN*DSQRT(RADICL)
C-----IF THE DISTANCE FROM (XVAL,YVAL) TO (X1,Y1) IS NOT R1 THEN XVAL IS
C-----NOT ON ARC 1 AND THE OTHER VALUE FOR XVAL SHOULD BE USED
      RA = DSQRT((XVAL-X1)**2+(YVAL-Y1)**2)
                      IF ( DABS(RA-R1).GT.ZERO )    GO TO 1010
C-----IF THE DISTANCE FROM (XVAL,YVAL) TO (X2,Y2) IS NOT R2 THEN XVAL IS
C-----NOT ON ARC 2 AND THE OTHER VALUE FOR XVAL SHOULD BE USED
      RB = DSQRT((XVAL-X2)**2+(YVAL-Y2)**2)
                      IF ( DABS(RB-R2).GT.ZERO )    GO TO 1010
      IFLAG = 0
      RETURN
 1010 CONTINUE
C-----FIND THE OTHER VALUE FOR XVAL AND CHECK IF IT IS ON BOTH ARCS
      XVAL = X1 - IISIGN*DSQRT(RADICL)
C-----IF THE DISTANCE FROM (XVAL,YVAL) TO (X1,Y1) IS NOT R1 THEN XVAL IS
C-----NOT ON ARC 1 THUS RETURN (IFLAG=1)
      RA = DSQRT((XVAL-X1)**2+(YVAL-Y1)**2)
                      IF ( DABS(RA-R1).GT.ZERO )    RETURN
C-----IF THE DISTANCE FROM (XVAL,YVAL) TO (X2,Y2) IS NOT R2 THEN XVAL IS
C-----NOT ON ARC 2 THUS RETURN (IFLAG=1)
      RB = DSQRT((XVAL-X2)**2+(YVAL-Y2)**2)
                      IF ( DABS(RB-R2).GT.ZERO )    RETURN
      IFLAG = 0
      RETURN
      END                                                          XVAL
```

```
      SUBROUTINE ADDAA  (IFS,    IBAND,  JFS,    NC,      NUM)          COLEASE
C     TASK,ADDAA,IFS,IBAND,JFS,NC,NUM
      COMMON / PATH   / IGEOCP(60),IXL   ( 2),IYL   ( 2),JXL   ( 2),    COLEASE
     *                  JYL   ( 2),IXA   ( 2),IYA   ( 2),LL1        ,   COLEASE
     *                  LA1        ,LA2        ,LL2        ,IIA        ,  COLEASE
     *                  IIL        ,IOA        ,IOL        ,IOPT       ,  COLEASE
     *                  ILCH       ,IBA   ( 2),IDA   ( 2),IRA   ( 2),   COLEASE
     *                  IPTURN     ,LENP       ,LIBL       ,LOBL       ,  COLEASE
     *                  LIMP       ,NGEOCP                               COLEASE
      COMMON / GEOCP  / XINT1,YINT1,XINT2,YINT2,MXL(2,5),MYL(2,5),
     *                  NXL(2,5),NYL(2,5),MXA(2,5),MYA(2,5),MBA(2,5),
     *                  MDA(2,5),MRA(2,5),MLL(2),MAL(2),MPTH,NPTH,MIA
      DOUBLE PRECISION  XINT1,YINT1,XINT2,YINT2
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / ZTEMPD / CONVAR(44),AZIM11,AZIM12,AZIM21,AZIM22,AZ11,
     *                  AZ12,AZ21,AZ22,DA11,DA12,DA21,DA22,X,XBEAR1,
     *                  XBEAR2,YBEAR1,YBEAR2,IL1,IL2,ITEST1,ITEST2,
     *                  JTEST1,JTEST2,NUMPTS,ZTEMPD(20)
      DOUBLE PRECISION  AZIM11,AZIM12,AZIM21,AZIM22,AZ11,AZ12,AZ21,AZ22,
     *                  DA11,DA12,DA21,DA22,X,XBEAR1,XBEAR2,YBEAR1,
     *                  YBEAR2
      DOUBLE PRECISION  AZIM36
C
C-----SUBROUTINE ADDAA ADDS INTERSECTION CONFLICTS BETWEEN THE ARC
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST
C
      NUMPTS = NUM
 1010 CONTINUE
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED
      XBEAR1 = XINT1 - MXA(IFS,IBAND)
      YBEAR1 = YINT1 - MYA(IFS,IBAND)
      AZIM11 = AZIM36( YBEAR1,XBEAR1 )
      AZ11 = AZIM11 + ISIGN(90,MDA(IFS,IBAND))
      ITEST1 = ICHKA( AZIM11,MBA(IFS,IBAND),MDA(IFS,IBAND),DA11 )
C-----CHECK IF THE FIRST POINT OF INTERSECTION LIES ON THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST
      XBEAR2 = XINT1 - IXA(JFS)
      YBEAR2 = YINT1 - IYA(JFS)
      AZIM12 = AZIM36( YBEAR2,XBEAR2 )
      AZ12 = AZIM12 + ISIGN(90,IDA(JFS))
      ITEST2 = ICHKA( AZIM12,IBA(JFS),IDA(JFS),DA12 )
      JTEST1 = 1
      JTEST2 = 1
                      IF ( NUMPTS . EQ . 1 )        GO TO 1020
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED
      XBEAR1 = XINT2 - MXA(IFS,IBAND)
      YBEAR1 = YINT2 - MYA(IFS,IBAND)
      AZIM21 = AZIM36( YBEAR1,XBEAR1 )
      AZ21 = AZIM21 + ISIGN(90,MDA(IFS,IBAND))
      JTEST1 = ICHKA( AZIM21,MBA(IFS,IBAND),MDA(IFS,IBAND),DA21 )
C-----CHECK IF THE SECOND POINT OF INTERSECTION LIES ON THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST
      XBEAR2 = XINT2 - IXA(JFS)
      YBEAR2 = YINT2 - IYA(JFS)
      AZIM22 = AZIM36( YBEAR2,XBEAR2 )
      AZ22 = AZIM22 + ISIGN(90,IDA(JFS))
      JTEST2 = ICHKA( AZIM22,IBA(JFS),IDA(JFS),DA22 )
 1020 CONTINUE
C-----IF NEITHER POINT OF INTERSECTION LIES ON BOTH THE ARC PORTION OF
C-----THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE
C-----INTERSECTION PATH BEING CHECKED AGAINST THEN RETURN
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )         RETURN
C-----IF ONLY THE FIRST POINT OF INTERSECTION LIES ON BOTH THE ARC
```

```
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST THEN ADD THE FIRST
C-----POINT OF INTERSECTION
      IF ( (ITEST1.EQ.0.AND.ITEST2.EQ.0) . AND .
     *     (JTEST1.NE.0.OR .JTEST2.NE.0) )          GO TO 2010
C-----IF ONLY THE SECOND POINT OF INTERSECTION LIES ON BOTH THE ARC
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST THEN ADD THE SECOND
C-----POINT OF INTERSECTION
      IF ( (ITEST1.NE.0.OR .ITEST2.NE.0) . AND .
     *     (JTEST1.EQ.0.AND.JTEST2.EQ.0) )          GO TO 3010
C-----IF THIS IS NOT THE MAIN INTERSECTION PATH THEN GO TO 4010
                          IF ( IBAND . NE . 1 )     GO TO 4010
C-----IF THE DISTANCE BETWEEN THE 2 POINTS OF CONFLICT ON THE MAIN
C-----INTERSECTION PATH IS LE ICLOSE THEN GO TO 4010
      X = DSQRT((XINT1-XINT2)**2+(YINT1-YINT2)**2)
                          IF ( X.LE.DFLOAT(ICLOSE) )   GO TO 4010
 2010 CONTINUE
C-----ADD FIRST POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = MRA(IFS,1)*DABS(DA11)*RADIAN + MLL(1) + XROUND
                          IF ( IFS . EQ . 1 )       GO TO 2020
      IL1 = IL1 + MAL(1)
 2020 CONTINUE
      IL2 = IRA(JFS)*DABS(DA12)*RADIAN + LL1 + XROUND
                          IF ( JFS .EQ. 1 )         GO TO 2030
      IL2 = IL2 + LA1
 2030 CONTINUE
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ11,NPTH,IIA,IL2,AZ12,NC )
C-----IF THE SECOND POINT OF INTERSECTION DOES NOT LIE ON THE ARC
C-----PORTION OF THE INTERSECTION PATH BEING CHECKED OR THE ARC PORTION
C-----OF THE INTERSECTION PATH BEING CHECKED AGAINST THEN RETURN
      IF ( JTEST1.NE.0 . OR . JTEST2.NE.0 )        RETURN
 3010 CONTINUE
C-----ADD THE SECOND POINT OF INTERSECTION AS AN INTERSECTION CONFLICT
      IL1 = MRA(IFS,1)*DABS(DA21)*RADIAN + MLL(1) + XROUND
                          IF ( IFS . EQ . 1 )       GO TO 3020
      IL1 = IL1 + MAL(1)
 3020 CONTINUE
      IL2 = IRA(JFS)*DABS(DA22)*RADIAN + LL1 + XROUND
                          IF ( JFS . EQ . 1 )       GO TO 3030
      IL2 = IL2 + LA1
 3030 CONTINUE
      CALL  ADDCON  ( MPTH,MIA,IL1,AZ21,NPTH,IIA,IL2,AZ22,NC )
      RETURN
 4010 CONTINUE
C-----COMBINE THE 2 POINTS OF INTERSECTION AND CHECK AGAIN
      XINT1 = 0.5*(XINT1+XINT2)
      YINT1 = 0.5*(YINT1+YINT2)
      NUMPTS = 1
      GO TO 1010
      END                                                               ADDAA
```

```
      SUBROUTINE SRTCON                                                 COLEASE
C     TASK,SRTCON
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),IDUMCO                               COLEASE
      COMMON / PATH   / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2), COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1     ,   COLEASE
     *                  LA1       ,LA2       ,LL2       ,IIA       ,    COLEASE
     *                  IIL       ,IOA       ,IOL       ,IOPT      ,    COLEASE
     *                  ILCH      ,IBA    ( 2),IDA    ( 2),IRA    ( 2), COLEASE
     *                  IPTURN    ,LENP      ,LIBL      ,LOBL      ,    COLEASE
     *                  LIMP      ,NGEOCP                               COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / ZTEMPD / I,ICON,IDIST(60),IPN,IPTH,ITEMP,J,JCON,
     *                  ZTEMPD(38)
C
C-----SUBROUTINE SRTCON SORTS THE INTERSECTION CONFLICTS FOR EACH
C-----INTERSECTION PATH BY THE DISTANCE DOWN THE INTERSECTION PATH TO
C-----THE INTERSECTION CONFLICT
C
C-----PROCESS EACH INTERSECTION PATH
      DO 3020  IPTH = 1 , NPATHS
C     COLEASE,EXTRAC,PATH,IPTH
      CALL  EXTRAC  (     6,IPTH   )                                    COLEASE
                          IF ( NGEOCP . LE . 1 )      GO TO 3020
C-----FIND THE DISTANCE DOWN THE INTERSECTION PATH TO EACH INTERSECTION
C-----CONFLICT AND TEMPORARILY STORE IN ARRAY IDIST
      DO 1010  ICON = 1 , NGEOCP
      JCON = IGEOCP(ICON)
C     COLEASE,EXTRAC,CONFLT,JCON
      CALL  EXTRAC  (     3,JCON   )                                    COLEASE
      IPN = 1
                          IF ( ICONP(2) . EQ . IPTH )  IPN = 2
      IDIST(ICON) = ICOND(IPN)
 1010 CONTINUE
C-----SORT THE DISTANCE DOWN THE INTERSECTION PATH TO THE INTERSECTION
C-----CONFLICT (IDIST) AND CARRY ARRAY IGEOCP FROM ENTRY IPTH OF ENTITY
C-----PATH USING A BUBBLE SORT
C-----SET THE SORT INDEX TO THE SECOND ELEMENT IN THE LIST
      I = 2
 2010 CONTINUE
C-----IF THE SORT INDEX IS GT THE NUMBER IN THE LIST THEN THE SORT IS
C-----FINISHED
                          IF ( I . GT . NGEOCP )      GO TO 3010
C-----IF THE ELEMENT IS OUT OF ORDER THEN BUBBLE IT UP TO ITS PROPER
C-----POSITION IN THE LIST
             IF ( IDIST(I) . LT . IDIST(I-1) )     GO TO 2020
C-----CHECK THE NEXT ELEMENT DOWN THE LIST
      I = I + 1
      GO TO 2010
 2020 CONTINUE
C-----SAVE THE INDEX OF THE NEXT ELEMENT TO BE CHECKED AFTER THIS
C-----ELEMENT HAS BEEN BUBBLED TO ITS PROPER POSITION IN THE LIST
      J = I + 1
 2030 CONTINUE
C-----SWAP ELEMENT I AND ELEMENT I-1 OF ARRAY IDIST AND IGEOCP
      ITEMP         = IGEOCP(I-1)
      IGEOCP(I-1)   = IGEOCP(I)
      IGEOCP(I)     = ITEMP
      ITEMP         = IDIST(I-1)
      IDIST(I-1)    = IDIST(I)
      IDIST(I)      = ITEMP
C-----CHECK NEXT ELEMENT ABOVE TO SEE IF THE ELEMENT HAS BEEN BUBBLED TO
C-----ITS PROPER POSITION IN THE LIST
      I = I - 1
C-----IF THE START OF THE LIST HAS BEEN REACHED THEN END BUBBLING THIS
C-----ELEMENT
                          IF ( I . EQ . 1 )           GO TO 2040
C-----IF THE ELEMENT IS STILL NOT IN ITS PROPER POSITION IN THE LIST
C-----THEN SWAP THE ELEMENTS AND CHECK AGAIN
             IF ( IDIST(I) . LT . IDIST(I-1))      GO TO 2030
```

```
 2040 CONTINUE
C-----SET THE INDEX TO THE NEXT ELEMENT TO BE CHECKED AND START CHECKING
C-----DOWN THE LIST AGAIN
      I = J
      GO TO 2010
 3010 CONTINUE
C-----STORE THE SORTED IGEOCP ARRAY BACK INTO ENTRY IPTH OF ENTITY PATH
C   COLEASE,REPACK,PATH,IPTH
      CALL  REPACK  (      6,IPTH  )                                         COLEASE
C-----END OF INTERSECTION PATH LOOP
 3020 CONTINUE
      RETURN
      END                                                                SRTCON
```

```
      SUBROUTINE WRITPA                                                  COLEASE
C   TASK,WRITPA
      COMMON / PATH   / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2),    COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1        ,    COLEASF
     *                  LA1        ,LA2        ,LL2        ,IIA        ,    COLEASE
     *                  IIL        ,IOA        ,IOL        ,IOPT       ,    COLEASE
     *                  ILCH       ,IBA    ( 2),IDA    ( 2),IRA    ( 2),    COLEASE
     *                  IPTURN     ,LENP       ,LIBL       ,LOBL       ,    COLEASE
     *                  LIMP       ,NGEOCP                                  COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / I,J,LTEST,ZTEMPD(102)
  601 FORMAT(8X,5HTABLE,I3,21H  -  LISTING OF PATHS,//)
  602 FORMAT(20I4)
  603 FORMAT(12X,4HPATH,I4,15H GOES FROM LANE,I2,12H OF APPROACH,I3,
     *       8H TO LANE,I2,12H OF APPROACH,I3,/,
     *       15X,16HLENGTH OF PATH =,I4,25H FEET AND SPEED OF PATH =,I3,
     *       16H FEET PER SECOND,/,
     *       15X,21HNUMBER OF CONFLICTS =,I3,23H AND TURN CODE FOR PATH,
     *       3H IS)
  604 FORMAT(1H+,65X,6HU-TURN)
  605 FORMAT(1H+,65X,4HLEFT)
  606 FORMAT(1H+,65X,8HSTRAIGHT)
  607 FORMAT(1H+,65X,5HRIGHT)
  608 FORMAT(15X,48HCONFLICT ENTRY NUMBERS ORDERED BY DISTANCE DOWN ,
     *       13HTHIS PATH ARE)
  609 FORMAT(18X,10I5)
  610 FORMAT(/)
  611 FORMAT(12X,34HTOTAL NUMBER OF PATHS CALCULATED =,I4,//)
C
C-----SUBROUTINE WRITPA WRITES THE INTERSECTION PATH INFORMATION ONTO
C-----TAPE MODELT FOR SIMPRO
C
                    IF ( NLINE+15.GT.LINES )      CALL  HEADER
      PRINT 601 , NTABL
      NTABL = NTABL + 1
      NLINE = NLINE + 3
      WRITE (MODELT,602)  NPATHS
C-----WRITE THE INFORMATION FOR EACH INTERSECTION PATH
      DO 1020  I = 1 , NPATHS
C   COLEASE,EXTRAC,PATH,I
      CALL  EXTRAC  (      6,I      )                                    COLEASE
C-----UN-BIAS THE INTERSECTION PATH ATTRIBUTES
      IXA(1) = IXA(1) - 900
      IXA(2) = IXA(2) - 900
      IYA(1) = IYA(1) - 900
      IYA(2) = IYA(2) - 900
      IDA(1) = IDA(1) - 360
      IDA(2) = IDA(2) - 360
C-----WRITE THE INTERSECTION PATH INFORMATION
      WRITE (MODELT,602)  IIA,IIL,IOA,IOL,
     *                    IXL(1),IYL(1),LL1,JXL(1),JYL(1),
     *                    IXA(1),IYA(1),LA1,IRA(1),IBA(1),IDA(1),
     *                    IXA(2),IYA(2),LA2,IRA(2),IBA(2),IDA(2),
     *                    IXL(2),IYL(2),LL2,JXL(2),JYL(2),
     *                    LENP,IPTURN,LIMP,IOPT,ILCH,LIBL,LOBL,NGEOCP
      LTEST = NLINE + 6 + (NGEOCP+9)/10
                    IF ( I . EQ . NPATHS )        LTEST = LTEST + 3
                    IF ( LTEST . GT . LINES )     CALL  HEADER
      PRINT 603 , I,IIL,IIA,IOL,IOA,LENP,LIMP,NGEOCP
                    IF ( IPTURN . EQ . 8 )        PRINT 604
                    IF ( IPTURN . EQ . 4 )        PRINT 605
                    IF ( IPTURN . EQ . 2 )        PRINT 606
                    IF ( IPTURN . EQ . 1 )        PRINT 607
      NLINE = NLINE + 3
                    IF ( NGEOCP . LE . 0 )        GO TO 1010
      WRITE (MODELT,602)  (IGEOCP(J),J=1,NGEOCP)
      PRINT 608
      PRINT 609 , (IGEOCP(J),J=1,NGEOCP)
      NLINE = NLINE + 1 + (NGEOCP+9)/10
```

```
 1010 CONTINUE
      PRINT 610
      NLINE = NLINE + 2
C-----END OF INTERSECTION PATH LOOP
 1020 CONTINUE
      PRINT 611 , NPATHS
      NLINE = NLINE + 3
      RETURN
      END                                                              WRITPA
```

```
      SUBROUTINE NDXCON                                                COLEASE
C   TASK,NDXCON
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,   COLEASE
     *                  ICONI ( 2),IDUMCO                              COLEASE
      COMMON / PATH   / IGEOCP(60),IXL   ( 2),IYL   ( 2),JXL   ( 2),   COLEASE
     *                  JYL   ( 2),IXA   ( 2),IYA   ( 2),LL1       ,   COLEASE
     *                  LA1       ,LA2       ,LL2       ,IIA       ,   COLEASE
     *                  IIL       ,IOA       ,IOL       ,IOPT      ,   COLEASE
     *                  ILCH      ,IBA   ( 2),IDA   ( 2),IRA   ( 2),   COLEASE
     *                  IPTURN    ,LENP      ,LIBL      ,LOBL      ,   COLEASE
     *                  LIMP      ,NGEOCP                              COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / ZTEMPD / ICON,IPTH,I12,JCON,ZTEMPD(101)
      DIMENSION         MSG918(14)
      DATA      MSG918 / 4H CON,4HFLIC,4HT WA,4HS NO,4HT FO,4HUND ,
     *                  4HON I,4HGEOC,4HP LI,4HST F,4HOR P,4HATH ,
     *                  4H- ND,4HXCON /
C
C-----SUBROUTINE NDXCON CROSS INDEXES THE INTERSECTION CONFLICTS WITH
C-----THE INTERSECTION PATHS
C
C-----PROCESS EACH INTERSECTION CONFLICT
      DO 2010  ICON = 1 , NCONFS
C   COLEASE,EXTRAC,CONFLT,ICON
      CALL  EXTRAC  (      3,ICON  )                                   COLEASE
C-----PROCESS EACH INTERSECTION PATH INVOLVED IN THE INTERSECTION
C-----CONFLICT
      DO 1030  I12 = 1 , 2
      IPTH = ICONP(I12)
C   COLEASE,EXTRAC,PATH,IPTH
      CALL  EXTRAC  (      6,IPTH  )                                   COLEASE
                    IF ( NGEOCP . LE . 0 )       GO TO 9180
C-----SEARCH EACH INTERSECTION CONFLICT FOR THIS INTERSECTION PATH AND
C-----FIND INTERSECTION CONFLICT ICON ON THE IGEOCP ARRAY
      DO 1010  JCON = 1 , NGEOCP
                    IF ( ICON.EQ.IGEOCP(JCON) )  GO TO 1020
 1010 CONTINUE
      GO TO 9180
 1020 CONTINUE
C-----SAVE THE INDEX JCON FOR THIS INTERSECTION CONFLICT
      ICONI(I12) = JCON
C-----END OF INTERSECTION PATH LOOP
 1030 CONTINUE
C-----STORE THE ICONI ARRAY IN ENTRY ICON OF ENTITY CONFLT
C   COLEASE,REPACK,CONFLT,ICON
      CALL  REPACK  (      3,ICON  )                                   COLEASE
C-----END OF INTERSECTION CONFLICT LOOP
 2010 CONTINUE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9180 CONTINUE
      CALL  ABORTR  ( MSG918,56 )
      STOP  918
      END                                                              NDXCON
```

```
      SUBROUTINE WRITCO                                                  COLEASE
C     TASK,WRITCO
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,     COLEASE
     *                  ICONI ( 2),IDUMCO                                COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / ZTEMPD / IADD,ICON,ZTEMPD(103)
  601 FORMAT(8X,5HTABLE,I3,25H  -  LISTING OF CONFLICTS,//,
     *       12X,40HCONFLICT  PATH1 PATH2  APPR1 APPR2  DIST,
     *       29H1 DIST2  ANGLE  INDEX1 INDEX2,/)
  602 FORMAT(20I4)
  603 FORMAT(12X,I5,2X,2I6,1X,2I6,1X,2I6,I8,2I7)
  604 FORMAT(//,12X,27HTOTAL NUMBER OF CONFLICTS =,I5,//)
C
C-----SUBROUTINE WRITCO WRITES THE INTERSECTION CONFLICT INFORMATION
C-----ONTO TAPE MODELT FOR SIMPRO
C
                      IF ( NLINE+10 . GT . LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 4
      NTABL = NTABL + 1
      WRITE (MODELT,602)  NCONFS
      IADD = 1
C-----WRITE THE INFORMATION FOR EACH INTERSECTION CONFLICT
      DO 1010  ICON = 1 , NCONFS
C     COLEASE,EXTRAC,CONFLT,ICON
      CALL  EXTRAC  (      3,ICON  )                                     COLEASE
C-----WRITE THE INTERSECTION CONFLICT INFORMATION
      WRITE (MODELT,602)  ICONP,ICONA,ICOND,ICONAN,ICONI
                      IF ( ICON . GT . NCONFS-4 )  IADD = NCONFS-ICON+6
                      IF ( NLINE+IADD.GT.LINES )   CALL  HEADER
      PRINT 603 , ICON,ICONP,ICONA,ICOND,ICONAN,ICONI
      NLINE = NLINE + 1
 1010 CONTINUE
      PRINT 604 , NCONFS
      NLINE = NLINE + 5
      RETURN
      END                                                                WRITCO
```

```
      SUBROUTINE XROTX  ( X,Y,IAZIM,RX,RY )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  COSA,RX,RY,SINA,X,Y
      DATA      COSA   / 1.0D+00 /
      DATA      IALAST / 0 /
      DATA      SINA   / 0.0D+00 /
C
C-----SUBROUTINE XROTX ROTATES A REAL VECTOR BY AN AZIMUTH AND RETURNS
C-----A REAL VECTOR
C
                      IF ( IAZIM . EQ . IALAST )   GO TO 1010
      SINA = DSIN(IAZIM*RADIAN)
      COSA = DCOS(IAZIM*RADIAN)
 1010 CONTINUE
      RX =  X*COSA + Y*SINA
      RY = -X*SINA + Y*COSA
      IALAST = IAZIM
      RETURN
      END                                                                XROTX
```

```
      SUBROUTINE XROTI  ( X,Y,IAZIM,IRX,IRY )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  RX,RY,X,Y
C
C-----SUBROUTINE XROTI ROTATES A REAL VECTOR BY AN AZIMUTH AND RETURNS
C-----AN INTEGER VECTOR
C
      CALL  XROTX   ( X,Y,IAZIM,RX,RY )
      IRX = SIGN( SNGL(DABS(RX)+XROUND),SNGL(RX) )
      IRY = SIGN( SNGL(DABS(RY)+XROUND),SNGL(RY) )
                     IF ( DABS(RX) . LT . XROUND )IRX = 0
                     IF ( DABS(RY) . LT . XROUND )IRY = 0
      RETURN
      END                                                              XROTI
```

```
      SUBROUTINE IROTX  ( IX,IY,IAZIM,RX,RY )
      DOUBLE PRECISION  RX,RY,X,Y
C
C-----SUBROUTINE IROTX ROTATES AN INTEGER VECTOR BY AN AZIMUTH AND
C-----RETURNS A REAL VECTOR
C
      X = IX
      Y = IY
      CALL  XROTX   ( X,Y,IAZIM,RX,RY )
      RETURN
      END                                                              IROTX
```

```
      SUBROUTINE XROTAX ( X,Y,IAZIM,IAX,IAY,RX,RY )
      DOUBLE PRECISION  RX,RY,X,Y
C
C-----SUBROUTINE XROTAX ROTATES A REAL VECTOR BY AN AZIMUTH, ADDS AN
C-----INTEGER COORDINATE, AND RETURNS A REAL COORDINATE
C
      CALL  XROTX   ( X,Y,IAZIM,RX,RY )
      RX = IAX + RX
      RY = IAY + RY
      RETURN
      END                                                       XROTAX
```

```
      SUBROUTINE XROTAI ( X,Y,IAZIM,IAX,IAY,IRX,IRY )
      DOUBLE PRECISION  X,Y
C
C-----SUBROUTINE XROTAI ROTATES A REAL VECTOR BY AN AZIMUTH, ADDS AN
C-----INTEGER COORDINATE, AND RETURNS AN INTEGER COORDINATE
C
      CALL  XROTI   ( X,Y,IAZIM,IRX,IRY )
      IRX = IAX + IRX
      IRY = IAY + IRY
      RETURN
      END                                                       XROTAI
```

```
      SUBROUTINE IROTAX ( IX,IY,IAZIM,IAX,IAY,RX,RY )
      DOUBLE PRECISION  RX,RY
C
C-----SUBROUTINE IROTAX ROTATES AN INTEGER VECTOR BY AN AZIMUTH, ADDS AN
C-----INTEGER COORDINATE, AND RETURNS A REAL COORDINATE
C
      CALL  IROTX   ( IX,IY,IAZIM,RX,RY )
      RX = IAX + RX
      RY = IAY + RY
      RETURN
      END                                                              IROTAX
```

```
      DOUBLE PRECISION
     *FUNCTION   AZIM36 ( Y,X )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  ATAN36,X,Y
C
C-----FUNCTION AZIM36 FINDS THE ARC TANGENT OF A COORDINATE AND RETURNS
C-----THE AZIMUTH FROM 0 TO 360 DEGREES (NORTH ZERO AND CLOCKWISE
C-----POSITIVE)
C
      AZIM36 = ATAN36( X,Y ) / RADIAN
      RETURN
      END                                                              AZIM36
```

```
      DOUBLE PRECISION
     *FUNCTION   ATAN36 ( Y,X )
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  X,Y
C
C-----FUNCTION ATAN36 FINDS THE ARC TANGENT OF A COORDINATE AND RETURNS
C-----THE ANGLE FROM 0 TO 360 DEGREES (EAST ZERO AND COUNTER-CLOCKWISE
C-----POSITIVE)
C
      ATAN36 = 2*PI
            IF ( Y.EQ.D0P0.AND.X.GE.D0P0 )       ATAN36 = D0P0
            IF ( X.EQ.D0P0.AND.Y.GT.D0P0 )       ATAN36 = 0.5*PI
            IF ( Y.EQ.D0P0.AND.X.LT.D0P0 )       ATAN36 = PI
            IF ( X.EQ.D0P0.AND.Y.LT.D0P0 )       ATAN36 = 1.5*PI
                    IF ( ATAN36 . NE . 2.0*PI )  RETURN
      ATAN36 = DATAN( Y/X )
                    IF ( X . LT . D0P0 )         ATAN36 = ATAN36 + PI
            IF ( X.GT.D0P0.AND.Y.LT.D0P0 )       ATAN36 = ATAN36 + 2*PI
      RETURN
      END                                                               ATAN36
```

```
      DOUBLE PRECISION
     *FUNCTION   DTAN    ( VAL )
      DOUBLE PRECISION  VAL
C
C-----FUNCTION DTAN FINDS THE DOUBLE PRECISION TANGENT OF VAL
C
      DTAN = DSIN(VAL)/DCOS(VAL)
      RETURN
      END
```

```
      SUBROUTINE ABORTR (MSG,   NCHS)                                   COLEASE
TASK,ABORTR,MSG,NCHS
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW    ,    COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),IDUMCO                               COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2           COLEASE
      COMMON / NOATTB / NOATTB( 7)                                      COLEASE
      COMMON / PATH   / IGEOCP(60),IXL   ( 2),IYL   ( 2),JXL   ( 2),    COLEASE
     *                  JYL   ( 2),IXA   ( 2),IYA   ( 2),LL1       ,    COLEASE
     *                  LA1       ,LA2       ,LL2       ,IIA       ,    COLEASE
     *                  IIL       ,IOA       ,IOL       ,IOPT      ,    COLEASE
     *                  ILCH      ,IBA   ( 2),IDA   ( 2),IRA   ( 2),    COLEASE
     *                  IPTURN    ,LENP      ,LIBL      ,LOBL      ,    COLEASE
     *                  LIMP      ,NGEOCP                               COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / ATTB   / IAT(3,  200)
      COMMON / ENTITY / IEN(9,    7)
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / GEOVAL / SCALEA,SCALEI,RADIUS,IPATH,IPLOT,ISAME,ICLOSE,
     *                  IPAPER,IXAPP(50),IYAPP(50)
      DOUBLE PRECISION  SCALEA,SCALEI,RADIUS
      COMMON / INDEX  / IAN,IA,ILN,IL,NLANEI,JAN,JA,JLN,JL,NLANEJ
      COMMON / OUTPUT / NPAGE,NLINE,NTABL,LINES,MODELT
      COMMON / PLOTTR / XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI,MINXA,MINYA,MAXXA,
     *                  MAXYA,MINXI,MINYI,MAXXI,MAXYI,LTDIRX(50),
     *                  LTDIRY(50)
      DOUBLE PRECISION  XMIN,YMIN,XMAX,YMAX,X0,Y0,XSIZEA,YSIZEA,XSIZEI,
     *                  YSIZEI,SCALE,CSIZEA,CSIZEI
      COMMON / RADIAN / PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      DOUBLE PRECISION  PI,RADIAN,XROUND,FPSMPH,ZERO,D0P0
      COMMON / SDRC   / IXSDRC(20),IYSDRC(20),NSDRC,LSDRC(20)
      COMMON / ZTEMPD / I,ICHS,J,MSGPP(9),NUM,NWDS,ZTEMPD(91)
      DIMENSION         COM01(1),COM02(1),COM03(1),COM04(1),COM05(1),
     *                  COM06(1),COM07(1),COM08(1),COM09(1),COM10(1),
     *                  COM11(1),COM12(1),COM13(1),COM14(1)
      DIMENSION         D01(2),D02(2),D03(2),D04(2),D05(2),D06(2),
     *                  D07(2),D08(2),D09(2),D10(2),D11(2),D12(2),
     *                  D13(2),D14(2),D15(2),D16(2),D17(2),D18(2),
     *                  D19(2),D20(2),D21(2),D22(2)
      DIMENSION         IC(2,16),MSG(1)
      DIMENSION         NCOM01(2,026),NCOM02(2,006),NCOM03(2,010),
     *                  NCOM04(2,020),NCOM05(2,004),NCOM06(2,094),
     *                  NCOM07(2,040),NCOM08(2,142),NCOM09(2,108),
     *                  NCOM10(2,010),NCOM11(2,005),NCOM12(2,121),
     *                  NCOM13(2,006),NCOM14(2,061)
      EQUIVALENCE       (IALEFT   ,COM01(1)),(IARCX    ,COM02(1)),
     *                  (ICONP (1),COM03(1)),(LWID     ,COM04(1)),
     *                  (ILX1     ,COM05(1)),(IGEOCP(1),COM06(1)),
     *                  (ICANSE(1),COM07(1)),(NIBA     ,COM08(1)),
     *                  (SCALEA   ,COM09(1)),(IAN      ,COM10(1)),
     *                  (NPAGE    ,COM11(1)),(XMIN     ,COM12(1)),
     *                  (PI       ,COM13(1)),(IXSDRC(1),COM14(1))
      EQUIVALENCE       (SCALEA,D01(1)),(SCALEI,D02(1)),
     *                  (RADIUS,D03(1)),(XMIN  ,D04(1)),
     *                  (YMIN  ,D05(1)),(XMAX  ,D06(1)),
     *                  (YMAX  ,D07(1)),(X0    ,D08(1)),
     *                  (Y0    ,D09(1)),(XSIZEA,D10(1)),
     *                  (YSIZEA,D11(1)),(XSIZEI,D12(1)),
     *                  (YSIZEI,D13(1)),(SCALE ,D14(1)),
     *                  (CSIZEA,D15(1)),(CSIZEI,D16(1)),
     *                  (PI    ,D17(1)),(RADIAN,D18(1)),
     *                  (XROUND,D19(1)),(FPSMPH,D20(1)),
     *                  (ZERO  ,D21(1)),(D0P0  ,D22(1))
      DATA      NCOM01 / 4HIALE,2HFT,4HIARG,2HHT,4HNLAN,2HES,4HLLAN,2HES,
     *                  10*1H       ,4HIAPX,2H  ,4HIAPY,2H  ,4HISLI,2HM ,
     *                  4HNSDR,2H  ,4HISDR,2HN ,8*1H        ,4HISDR,2HA ,
     *                  8*1H        ,4HIAAZ,2HIM,4HNDEG,2HST,4HNDEG,2HUT/
      DATA      NCOM02 / 4HIARC,2HX ,4HIARC,2HY ,4HIARC,2HAZ,4HIARC,2HSW,
     *                  4HIARC,2HR ,4HIDUM,2HAR/
      DATA      NCOM03 / 4HICON,2HP ,2*1H        ,4HICON,2HA ,2*1H        ,
     *                  4HICON,2HD ,2*1H        ,4HICON,2HAN,4HICON,2HI ,
     *                  2*1H        ,4HIDUM,2HCO/
      DATA      NCOM04 / 4HLWID,2H  ,4HNLL ,2H  ,4HNLR ,2H  ,4HISNA,2H  ,
     *                  4HNPIN,2HT ,4HLINT,2HP ,12*1H       ,4HLTUR,2HN ,
     *                  4HLGEO,2HM ,6*1H        ,4HLTYP,2HE ,4HIDX ,2H  ,
     *                  4HIBLN,2H  /
      DATA      NCOM05 / 4HILX1,2H  ,4HILY1,2H  ,4HILX2,2H  ,4HILY2,2H  /
      DATA      NCOM06 / 4HIGEO,2HCP,118*1H      ,4HIXL ,2H  ,2*1H        ,
     *                  4HIYL ,2H  ,2*1H        ,4HJXL ,2H  ,2*1H        ,
     *                  4HJYL ,2H  ,2*1H        ,4HIXA ,2H  ,2*1H        ,
     *                  4HIYA ,2H  ,2*1H        ,4HLL1 ,2H  ,4HLA1 ,2H  ,
     *                  4HLA2 ,2H  ,4HLL2 ,2H  ,4HIIA ,2H  ,4HIIL ,2H  ,
     *                  4HIOA ,2H  ,4HIOL ,2H  ,4HIOPT,2H  ,4HILCH,2H  ,
     *                  4HIBA ,2H  ,2*1H        ,4HIDA ,2H  ,2*1H        ,
     *                  4HIRA ,2H  ,2*1H        ,4HIPTU,2HRN,4HLENP,2H  ,
     *                  4HLIBL,2H  ,4HLOBL,2H  ,4HLIMP,2H  ,4HNGEO,2HCP/
      DATA      NCOM07 / 4HICAN,2HSE,78*1H       /
      DATA      NCOM08 / 4HNIBA,2H  ,4HLIBA,2H  ,10*1H       ,4HNOBA,2H  ,
     *                  4HLOBA,2H  ,10*1H       ,4HNIBL,2H  ,4HNOBL,2H  ,
     *                  4HNAP ,2H  ,4HNARC,2HS ,4HLARC,2HS ,38*1H       ,
     *                  4HNLIN,2HES,4HLLIN,2HES,198*1H      ,4HNSDR,2HS ,
     *                  4HNPAT,2HHS,4HNCON,2HFS/
      DATA      NCOM09 / 4HSCAL,2HEA,4HSCAL,2HEI,4HRADI,2HUS,4HIPAT,2HH ,
     *                  4HIPLO,2HT ,4HISAM,2HE ,4HICLO,2HSE,4HIPAP,2HER,
     *                  4HIXAP,2HP ,98*1H       ,4HIYAP,2HP ,98*1H       /
      DATA      NCOM10 / 4HIAN ,2H  ,4HIA  ,2H  ,4HILN ,2H  ,4HIL  ,2H  ,
     *                  4HNLAN,2HEI,4HJAN ,2H  ,4HJA  ,2H  ,4HJLN ,2H  ,
     *                  4HJL  ,2H  ,4HNLAN,2HEJ/
      DATA      NCOM11 / 4HNPAG,2HE ,4HNLIN,2HE ,4HNTAB,2HL ,4HLINE,2HS ,
     *                  4HMODE,2HLT/
      DATA      NCOM12 / 4HXMIN,2H  ,4HYMIN,2H  ,4HXMAX,2H  ,4HYMAX,2H  ,
     *                  4HX0  ,2H  ,4HY0  ,2H  ,4HXSIZ,2HEA,4HYSIZ,2HEA,
     *                  4HXSIZ,2HEI,4HYSIZ,2HEI,4HSCAL,2HE ,4HCSIZ,2HEA,
     *                  4HCSIZ,2HEI,4HMINX,2HA ,4HMINY,2HA ,4HMAXX,2HA ,
     *                  4HMAXY,2HA ,4HMINX,2HI ,4HMINY,2HI ,4HMAXX,2HI ,
     *                  4HMAXY,2HI ,4HLTDI,2HRX,98*1H       ,4HLTDI,2HRY,
     *                  98*1H       /
      DATA      NCOM13 / 4HPI  ,2H  ,4HRADI,2HAN,4HXROU,2HND,4HFPSM,2HPH,
     *                  4HZERO,2H  ,4HD0P0,2H  /
      DATA      NCOM14 / 4HIXSD,2HRC,38*1H       ,4HIYSD,2HRC,38*1H       ,
     *                  4HNSDR,2HC ,4HLSDR,2HC ,38*1H       /
      DATA      IC     / 4HAPPR,2HO ,4HARC ,2H  ,4HCONF,2HLT,4HLANE,2H  ,
     *                  4HLINE,2H  ,4HPATH,2H  ,4HSDR ,2H  ,4HGEOP,2HRO,
     *                  4HGEOV,2HAL,4HINDE,2HX ,4HOUTP,2HUT,4HPLOT,2HTR,
     *                  4HRADI,2HAN,4HSDRC,2H  ,4HATTB,2H  ,4HENTI,2HTY/
  601 FORMAT(20A4)
  602 FORMAT(15H1 COMMON BLOCK ,A4,A2,/)
C¬603 FORMAT(2X,A4,A2,3H =  ,O20,5HB =  I10)
C;603 FORMAT(2X,A4,A2,3H =  ,Z8 ,5HZ =  I10)
C¬604 FORMAT(2X,A4,A2,3H =  ,2O20,5HB =  D25.15)
C;604 FORMAT(2X,A4,A2,3H =  ,2Z8 ,5HZ =  D25.15)
  605 FORMAT()
  606 FORMAT(11H  ATTRIBUTE,I4,7H WORD =,I4,8H SHIFT =,I3,8H MASK = ,
C¬   *          O20,1HB)
C;   *          Z8 ,1HZ)
  607 FORMAT(8H  ENTITY,I3,7H DATA =,9I5)
C
C-----SUBROUTINE ABORTR PRINTS THE ERROR MESSAGE, PRINTS THE VALUE OF
C-----THE ATTRIBUTES IN EACH ENTITY, AND PRINTS THE VALUE OF THE
C-----VARIABLES IN SELECTED COMMON BLOCKS
C
C¬    ASSIGN 101 TO IRECAD
```

```
C-     ASSIGN 102 TO JRECAD
C-     CALL   XMIT     ( IRECAD )
C------PRINT THE ERROR MESSAGE
       NWDS = (NCHS+3)/4
       PRINT 601
       PRINT 601 , (MSG(I),I=1,NWDS)
C------PRINT THE VALUE OF THE ATTRIBUTES IN EACH ENTITY
       NUM = NOATTB(1)
       PRINT 602 , IC(1,1),IC(2,1)
       PRINT 603 , (NCOM01(1,I),NCOM01(2,I),COM01(I),COM01(I),I=1,NUM)
       NUM = NOATTB(2)
       PRINT 602 , IC(1,2),IC(2,2)
       PRINT 603 , (NCOM02(1,I),NCOM02(2,I),COM02(I),COM02(I),I=1,NUM)
       NUM = NOATTB(3)
       PRINT 602 , IC(1,3),IC(2,3)
       PRINT 603 , (NCOM03(1,I),NCOM03(2,I),COM03(I),COM03(I),I=1,NUM)
       NUM = NOATTB(4)
       PRINT 602 , IC(1,4),IC(2,4)
       PRINT 603 , (NCOM04(1,I),NCOM04(2,I),COM04(I),COM04(I),I=1,NUM)
       NUM = NOATTB(5)
       PRINT 602 , IC(1,5),IC(2,5)
       PRINT 603 , (NCOM05(1,I),NCOM05(2,I),COM05(I),COM05(I),I=1,NUM)
       NUM = NOATTB(6)
       PRINT 602 , IC(1,6),IC(2,6)
       PRINT 603 , (NCOM06(1,I),NCOM06(2,I),COM06(I),COM06(I),I=1,NUM)
       NUM = NOATTB(7)
       PRINT 602 , IC(1,7),IC(2,7)
       PRINT 603 , (NCOM07(1,I),NCOM07(2,I),COM07(I),COM07(I),I=1,NUM)
C------PRINT THE VALUE OF THE VARIABLES IN SELECTED COMMON BLOCKS
       PRINT 602 , IC(1,8),IC(2,8)
       PRINT 603 , (NCOM08(1,I),NCOM08(2,I),COM08(I),COM08(I),I=1,142)
       PRINT 602 , IC(1,9),IC(2,9)
       PRINT 604 , NCOM09(1,1),NCOM09(2,1),D01,SCALEA
       PRINT 604 , NCOM09(1,2),NCOM09(2,2),D02,SCALEI
       PRINT 604 , NCOM09(1,3),NCOM09(2,3),D03,RADIUS
       PRINT 603 , (NCOM09(1,I),NCOM09(2,I),COM09(I+3),COM09(I+3),
      *            I=4,108)
       PRINT 602 , IC(1,10),IC(2,10)
       PRINT 603 , (NCOM10(1,I),NCOM10(2,I),COM10(I),COM10(I),I=1,10)
       PRINT 602 , IC(1,11),IC(2,11)
       PRINT 603 , (NCOM11(1,I),NCOM11(2,I),COM11(I),COM11(I),I=1,5)
       PRINT 602 , IC(1,12),IC(2,12)
       PRINT 604 , NCOM12(1,01),NCOM12(2,01),D04,XMIN
       PRINT 604 , NCOM12(1,02),NCOM12(2,02),D05,YMIN
       PRINT 604 , NCOM12(1,03),NCOM12(2,03),D06,XMAX
       PRINT 604 , NCOM12(1,04),NCOM12(2,04),D07,YMAX
       PRINT 604 , NCOM12(1,05),NCOM12(2,05),D08,X0
       PRINT 604 , NCOM12(1,06),NCOM12(2,06),D09,Y0
       PRINT 604 , NCOM12(1,07),NCOM12(2,07),D10,XSIZEA
       PRINT 604 , NCOM12(1,08),NCOM12(2,08),D11,YSIZEA
       PRINT 604 , NCOM12(1,09),NCOM12(2,09),D12,XSIZEI
       PRINT 604 , NCOM12(1,10),NCOM12(2,10),D13,YSIZEI
       PRINT 604 , NCOM12(1,11),NCOM12(2,11),D14,SCALE
       PRINT 604 , NCOM12(1,12),NCOM12(2,12),D15,CSIZEA
       PRINT 604 , NCOM12(1,13),NCOM12(2,13),D16,CSIZEI
       PRINT 603 , (NCOM12(1,I),NCOM12(2,I),COM12(I+13),COM12(I+13),
      *            I=14,121)
       PRINT 602 , IC(1,13),IC(2,13)
       PRINT 604 , NCOM13(1,1),NCOM13(2,1),D17,PI
       PRINT 604 , NCOM13(1,2),NCOM13(2,2),D18,RADIAN
       PRINT 604 , NCOM13(1,3),NCOM13(2,3),D19,XROUND
       PRINT 604 , NCOM13(1,4),NCOM13(2,4),D20,FPSMPH
       PRINT 604 , NCOM13(1,5),NCOM13(2,5),D21,ZERO
       PRINT 604 , NCOM13(1,6),NCOM13(2,6),D22,D0P0
       PRINT 602 , IC(1,14),IC(2,14)
       PRINT 603 , (NCOM14(1,I),NCOM14(2,I),COM14(I),COM14(I),I=1,61)
       PRINT 602 , IC(1,15),IC(2,15)
       PRINT 606 , (I,(IAT(J,I),J=1,3),I=1,200)
       PRINT 602 , IC(1,16),IC(2,16)
       PRINT 607 , (I,(IEN(J,I),J=1,9),I=1,7)
       PRINT 605
C-101 CONTINUE
C-     CALL   XMIT     ( JRECAD )
C------ECHO-PRINT THE VALUE OF THE ATTRIBUTES IN EACH ENTRY OF EACH
C------ENTITY
       CALL  ECHO
C-102 CONTINUE
C------ISSUE THE ERROR MESSAGE TO THE DAYFILE
C-     ICHS = NWDS*4
C-            ENCODE  ( ICHS,601,MSGPP )           (MSG(I),I=1,NWDS)
C-     I = (NCHS+9)/10 + 1
C-     MSGPP(I) = 0
C-     CALL   XMIT     ( 0 )
C-     CALL   REMARK   ( MSGPP )
                     IF ( IPLOT . EQ . 3 )          RETURN
C------END THE PLOT
C-     CALL   ENDPLT
C!     CALL   PLOT     ( 0.0,0.0,999 )
       RETURN
C-103 GO TO IRECAD                                                      *DEBUG*
C-104 GO TO JRECAD                                                      *DEBUG*
       END                                                              ABORTR
```

```
      SUBROUTINE ECHO                                                   COLEASE
C     TASK,ECHO
      COMMON / APPRO  / IALEFT    ,IARGHT    ,NLANES    ,LLANES( 6),    COLEASE
     *                  IAPX      ,IAPY      ,ISLIM     ,NSDR      ,    COLEASE
     *                  ISDRN ( 5),ISDRA ( 5),IAAZIM    ,NDEGST    ,    COLEASE
     *                  NDEGUT                                          COLEASE
      COMMON / ARC    / IARCX     ,IARCY     ,IARCAZ    ,IARCSW    ,    COLEASE
     *                  IARCR     ,IDUMAR                               COLEASE
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),IDUMCO                               COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),LTURN     ,LGEOM ( 4),    COLEASE
     *                  LTYPE     ,IDX       ,IBLN                      COLEASE
      COMMON / LINE   / ILX1      ,ILY1      ,ILX2      ,ILY2           COLEASE
      COMMON / NOATTB / NOATTB( 7)                                      COLEASE
      COMMON / PATH   / IGEOCP(60),IXL    ( 2),IYL    ( 2),JXL    ( 2), COLEASE
     *                  JYL    ( 2),IXA    ( 2),IYA    ( 2),LL1       , COLEASE
     *                  LA1       ,LA2       ,LL2       ,IIA       ,    COLEASE
     *                  IIL       ,IOA       ,IOL       ,IOPT      ,    COLEASE
     *                  ILCH      ,IBA    ( 2),IDA    ( 2),IRA    ( 2), COLEASE
     *                  IPTURN    ,LENP      ,LIBL      ,LOBL      ,    COLEASE
     *                  LIMP      ,NGEOCP                               COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / GEOPRO / NIBA,LIBA(6),NOBA,LOBA(6),NIBL,NOBL,NAP,NARCS,
     *                  LARCS(20),NLINES,LLINES(100),NSDRS,NPATHS,NCONFS
      COMMON / ZTEMPD / ABORTR(14),I,J,K,NUM,NUMLAN,ZTEMPD(86)
      DIMENSION         IENT1(1),IENT2(1),IENT3(1),IENT4(1),
     *                  IENT5(1),IENT6(1),IENT7(1)
      EQUIVALENCE       (IALEFT    ,IENT1(1)),(IARCX     ,IENT2(1)),
     *                  (ICONP (1),IENT3(1)),(LWID      ,IENT4(1)),
     *                  (ILX1      ,IENT5(1)),(IGEOCP(1),IENT6(1)),
     *                  (ICANSE(1),IENT7(1))
  601 FORMAT(1H1,I3,8H ARCS   =,16I6)
  602 FORMAT(18I6,/,(12X,16I6))
  603 FORMAT(1H1,I3,8H LINES  =,16I6)
  604 FORMAT(1H1,I3,8H IBAPS  =,16I6)
  605 FORMAT(1H1,I3,8H OBAPS  =,16I6)
  606 FORMAT(1H1,I3,8H LANES  =,16I6,/,(12X,16I6))
  607 FORMAT(1H1,I3,8H SDRS   =,16I6,/,(12X,16I6))
  608 FORMAT(1H1,I3,8H PATHS  =,16I6,/,(12X,16I6))
  609 FORMAT(1H1,I3,8H CONFLT=,16I6,/,(12X,16I6))
C
C-----SUBROUTINE ECHO ECHO-PRINTS THE VALUE OF THE ATTRIBUTES IN EACH
C-----ENTRY OF EACH ENTITY
C
                    IF ( NARCS  . EQ . 0 )          GO TO 1020
C-----ECHO-PRINT ENTITY ARC
      PRINT 601 , NARCS,(LARCS(I),I=1,NARCS)
      NUM = NOATTB(2)
      DO 1010  I = 1 , NARCS
      J = LARCS(I)
C     COLEASE,EXTRAC,ARC,J
      CALL  EXTRAC  (      2,J       )                                  COLEASE
      IARCSW = IARCSW - 360
      PRINT 602 , I,J,(IENT2(K),K=1,NUM)
 1010 CONTINUE
 1020 CONTINUE
                    IF ( NLINES . EQ . 0 )          GO TO 2020
C-----ECHO-PRINT ENTITY LINE
      PRINT 603 , NLINES,(LLINES(I),I=1,NLINES)
      NUM = NOATTB(5)
      DO 2010  I = 1 , NLINES
      J = LLINES(I)
C     COLEASE,EXTRAC,LINE,J
      CALL  EXTRAC  (      5,J       )                                  COLEASE
      PRINT 602 , I,J,(IENT5(K),K=1,NUM)
 2010 CONTINUE
 2020 CONTINUE
                    IF ( NAP    . EQ . 0 )          GO TO 3040
                    IF ( NIBA   . EQ . 0 )          GO TO 3020
C-----ECHO-PRINT ENTITY APPRO FOR EACH INBOUND APPROACH
      PRINT 604 , NIBA,(LIBA(I),I=1,NIBA)
      NUM = NOATTB(1)
      DO 3010  I = 1 , NIBA
      J = LIBA(I)
C     COLEASE,EXTRAC,APPRO,J
      CALL  EXTRAC  (      1,J       )                                  COLEASE
      PRINT 602 , I,J,(IENT1(K),K=1,NUM)
 3010 CONTINUE
 3020 CONTINUE
                    IF ( NOBA   . EQ . 0 )          GO TO 3040
C-----ECHO-PRINT ENTITY APPRO FOR EACH OUTBOUND APPROACH
      PRINT 605 , NOBA,(LOBA(I),I=1,NOBA)
      NUM = NOATTB(1)
      DO 3030  I = 1 , NOBA
      J = LOBA(I)
C     COLEASE,EXTRAC,APPRO,J
      CALL  EXTRAC  (      1,J       )                                  COLEASE
      PRINT 602 , I,J,(IENT1(K),K=1,NUM)
 3030 CONTINUE
 3040 CONTINUE
      NUMLAN = NIBL + NOBL
                    IF ( NUMLAN . EQ . 0 )          GO TO 4020
C-----ECHO-PRINT ENTITY LANE
      PRINT 606 , NUMLAN,(I,I=1,NUMLAN)
      NUM = NOATTB(4)
      DO 4010  I = 1 , NUMLAN
C     COLEASE,EXTRAC,LANE,I
      CALL  EXTRAC  (      4,I       )                                  COLEASE
      PRINT 602 , I,I,(IENT4(K),K=1,NUM)
 4010 CONTINUE
 4020 CONTINUE
                    IF ( NSDRS  . EQ . 0 )          GO TO 5020
C-----ECHO-PRINT ENTITY SDR
      PRINT 607 , NSDRS,(I,I=1,NSDRS)
      NUM = NOATTB(7)
      DO 5010  I = 1 , NSDRS
C     COLEASE,EXTRAC,SDR,I
      CALL  EXTRAC  (      7,I       )                                  COLEASE
      PRINT 602 , I,I,(IENT7(K),K=1,NUM)
 5010 CONTINUE
 5020 CONTINUE
                    IF ( NPATHS . EQ . 0 )          GO TO 6020
C-----ECHO-PRINT ENTITY PATH
      PRINT 608 , NPATHS,(I,I=1,NPATHS)
      NUM = NOATTB(6)
      DO 6010  I = 1 , NPATHS
C     COLEASE,EXTRAC,PATH,I
      CALL  EXTRAC  (      6,I       )                                  COLEASE
C-----UN-BIAS THE INTERSECTION PATH ATTRIBUTES
      IXA(1) = IXA(1) - 900
      IXA(2) = IXA(2) - 900
      IYA(1) = IYA(1) - 900
      IYA(2) = IYA(2) - 900
      IDA(1) = IDA(1) - 360
      IDA(2) = IDA(2) - 360
      PRINT 602 , I,I,(IENT6(K),K=1,NUM)
 6010 CONTINUE
 6020 CONTINUE
                    IF ( NCONFS . EQ . 0 )          GO TO 7020
C-----ECHO-PRINT ENTITY CONFLT
      PRINT 609 , NCONFS,(I,I=1,NCONFS)
      NUM = NOATTB(3)
      DO 7010  I = 1 , NCONFS
C     COLEASE,EXTRAC,CONFLT,I
      CALL  EXTRAC  (      3,I       )                                  COLEASE
      PRINT 602 , I,I,(IENT3(K),K=1,NUM)
 7010 CONTINUE
 7020 CONTINUE
      RETURN
      END                                                               ECHO
```

```
      SUBROUTINE SMEP    ( IR,IY,IN,IV,IE,ISNAME )                      COLEASE
      DIMENSION           ISNAME(2),IERROR(8)                           COLEASE
      DATA      IERROR / 4H FAT,4HAL E,4HRROR,4H IN ,4HCOLE,4HASE /     COLEASE
  901 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,13H OUT OF RANGE)  COLEASE
  902 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,11H ATTRIBUTE ,I3, COLEASE
     *        13H OUT OF RANGE)                                          COLEASE
  903 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,11H ATTRIBUTE ,I3, COLEASE
C¬   *        3H = ,O20,4HB = ,I10,9H OVERFLOW)                         COLEASE
C;   *        3H = ,Z8,4HZ = ,I10,9H OVERFLOW)                          COLEASE
      GO TO ( 9010,9020,9030 ) , IE                                     COLEASE
 9010 CONTINUE                                                          COLEASE
      PRINT 901 , ISNAME,IY,IN                                          COLEASE
      GO TO 9040                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      PRINT 902 , ISNAME,IY,IN,IV                                       COLEASE
      GO TO 9040                                                        COLEASE
 9030 CONTINUE                                                          COLEASE
      PRINT 903 , ISNAME,IY,IN,IV,IR,IR                                 COLEASE
 9040 CONTINUE                                                          COLEASE
      IERROR(7) = ISNAME(1)                                             COLEASE
      IERROR(8) = ISNAME(2)                                             COLEASE
      CALL  ABORTR  ( IERROR,30 )                                       COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE EXTRAC ( IY,IN )                                       COLEASE
      COMMON / APPRO  / ID(1)                                           COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION          ISNAME(2)                                      COLEASE
      DATA      ISNAME / 4HEXTR,4HACT  /                                COLEASE
C¬    DATA      NBITS  / 60 /                                           COLEASE
C                                                                       COLEASE
C-----SUBROUTINE EXTRAC EXTRACTS THE ATTRIBUTES FOR ENTRY IN OF ENTITY   COLEASE
C-----IY FROM THE STORAGE STACK AND PUTS THEM IN THE COMMON BLOCK FOR    COLEASE
C-----ENTITY IY                                                          COLEASE
C                                                                       COLEASE
C-----CHECK BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY                 COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                    IF ( IN . LT . 1 )            GO TO 9010            COLEASE
                    IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010             COLEASE
C-----SET UP THE PARAMETERS FOR EXTRAC                                   COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      IEA = IBA + IEN(IIEN+2) - 1                                       COLEASE
C-----EXTRACT EACH ATTRIBUTE FROM THE STORAGE STACK FOR ENTRY IN OF      COLEASE
C-----ENTITY IY                                                          COLEASE
      IIAT = (IBA-1)*3                                                  COLEASE
      DO 1010  I = IBA , IEA                                            COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C¬    ID(I) = LSHIFT((IS(IWD).AND.IAT(IIAT+3)),NBITS-IAT(IIAT+2))       COLEASE
C;    ID(I) = LSHIFT(IAND(IS(IWD),IAT(IIAT+3)),-IAT(IIAT+2))            COLEASE
      IIAT = IIAT + 3                                                   COLEASE
 1010 CONTINUE                                                          COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERROR AND STOP                               COLEASE
 9010 CONTINUE                                                          COLEASE
      CALL  SMEP    ( 0,IY,IN,0,1,ISNAME )                              COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE FIND    ( IR,IY,IN,IV )                                COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION        ISNAME(2)                                        COLEASE
      DATA     ISNAME / 4HFIND,4H                                       COLEASE
C¬    DATA     NBITS  / 60 /                                            COLEASE
C                                                                       COLEASE
C-----SUBROUTINE FIND FINDS THE VALUE OF ATTRIBUTE IV OF ENTRY IN OF    COLEASE
C-----ENTITY IY AND PUTS IT INTO LOCAL INTEGER IR                       COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                     IF ( IN . LT . 1 )            GO TO 9010           COLEASE
                     IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010            COLEASE
C-----CHECK THE BOUNDARIES FOR ATTRIBUTE NUMBER IV                      COLEASE
                     IF ( IV . LT . 1 )            GO TO 9020           COLEASE
                     IF ( IV . GT . IEN(IIEN+2) ) GO TO 9020            COLEASE
C-----SET UP THE PARAMETERS FOR FIND                                    COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      I = IBA + IV - 1                                                  COLEASE
      IIAT = (I-1)*3                                                    COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C-----FIND LOCAL INTEGER IR FROM ATTRIBUTE IV OF ENTRY IN OF ENTITY IV  COLEASE
C¬    IR = LSHIFT((IS(IWD).AND.IAT(IIAT+3)),NBITS-IAT(IIAT+2))          COLEASE
C;    IR = LSHIFT(IAND(IS(IWD),IAT(IIAT+3)),-IAT(IIAT+2))               COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IE = 1                                                            COLEASE
      GO TO 9030                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IE = 2                                                            COLEASE
 9030 CONTINUE                                                          COLEASE
      CALL  SMEP    ( 0,IY,IN,IV,IE,ISNAME )                            COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE REPACK ( IY,IN )                                       COLEASE
      COMMON / APPRO  / ID(1)                                           COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION        ISNAME(2)                                        COLEASE
      DATA     ISNAME / 4HREPA,4HCK                                     COLEASE
C                                                                       COLEASE
C-----SUBROUTINE REPACK REPACKS THE VALUES OF THE ATTRIBUTES FROM THE   COLEASE
C-----COMMON BLOCK FOR ENTITY IY INTO ENTRY IN OF ENTITY IY IN THE      COLEASE
C-----STORAGE STACK                                                     COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                     IF ( IN . LT . 1 )            GO TO 9010           COLEASE
                     IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010            COLEASE
C-----SET UP THE PARAMETERS FOR REPACK                                  COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      ILW = IFW + NWE - 1                                               COLEASE
C-----ZERO OUT ENTRY IN OF ENTITY IY IN THE STORAGE STACK               COLEASE
      DO 1010  I = IFW , ILW                                            COLEASE
      IS(I) = 0                                                         COLEASE
 1010 CONTINUE                                                          COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      IEA = IBA + IEN(IIEN+2) - 1                                       COLEASE
      IIAT = (IBA-1)*3                                                  COLEASE
C-----REPACK EACH ATTRIBUTE FOR ENTRY IN OF ENTITY IY                   COLEASE
      DO 1020  I = IBA , IEA                                            COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
      IT = LSHIFT(ID(I),IAT(IIAT+2))                                    COLEASE
C¬    IX = IT . AND . (.NOT.IAT(IIAT+3))                                COLEASE
C;    IX = IAND(IT,INOT(IAT(IIAT+3)))                                   COLEASE
C-----IF ATTRIBUTE I IS OUT OF RANGE THEN ERROR                         COLEASE
                     IF ( IX . NE . 0 )            GO TO 9020           COLEASE
C¬    IS(IWD) = IT . OR . IS(IWD)                                       COLEASE
C;    IS(IWD) = IOR(IT,IS(IWD))                                         COLEASE
      IIAT = IIAT + 3                                                   COLEASE
 1020 CONTINUE                                                          COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IR = 0                                                            COLEASE
      IV = 0                                                            COLEASE
      IE = 1                                                            COLEASE
      GO TO 9030                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IR = ID(I)                                                        COLEASE
      IV = I - IBA + 1                                                  COLEASE
      IE = 3                                                            COLEASE
 9030 CONTINUE                                                          COLEASE
      CALL  SMEP    ( IR,IY,IN,IV,IE,ISNAME )                           COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE STORE  ( IR,IY,IN,IV )                                 COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION         ISNAME(2)                                       COLEASE
      DATA      ISNAME / 4HSTOR,4HE    /                                COLEASE
C                                                                       COLEASE
C-----SUBROUTINE STORE STORES THE VALUE OF LOCAL INTEGER IR INTO        COLEASE
C-----ATTRIBUTE IV OF ENTRY IN OF ENTITY IY IN THE STORAGE STACK        COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                    IF ( IN . LT . 1 )           GO TO 9010             COLEASE
                    IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010             COLEASE
C-----CHECK THE BOUNDARIES FOR ATTRIBUTE NUMBER IV                      COLEASE
                    IF ( IV . LT . 1 )           GO TO 9020             COLEASE
                    IF ( IV . GT . IEN(IIEN+2) ) GO TO 9020             COLEASE
C-----SET UP THE PARAMETERS FOR STORE                                   COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      I = IBA + IV - 1                                                  COLEASE
      IIAT = (I-1)*3                                                    COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C-----STORE LOCAL INTEGER IR INTO ATTRIBUTE IV OF ENTRY IN OF ENTITY IY COLEASE
      IT = LSHIFT(IR,IAT(IIAT+2))                                       COLEASE
C¬    IX = IT . AND . (.NOT.IAT(IIAT+3))                                COLEASE
C;    IX = IAND(IT,INOT(IAT(IIAT+3)))                                   COLEASE
C-----IF LOCAL INTEGER IR IS OUT OF RANGE THEN ERROR                    COLEASE
                    IF ( IX . NE . 0 )           GO TO 9030             COLEASE
C¬    IS(IWD) = IT.OR.(IS(IWD).AND.(.NOT.IAT(IIAT+3)))                  COLEASE
C;    IS(IWD) = IOR(IT,IAND(IS(IWD),INOT(IAT(IIAT+3))))                 COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IE = 1                                                            COLEASE
      GO TO 9040                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IE = 2                                                            COLEASE
      GO TO 9040                                                        COLEASE
 9030 CONTINUE                                                          COLEASE
      IE = 3                                                            COLEASE
 9040 CONTINUE                                                          COLEASE
      CALL  SMEP    ( IR,IY,IN,IV,IE,ISNAME )                           COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

PROGRAMMERS DOCUMENTATION

GEOMETRY PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKAGE

LATEST UPDATE: 27 AUG 77

THIS DOCUMENTATION IS DIVIDED INTO THE FOLLOWING SECTIONS:

1. GEOMETRY PROCESSOR LIMITATIONS

```
MAXIMUM NUMBER OF INBOUND APPROACHES ---------------- 6
MAXIMUM NUMBER OF OUTBOUND APPROACHES --------------- 6
RANGE OF APPROACH NUMBERS --------------------------- 1→12
MAXIMUM SPEED LIMIT FOR APPROACHES ------------------ 118 FT/SEC (80 MPH)
MAXIMUM NUMBER OF LANES PER APPROACH ---------------- 6
MAXIMUM SIGHT DISTANCE RESTRICTIONS PER APPROACH ---- 5

MAXIMUM NUMBER OF INBOUND LANES --------------------- 25
MAXIMUM NUMBER OF OUTBOUND LANES -------------------- 25
MAXIMUM LENGTH OF LANES ----------------------------- 1000 FEET
MAXIMUM WIDTH OF LANES ------------------------------ 15 FEET
MAXIMUM NUMBER OF INTERSECTION PATHS PER LANE ------- 7

MAXIMUM NUMBER OF INTERSECTION PATHS ---------------- 125
MAXIMUM LENGTH OF PATHS ----------------------------- 250 FEET
MAXIMUM SPEED LIMIT FOR PATHS ----------------------- 118 FT/SEC (80 MPH)
MAXIMUM NUMBER OF CONFLICTS PER PATH ---------------- 60

MAXIMUM NUMBER OF ARCS ------------------------------ 20
RANGE OF ARC NUMBERS -------------------------------- 1→20
MAXIMUM RADIUS OF ARC ------------------------------- 127 FEET

MAXIMUM NUMBER OF LINES ----------------------------- 100
RANGE OF LINE NUMBERS ------------------------------- 1→100

MAXIMUM NUMBER OF SIGHT DISTANCE RESTRICTIONS ------- 20
RANGE OF SIGHT DISTANCE RESTRICTIONS ---------------- 1→20

MAXIMUM NUMBER OF CONFLICTS ------------------------- 1000

RANGE OF X OR Y COORDINATES ------------------------- 0→2250 FT
```

2. EXPLANATION OF INPUT ERRORS

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READIO:

STOP 801 = NUMBER OF INBOUND APPROACHES = <NIBA> IS LE 0 OR GT 6
(NUMBER OF INBOUND APPROACHES OUT OF RANGE 1⇒6)
STOP 802 = INBOUND APPROACH <I> = <LIBA(I)> IS LE 0 OR GT 12
(INBOUND APPROACH NUMBER OUT OF RANGE 1⇒12)
STOP 803 = INBOUND APPROACH <I> = <LIBA(I)> IS EQUAL TO INBOUND
APPROACH <K> = <LIBA(K)>
(APPROACH NUMBER IS ALREADY ON LIST OF INBOUND APPROACHES)
STOP 804 = NUMBER OF OUTBOUND APPROACHES = <NOBA> IS LE 0 OR GT 6
(NUMBER OF OUTBOUND APPROACHES OUT OF RANGE 1⇒6)
STOP 805 = OUTBOUND APPROACH <I> = <LOBA(I)> IS LE 0 OR GT 12
(OUTBOUND APPROACH NUMBER OUT OF RANGE 1⇒12)
STOP 806 = OUTBOUND APPROACH <I> = <LOBA(I)> IS EQUAL TO OUTBOUND
APPROACH <K> = <LOBA(K)>
(APPROACH NUMBER IS ALREADY ON LIST OF OUTBOUND APPROACHES)
STOP 807 = INBOUND APPROACH <I> = <LIBA(I)> IS EQUAL TO OUTBOUND
APPROACH <J> = <LOBA(J)>
(APPROACH NUMBER IS ON BOTH INBOUND AND OUTBOUND LISTS)
STOP 808 = NUMBER OF APPROACHES = <NAP> IS LE 0 OR GT 12
(NUMBER OF APPROACHES IS OUT OF RANGE 1⇒12)
STOP 809 = NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND APPROACHES =
<NTEST> IS NE NUMBER OF APPROACHES <NAP>
(NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND APPROACHES
DOES NOT EQUAL THE NUMBER OF APPROACHES)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READAP:

STOP 810 = APPROACH NUMBER <J> IS LE 0 OR GT 12
(APPROACH NUMBER OUT OF RANGE 1⇒12)
STOP 811 = APPROACH NUMBER <J> IS USED MORE THAN ONCE
(APPROACH DATA ALREADY ENTERED FOR THIS APPROACH)
STOP 812 = APPROACH NUMBER <J> AZIMUTH = <IAAZIM> IS LT 0 OR GT 360
(APPROACH AZIMUTH OUT OF RANGE 0⇒360)
STOP 813 = APPROACH NUMBER <J> X COORDINATE = <IAPX> IS LT 0 OR GT 2250
(APPROACH X COORDINATE OUT OF RANGE 0⇒2250)
STOP 814 = APPROACH NUMBER <J> Y COORDINATE = <IAPY> IS LT 0 OR GT 2250
(APPROACH Y COORDINATE OUT OF RANGE 0⇒2250)
STOP 815 = APPROACH NUMBER <J> SPEED LIMIT = <ISLIM> IS LT 10 OR GT 80
(APPROACH SPEED LIMIT IS OUT OF RANGE 10⇒80)
STOP 816 = APPROACH NUMBER <J> NUMBER OF LANES = <NLANES> IS LE 0 OR GT 6
(APPROACH NUMBER OF LANES IS OUT OF RANGE 1⇒6)
STOP 817 = APPROACH NUMBER <J> NUMBER OF DEGREES FOR STRAIGHT = <NDEGST>
IS LT 0 OR GT 45
(NUMBER OF DEGREES FOR STRAIGHT IS OUT OF RANGE 0⇒45)
STOP 818 = APPROACH NUMBER <J> NUMBER OF DEGREES FOR U=TURN = <NDEGUT>
IS LT 0 OR GT 45
(NUMBER OF DEGREES FOR U=TURN IS OUT OF RANGE 0⇒45)
STOP 819 = APPROACH NUMBER <J> IS NOT ON INBOUND OR OUTBOUND LISTS
(APPROACH DATA SPECIFIED FOR AN APPROACH THAT IS NOT ON THE
INBOUND OR OUTBOUND LISTS)
STOP 820 = APPROACH NUMBER <J> IS ON INBOUND LIST YET HAS OUTBOUND DATA
SPECIFIED
(APPROACH IS ON INBOUND LIST YET DOES NOT HAVE A HEADWAY
DISTRIBUTION SPECIFIED)
STOP 821 = NUMBER OF INBOUND LANES = <NIBL> IS GT 25
(NUMBER OF INBOUND LANES OUT OF RANGE 1⇒25)
STOP 822 = APPROACH NUMBER <J> IS ON OUTBOUND LIST YET HAS INBOUND DATA
SPECIFIED
(APPROACH IS ON OUTBOUND LIST YET HAS A HEADWAY
DISTRIBUTION SPECIFIED)
STOP 823 = NUMBER OF OUTBOUND LANES = <NOBL> IS GT 25
(NUMBER OF OUTBOUND LANES OUT OF RANGE 1⇒25)
STOP 824 = APPROACH NUMBER <J> IS OUTBOUND YET HAS DATA FOR PERCENT OF
EACH VEHICLE CLASS MAKING THE TRAFFIC STREAM
(APPROACH IS ON OUTBOUND LIST YET HAS PERCENT OF EACH
VEHICLE CLASS MAKING THE TRAFFIC STREAM)
STOP 825 = LANE NUMBER <ILN> LANE WIDTH = <LWID> IS LT 8 OR GT 15
(LANE WIDTH IS OUT OF RANGE 8⇒15)
STOP 826 = LANE NUMBER <ILN> LANE GEOMETRY <IZ> = <LGEOM(IZ)> IS LT 0 OR
GT 1000
(LANE GEOMETRY IS OUT OF RANGE 0⇒1000)
STOP 827 = LANE NUMBER <ILN> LANE GEOMETRY ORDER INCORRECT
(LANE GEOMETRY ORDER INCORRECT = SHOULD PASS ONE OF THESE TESTS:
(1).EQ.(3).AND.(2).EQ.(4).AND.(2).GT.(1) (REGULAR)
(1).EQ.(2).AND.(3).GT.(2).AND.(4).GT.(3) (ONLY AT END)
(3).EQ.(4).AND.(2).GT.(1).AND.(3).GT.(2) (ONLY AT START)
(2).GT.(1).AND.(3).GT.(2).AND.(4).GT.(3) (BLOCKED IN MIDDLE)
STOP 828 = LANE NUMBER <ILN> LANE GEOMETRY 1 = <LGEOM(1)> IS NE
LANE GEOMETRY 1 OF LAST LANE = <LGEOM(1)>
(ALL LGEOM(1)S FOR AN INBOUND APPROACH MUST BE THE SAME)
STOP 829 = LANE NUMBER <ILN> TURN CODE = <IUT> IS NOT ( ) OR (U)
(LANE TURN CODE IS NOT ( ) OR (U))
STOP 830 = LANE NUMBER <ILN> TURN CODE = <ILT> IS NOT ( ) OR (L)
(LANE TURN CODE IS NOT ( ) OR (L))
STOP 831 = LANE NUMBER <ILN> TURN CODE = <IST> IS NOT ( ) OR (S)
(LANE TURN CODE IS NOT ( ) OR (S))
STOP 832 = LANE NUMBER <ILN> TURN CODE = <IRT> IS NOT ( ) OR (R)
(LANE TURN CODE IS NOT ( ) OR (R))
STOP 833 = LANE NUMBER <ILN> NO TURN CODE SPECIFIED
(NO LANE TURN CODE SPECIFIED FOR A LANE THAT MUST HAVE IT)
STOP 834 = INFORMATION FOR APPROACH <IA> NOT SPECIFIED
(APPROACH WAS ON INBOUND OR OUTBOUND LIST BUT NO APPROACH DATA
WAS SPECIFIED)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READAI:

STOP 835 = NUMBER OF ARCS = <NARCS> IS LT 0 OR GT 20
(NUMBER OF ARCS OUT OF RANGE 0⇒20)
STOP 836 = ARC NUMBER <I> = <J> IS LE 0 OR GT 20
(ARC NUMBER IS OUT OF RANGE 1⇒20)
STOP 837 = ARC NUMBER <J> IS USED MORE THAN ONCE
(ARC DATA ALREADY ENTERED FOR THIS ARC)
STOP 838 = ARC NUMBER <J> X COORDINATE = <IARCX> IS LT 0 OR GT 2250
(ARC X COORDINATE OUT OF RANGE 0⇒2250)
STOP 839 = ARC NUMBER <J> Y COORDINATE = <IARCY> IS LT 0 OR GT 2250
(ARC Y COORDINATE OUT OF RANGE 0⇒2250)
STOP 840 = ARC NUMBER <J> AZIMUTH = <IARCAZ> IS LT 0 OR GT 360
(ARC AZIMUTH OUT OF RANGE 0⇒360)
STOP 841 = ARC NUMBER <J> NUMBER OF DEGREES = <IARCSW> IS LT =360 OR GT +360
(ARC NUMBER OF DEGREES IS OUT OF RANGE =360⇒+360)
STOP 842 = ARC NUMBER <J> RADIUS = <IARCR> IS LE 0 OR GT 127
(ARC RADIUS IS OUT OF RANGE 1⇒127)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READLI:

STOP 843 = NUMBER OF LINES = <NLINES> IS LT 0 OR GT 100
(NUMBER OF LINES OUT OF RANGE 0⇒100)
STOP 844 = LINE NUMBER <I> = <J> IS LE 0 OR GT 100
(LINE NUMBER IS OUT OF RANGE 1⇒100)
STOP 845 = LINE NUMBER <J> IS USED MORE THAN ONCE
(LINE DATA ALREADY ENTERED FOR THIS LINE)
STOP 846 = LINE NUMBER <J> BEGINING X COORDINATE = <ILX1> IS LT 0 OR GT 2250
(LINE BEGINNING X COORDINATE IS OUT OF RANGE 0⇒2250)
STOP 847 = LINE NUMBER <J> BEGINING Y COORDINATE = <ILY1> IS LT 0 OR GT 2250
(LINE BEGINNING Y COORDINATE IS OUT OF RANGE 0⇒2250)
STOP 848 = LINE NUMBER <J> ENDING X COORDINATE = <ILX2> IS LT 0 OR GT 2250
(LINE ENDING X COORDINATE IS OUT OF RANGE 0⇒2250)
STOP 849 = LINE NUMBER <J> ENDING Y COORDINATE = <ILY2> IS LT 0 OR GT 2250
(LINE ENDING Y COORDINATE IS OUT OF RANGE 0⇒2250)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READSI:

STOP 850 = NUMBER OF SIGHT DISTANCE RESTRICTIONS = <NSDRC> IS LT 0 OR GT 20
(NUMBER OF SIGHT DISTANCE RESTRICTIONS IS OUT OF RANGE 0⇒20)

STOP 851 = SIGHT DISTANCE RESTRICTION NUMBER <I> = <J> LE 0 OR GT 20
(SIGHT DISTANCE RESTRICTION NUMBER IS OUT OF RANGE 1-20)
STOP 852 = SIGHT DISTANCE NUMBER <J> IS USED MORE THAN ONCE
(SIGHT DISTANCE RESTRICTION DATA ALREADY ENTERED FOR THIS SIGHT DISTANCE RESTRICTION)
STOP 853 = SIGHT DISTANCE RESTRICTION <J> X COORDINATE = <IXSDRC> IS LT 0 OR GT 2250
(SIGHT DISTANCE RESTRICTION X COORDINATE IS OUT OF RANGE 0-2250)
STOP 854 = SIGHT DISTANCE RESTRICTION <J> Y COORDINATE = <IYSDRC> IS LT 0 OR GT 2250
(SIGHT DISTANCE RESTRICTION Y COORDINATE IS OUT OF RANGE 0-2250)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READOP:

STOP 855 = PATH OPTION = (<JPATH>) IS NE (   )OR(PRIMARY)OR(OPTION1)
(PATH OPTION IS NOT (   ) OR (PRIMARY) OR (OPTION1))
STOP 856 = PLOT OPTION = (<JPLOT>) IS NE (   )OR(PLOT   )OR(PLOTI   )OR (NOPLOT   )
(PLOT OPTION IS NOT (   ) OR (PLOT   ) OR (PLOTI   ) OR (NOPLOT   ))
STOP 857 = PATH PLOT OPTION = (<JSAME>) IS NE (   )OR(SAME   )OR (SEPARATE)
(PATH PLOT OPTION IS NOT (   ) OR (SAME   ) OR (SEPARATE))
STOP 858 = CLOSE DISTANCE = <ICLOSE> IS LT 6 OR GT 20
(CLOSE DISTANCE IS OUT OF RANGE 6-20)
STOP 859 = PLOT PAPER WIDTH = <IPAPER> IS NE 12 OR 30
(PLOT PAPER WIDTH IS NOT 12 OR 30 INCHES)

3. EXPLANATION OF EXECUTION ERRORS

STOP 901 IN FNDXYP = NO SCALE FACTOR ON SCALEF LIST WILL ALLOW THE APPROACH TO BE PLOTTED
(CAN NOT GET HERE HALT)
STOP 902 IN FNDXYP = NO SCALE FACTOR ON SCALEF LIST WILL ALLOW THE INTERSECTION TO BE PLOTTED
(CAN NOT GET HERE HALT)
STOP 903 IN FNDSDR = APPROACHES DO NOT INTERSECT
(CAN NOT GET HERE HALT)
STOP 904 IN FNDSDR = NUMBER OF SIGHT DISTANCE RESTRICTIONS FOR APPROACH IS GT 5
(CAN NOT GET HERE HALT)
STOP 905 IN FNDSDR = NUMBER OF ENTRIES FOR SIGHT RESTRICTION ENTITY IS GT 30
(CAN NOT GET HERE HALT)
STOP 906 IN FNDPTH = NUMBER OF PATHS IS LE 0
(NO PATHS COULD BE CALCULATED FOR THE INTERSECTION)
STOP 907 IN CALPTH = PATH TURN CODE DOES NOT MATCH ANY TURN CODE FOR INBOUND APPROACH
(PATH TURN CODE ALREADY MATCHES LANE TURN CODE FOR CONNECTING INBOUND AND OUTBOUND LANES, YET WHEN TRYING TO FIND LNI, THE PATH TURN CODE DOES NOT MATCH ANY OF THE LANE TURN CODES FOR THE INBOUND APPROACH - CAN NOT GET HERE HALT)
STOP 908 IN CALPTH = PATH TURN CODE DOES NOT MATCH ANY TURN CODE FOR OUTBOUND APPROACH
(PATH TURN CODE ALREADY MATCHES LANE TURN CODE FOR CONNECTING INBOUND AND OUTBOUND LANES, YET WHEN TRYING TO FIND LNJ, THE PATH TURN CODE DOES NOT MATCH ANY OF THE LANE TURN CODES FOR THE OUTBOUND APPROACH - CAN NOT GET HERE HALT)
STOP 909 IN ADDPTH = NUMBER OF PATHS IS GT 125
(CHECK EVERYTHING - IF OK THEN CONTACT AGENCY SUPPLING PROGRAM AND REQUEST MODIFICATION OF PROGRAM TO ACCOMMODATE MORE THAN 125 INTERSECTION PATHS
STOP 910 IN ADDPTH = NUMBER OF PATHS FROM LANE IS GT 7
(CHECK EVERYTHING - IF OK THEN CONTACT AGENCY SUPPLING PROGRAM AND REQUEST MODIFICATION OF PROGRAM TO ACCOMMODATE MORE THAN 7 INTERSECTION PATHS PER LANE
STOP 911 IN CHKPTH = NO PATH INTO INTERSECTION
(NO INTERSECTION PATHS CALCULATED FOR A LANE THAT IS AVAILABLE AT THE INTERSECTION - CHECK TURN CODES)
STOP 912 IN CHKPTH = PATH WAS NOT GENERATED FOR EACH TURN CODE FOR LANE
(LANE TURN CODE SPECIFIED A PATH THAT WAS NOT CALCULATED - CHECK TURN CODES)
STOP 913 IN FNDCON = TOTAL NUMBER OF CONFLICTS IS LE 0
(NO CONFLICTS FOUND BETWEEN ANY INTERSECTION PATHS - CHECK ICLOSE VALUE AND PATHS)
STOP 914 IN ADDCON = TOTAL NUMBER OF CONFLICTS IS GT 1000
(CHECK EVERYTHING - IF OK THEN CONTACT AGENCY SUPPLING PROGRAM AND REQUEST MODIFICATION OF PROGRAM TO ACCOMMODATE MORE THAN 1000 CONFLICTS)
STOP 915 IN ADDCON = NUMBER OF CONFLICTS FOR PATH IS GT 60
(CHECK EVERYTHING - IF OK THEN CONTACT AGENCY SUPPLING PROGRAM AND REQUEST MODIFICATION OF PROGRAM TO ACCOMMODATE MORE THAN 60 CONFLICTS PER PATH)
STOP 916 IN ICHKA = SWEEP ANGLE FOR ARC PORTION OF PATH EQ 0
(VERY UNLIKELY HALT - CHECK ARC PORTIONS OF PATHS)
STOP 917 IN CATOAC = CIRCLES ARE IDENTICAL
(ARC PORTION OF PATHS HAVE EXACT SAME CENTER COORDINATES AND RADIUS - VERY UNLIKELY HALT - CHECK ARC PORTIONS OF PATHS)
STOP 918 IN NDXCON = CONFLICT WAS NOT FOUND IN IGEOCP LIST FOR PATH
(WHEN CROSS INDEXING, THE PATH INDEXED BY THE ICONP ARRAY IN ENTITY CONFLT DOES NOT HAVE THIS CONFLICT IN ITS IGEOCP ARRAY IN ENTITY PATH - CAN NOT GET HERE HALT)

4. DEFINITION OF ATTRIBUTES IN EACH ENTITY AND THE ROUTINES IN WHICH EACH ENTITY IS USED

APPRO ENTITY FOR APPROACHES (12 ENTRIES)
GEOPRO READAP FNDXYP FNDSDR WRITAP DRWAPR DRWBOX DRWINT DRWUTA CHKPTH ABORTR ECHO EXTRAC REPACK

IAAZIM	AZIMUTH OF APPROACH [0,360]
IALEFT	ENTRY NUMBER OF APPROACH TO THE LEFT [1,12]
IAPX	X COORDINATE OF BEGINING OF APPROACH AT THE MEDIAN [0,2250]
IAPY	Y COORDINATE OF BEGINING OF APPROACH AT THE MEDIAN [0,2250]
IARGHT	ENTRY NUMBER OF APPROACH TO THE RIGHT [1,12]
ISDRA(5)	LIST OF ENTRY NUMBERS OF OTHER APPROACH FOR SIGHT DISTANCE RESTRICTION [1,12]
ISDRN(5)	LIST OF ENTRY NUMBERS FOR SDR ENTITY OF SIGHT DISTANCE RESTRICTION [1,30]
ISLIM	THE LEGAL SPEED LIMIT (FT/SEC) [0,110]
LLANES(6)	LIST OF ENTRY NUMBERS FOR LANE ENTITY OF LANES IN THE APPROACH, SUBSCRIPTED BY LANE NUMBER COUNTED FROM MEDIAN TO CURB [1,50]
NDEGST	NUMBER OF DEGREES LEFT OR RIGHT OF STRAIGHT FOR PATH TO BE CONSIDERED STRAIGHT (DEFAULT IS 20) [0,45]
NDEGUT	NUMBER OF DEGREES LESS THAN 180 FOR PATH TO BE CONSIDERED AS A U-TURN (DEFAULT IS 10) [0,45]
NLANES	NUMBER OF LANES [1,6]
NSDR	NUMBER OF SIGHT DISTANCE RESTRICTIONS [0,5]

ARC ENTITY FOR ARC DEFINITIONS (20 ENTRIES)
GEOPRO READAI WRITAL DRWAPR DRWINT ABORTR ECHO

IARCAZ	AZIMUTH OF BEGINING OF ARC [0,360]
IARCR	RADIUS OF ARC (FEET) [1,127]
IARCSW	NUMBER OF DEGREES OF ARC (BIASED) [0,720]
IARCX	X COORDINATE OF CENTER OF ARC [0,2250]
IARCY	Y COORDINATE OF CENTER OF ARC [0,2250]
IDUMAR	DUMMY VARIABLE FOR ARC ENTITY TO MAKE NUMBER OF ATTRIBUTES EVEN

CONFLT ENTITY FOR INTERSECTION CONFLICTS (1000 ENTRIES)
GEOPRO ADDCON SRTCON NDXCON WRITCO ABORTR ECHO

ICONA(2)	ENTRY NUMBER FOR APPRO ENTITY OF LINKING INBOUND APPROACH OF PATHS INVOLVED IN CONFLICT [1,12]
ICONAN	CONFLICT ANGLE MEASURED FROM FIRST PATH CLOCKWISE [0,360]
ICOND(2)	DISTANCE DOWN PATH FROM START OF PATH TO CONFLICT [0,250]
ICONI(2)	INDEX NUMBER FOR IGEOCP/ICPSET ARRAYS IN PATH ENTITY FOR ENTRY ICONP() [1,60]
ICONP(2)	ENTRY NUMBER FOR PATH ENTITY OF PATHS INVOLVED IN CONFLICT [1,125]
IDUMCO	DUMMY VARIABLE FOR CONFLT ENTITY TO MAKE NUMBER OF ATTRIBUTES EVEN

LANE ENTITY FOR THE LANES IN THE APPROACHES (50 ENTRIES)
GEOPRO READAP FNDSDR DRWAPR DRWINT CHKPTH WRITLA ABORTR ECHO

IBLN	INBOUND LANE NUMBER [1,25]
IDX	DISTANCE FROM MEDIAN TO CENTER OF LANE (FEET) [0,90]
ISNA	ENTRY NUMBER FOR APPRO ENTITY OF APPROACH CONTAINING LANE [1,12]
LGEOM(4)	BEGINNING AND END POINTS OF LANE, WITH THE FOLLOWING INDEXES: [0,1000] (1)=FIRST BEGINNING POINT (2)=FIRST END POINT (3)=SECOND BEGINNING POINT (4)=SECOND END POINT
LINTP(7)	LIST OF ENTRY NUMBERS FOR PATH ENTITY OF PATHS INTO THE INTERSECTION [1,125]
LTURN	TURN CODE OF THE LANE: [0,15] 0=OUTBOUND 1= RIGHT 2= STRAIGHT 3= STRAIGHT RIGHT 4= LEFT 5= LEFT RIGHT 6= LEFT STRAIGHT 7= LEFT STRAIGHT RIGHT 8=U-TURN 9=U-TURN RIGHT 10=U-TURN STRAIGHT 11=U-TURN STRAIGHT RIGHT 12=U-TURN LEFT 13=U-TURN LEFT RIGHT 14=U-TURN LEFT STRAIGHT 15=U-TURN LEFT STRAIGHT RIGHT
LTYPE	TYPE OF LANE: [1,2] 1=INBOUND 2=OUTBOUND
LWID	WIDTH OF LANE (FEET) [8,15]
NLL	ENTRY NUMBER OF LANE TO LEFT [1,50]
NLR	ENTRY NUMBER OF LANE TO RIGHT [1,50]
NPINT	NUMBER OF PATHS INTO THE INTERSECTION [0,7]

LINE ENTITY FOR LINE DEFINITIONS (100 ENTRIES)
GEOPRO READLI WRITAL DRWAPR DRWINT ABORTR ECHO

ILX1	X COORDINATE OF BEGINNING OF LINE [0,2250]
ILX2	X COORDINATE OF END OF LINE [0,2250]
ILY1	Y COORDINATE OF BEGINNING OF LINE [0,2250]
ILY2	Y COORDINATE OF END OF LINE [0,2250]

PATH ENTITY FOR INTERSECTION PATHS (125 ENTRIES)
GEOPRO FNDPTH ADDPTH DRWPTH FNDCON CLTOLC CLTOAC ADDLA CATOLC ADDAL CATOAC ADDAA SRTCON WRITPA NDXCON ABORTR ECHO

IBA(2)	BEGINING AZIMUTH OF ARCS [0,360]
IGEOCP(60)	LIST OF ENTRY NUMBERS FOR CONFLT ENTITY FOR THE GEOMETRIC CONFLICT POINTS [1,1000]
IDA(2)	NUMBER OF DEGREES OF ARCS (BIASED) [0,720]
IIA	ENTRY NUMBER FOR APPRO ENTITY OF CONNECTING INBOUND APPROACH [1,12]
IIL	INDEX NUMBER OF CONNECTING INBOUND LANE [1,6]
ILCH	LANE CHANGE WITHIN THE INTERSECTION FLAG 0=NO 1=YES
IOPT	PATH OPTION [0,1] 0=PRIMARY 1=OPTION1
IOA	ENTRY NUMBER FOR APPRO ENTITY OF CONNECTING OUTBOUND APPROACH [1,12]
IOL	INDEX NUMBER OF CONNECTING OUTBOUND LANE [1,6]
IPTURN	PATH TURN CODE [1,8] 1= RIGHT 2= STRAIGHT 4= LEFT 8=U-TURN
IRA(2)	RADIUS OF ARCS [0,900]
IXA(2)	X COORDINATE OF CENTER OF ARCS (BIASED) [0,4050]
IXL(2)	X COORDINATE OF BEGINING OF LINES [0,2250]
IYA(2)	Y COORDINATE OF CENTER OF ARCS (BIASED) [0,4050]
IYL(2)	Y COORDINATE OF BEGINING OF LINES [0,2250]
JXL(2)	X COORDINATE OF END OF LINES [0,2250]
JYL(2)	Y COORDINATE OF END OF LINES [0,2250]

```
LA1         LENGTH OF FIRST ARC (FEET) [0→250]
LA2         LENGTH OF SECOND ARC (FEET) [0→250]
LENP        LENGTH OF PATH (FEET) (KL+LL+ML+NL) [0→250]
LIBL        ENTRY NUMBER FOR LANE ENTITY OF LINKING INBOUND LANE [1→50]
LIMP        THE MINIMUM OF THE PHYSICAL SPEED LIMIT OF THE PATH AND
            THE LEGAL SPEED LIMIT OF THE LINKING APPROACHES
            (FT/SEC) [0→118]
LL1         LENGTH OF FIRST LINE (FEET) [0→250]
LL2         LENGTH OF SECOND LINE (FEET) [0→250]
LOBL        ENTRY NUMBER FOR LANE ENTITY OF LINKING OUTBOUND LANE
            [1→50]
NGEOCP      NUMBER OF GEOMETRIC CONFLICT POINTS [0→60]


SDR         ENTITY FOR AVAILABLE SIGHT DISTANCES (30 ENTRIES)
                GEOPRO FNDSDR WRITLA ABORTR ECHO

 ICANSE(40) POSITION ALONG ANOTHER APPROACH THAT IS JUST VISIBLE
            FOR AN APPROACH (INDEXED BY (POSITION DOWN APPROACH)/
            25 + 1) [0→1000]
```

```
5. DEFINITION OF VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH
   EACH COMMON BLOCK IS USED


COMMON BLOCKS <APPRO>, <ARC>, <CONFLT>, <LANE>, <LINE>, <PATH>, AND <SDR>
ARE ENTITIES AND ARE EXPLAINED IN SECTION 4


COMMON / ATTB /   COLEASE GENERATED DATA TO DESCRIBE THE ATTRIBUTES IN EACH
                  ENTITY
                      GEOPRO BLKDAT ABORTR EXTRAC  REPACK STORE  FIND

   IAT(2,200)     DESCRIBES THE LOCATION AND SIZE OF THE ATTRIBUTES
                      (1,I)=THE STARTING BIT POSITION FOR EACH ATTRIBUTE
                      (2,I)=NUMBER OF BITS FOR EACH ATTRIBUTE (AFTER THE
                            DO 1010 LOOP IN GEOPRO IT IS THE MASK FOR EACH
                            ATTRIBUTE POSITIONED PROPERLY)


COMMON / DATA /   VARIABLES USED TO CALCULATE THE PATHS THROUGH THE
                  INTERSECTION
                      FNDPTH CALPTH STRLFT STRSTR STRRGH UTURNL UTURNR LTLTGE
                      LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT ZEROP1
                      ZEROP2 ZEROP3 ZEROP4 MAXVEL ADDPTH DRWPTH

   ADX            ABS(XI-XO) AFTER BEING ROTATED BY THE NEGATIVE VALUE
                  OF THE AZIMUTH
   ADY            ABS(YI-YO) AFTER BEING ROTATED BY THE NEGATIVE VALUE
                  OF THE AZIMUTH
   IFLAG          FLAG TO INDICATE IF PATH IS LEGAL [0→1]
                      0=PATH LEGAL
                      1=PATH NOT LEGAL
   JANGLE         NUMBER OF DEGREES THE VEHICLE TURNS THROUGH NEGOTIATING
                  THE PATH
   JAZIM          AZIMUTH OF INBOUND APPROACH [0→360]
   JB2            BEGINING AZIMUTH OF FIRST ARC OF PATH [0→360]
   JB3            BEGINING AZIMUTH OF SECOND ARC OF PATH [0→360]
   JD2            NUMBER OF DEGREES OF FIRST ARC OF PATH [-360→+360]
   JD3            NUMBER OF DEGREES OF SECOND ARC OF PATH [-360→+360]
   JLCH           LANE CHANGE WITHIN THE INTERSECTION FLAG
                      0=NO
                      1=YES
   JOPT           PATH OPTION [0→1]
                      0=PRIMARY
                      1=OPTION1
   JSPEED         MAXIMUM PHYSICAL SPEED POSSIBLE FOR PATH (BASED ON
                  RADIUS OF PATH) (FT/SEC) [0→118]
   KAZIM          AZIMUTH OF OUTBOUND APPROACH [0→360]
   KTURN          PATH TURN CODE [1→8]
                      1=                    RIGHT
                      2=            STRAIGHT
                      4=       LEFT
                      8=U-TURN
   L1             LENGTH OF FIRST LINE OF PATH [0→250]
   L2             LENGTH OF FIRST ARC OF PATH [0→250]
   L3             LENGTH OF SECOND ARC OF PATH [0→250]
   L4             LENGTH OF SECOND LINE OF PATH [0→250]
   RA2            RADIUS OF FIRST ARC OF PATH [0→900]
   RA3            RADIUS OF SECOND ARC OF PATH [0→900]
   RC             CRITICAL ARC RADIUS  (WHEN ARC IS TANGENT AT BOTH ENDS)
                  [0→1000]
   XC2            X COORDINATE OF THE CENTER OF FIRST ARC OF PATH [-900→+3150]
   XC3            X COORDINATE OF THE CENTER OF SECOND ARC OF PATH [-900→+3150]
   XI             X COORDINATE OF THE END OF INBOUND LANE [0→2250]
   XO             X COORDINATE OF THE BEGINING OF OUTBOUND LANE [0→2250]
   X11            X COORDINATE OF THE BEGINING OF FIRST LINE OF PATH [0→2250]
   X12            X COORDINATE OF THE END OF FIRST LINE OF PATH [0→2250]
   X41            X COORDINATE OF THE BEGINING OF SECOND LINE OF PATH  [0→2250]
   X42            X COORDINATE OF THE END OF SECOND LINE OF PATH [0→2250]
   YC             CRITICAL ADY  (WHEN RADIUS IS RC)
   YC2            Y COORDINATE OF THE CENTER OF FIRST ARC OF PATH [-900→+3150]
```

YC3	Y COORDINATE OF THE CENTER OF SECOND ARC OF PATH [-900,+3150]
YI	Y COORDINATE OF THE END OF INBOUND LANE [0,2250]
YO	Y COORDINATE OF THE BEGINING OF OUTBOUND LANE [0,2250]
Y11	Y COORDINATE OF THE BEGINING OF FIRST LINE OF PATH [0,2250]
Y12	Y COORDINATE OF THE END OF FIRST LINE OF PATH [0,2250]
Y41	Y COORDINATE OF THE BEGINING OF SECOND LINE OF PATH [0,2250]
Y42	Y COORDINATE OF THE END OF SECOND LINE OF PATH [0,2250]

COMMON / ENTITY / COLEASE GENERATED DATA TO DESCRIBE THE ENTITIES
GEOPRO BLKDAT ABORTR EXTRAC REPACK STORE FIND

IEN(9,7) DATA TO DESCRIBE THE ENTITIES
(1,I)=NUMBER OF ENTRIES FOR ENTITY I
(2,I)=NUMBER OF ATTRIBUTES FOR ENTITY I
(3,I)=NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR AN ENTRY FOR ENTITY I
(4,I)=LOCATION OF THE FIRST ENTRY IN THE STORAGE STACK FOR ENTITY I
(5,I)=NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR THE LOGICAL INDEPENDENT ATTRIBUTES FOR ENTITY I
(6,I)=LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK (RELATIVE TO THE FIRST) FOR THE LOGICAL INDEPENDENT ATTRIBUTES FOR ENTITY I
(7,I)=NUMBER OF FUNCTION MASKS FOR THE LOGICAL ATTRIBUTES FOR ENTITY I
(8,I)=LOCATION OF THE FIRST FUNCTION MASK IN THE IFU ARRAY IN /FUN/ FOR ENTITY I
(9,I)=LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/ FOR ENTITY I

COMMON / GEOCP / VARIABLES USED TO CHECK PATH TO PATH CONFLICTS
FNDCON BAND CLTOLC CLTOAC ADDLA CATOLC ADDAL CATOAC ADDAA

MAL(2)	ARC LENGTH SUBSCRIPTED BY (IFS) [0,250]
MBA(2,5)	BEGINING AZIMUTH OF ARC SUBSCRIPTED BY (IFS,IBAND) [0,360]
MDA(2,5)	NUMBER OF DEGREES OF ARC SUBSCRIPTED BY (IFS,IBAND) [-360,+360]
MIA	ENTRY NUMBER FOR APPRO ENTITY OF CONNECTING INBOUND APPROACH [1,12]
MLL(2)	LINE LENGTH SUBSCRIPTED BY (IFS) [0,250]
MPTH	ENTRY NUMBER FOR PATH ENTITY OF PATH ALONG WHICH CONFLICTS ARE BEING CHECKED (PATH WITH BANDS) [1,124]
MRA(2,5)	RADIUS OF ARCS SUBSCRIPTED BY (IFS,IBAND) [0,900]
MXA(2,5)	X COORDINATE OF CENTER OF ARCS SUBSCRIPTED BY (IFS,IBAND) [-900,+3150]
MXL(2,5)	X COORDINATE OF BEGINING OF LINES SUBSCRIPTED BY (IFS,IBAND) [0,2250]
MYA(2,5)	Y COORDINATE OF CENTER OF ARCS SUBSCRIPTED BY (IFS,IBAND) [-900,+3150]
MYL(2,5)	Y COORDINATE OF BEGINING OF LINES SUBSCRIPTED BY (IFS,IBAND) [0,2250]
NPTH	ENTRY NUMBER FOR PATH ENTITY OF PATH TO WHICH CONFLICTS ARE BEING CHECKED [2,125]
NXL(2,5)	X COORDINATE OF END OF LINES SUBSCRIPTED BY (IFS,IBAND) [0,2250]
NYL(2,5)	Y COORDINATE OF END OF LINES SUBSCRIPTED BY (IFS,IBAND) [0,2250]
XINT1	X COORDINATE OF FIRST POINT OF INTERSECTION [0,2250]
XINT2	X COORDINATE OF SECOND POINT OF INTERSECTION [0,2250]
YINT1	Y COORDINATE OF FIRST POINT OF INTERSECTION [0,2250]
YINT2	Y COORDINATE OF SECOND POINT OF INTERSECTION [0,2250]

COMMON / GEOPRO / DATA ABOUT INTERSECTION
BLKDAT READIN READIO READAP READAI READLI WRITAL FNDXYP
FNDSDR WRITAP DRWAPR DRWINT FNDPTH ADDPTH CHKPTH WRITLA
FNDCON ADDCON SRTCON WRITPA NDXCON WRITCO ABORTR ECHO

LIBA(6)	LIST OF ENTRY NUMBERS FOR APPRO ENTITY OF INBOUND APPROACHES [1,12]
LLINES(16)	LIST OF ENTRY NUMBER FOR LINE ENTITY OF LINES [1,16]
LOBA(6)	LIST OF ENTRY NUMBERS FOR APPRO ENTITY OF OUTBOUND APPROACHES [1,12]
NAP	TOTAL NUMBER OF APPROACHES IN THE INTERSECTION [1,12]
NARCS	TOTAL NUMBER OF ARCS [0,16]
NCONFS	TOTAL NUMBER OF POINTS OF CONFLICT [0,1000]
NIBA	NUMBER OF INBOUND APPROACHES [1,6]
NIBL	NUMBER OF INBOUND LANES [1,25]
NLINES	TOTAL NUMBER OF LINES [0,16]
NOBA	NUMBER OF OUTBOUND APPROACHES [1,6]
NOBL	NUMBER OF OUTBOUND LANES [1,25]
NPATHS	TOTAL NUMBER OF PATHS IN THE INTERSECTION [1,125]
NSDRS	TOTAL NUMBER OF SIGHT DISTANCE RESTRICTIONS [0,30]

COMMON / GEOVAL / USER SUPPLIED DATA FOR OPTIONS AND COORDINATES OF CENTER OF LANES AT THE INTERSECTION
BLKDAT EXEC READOP FNDXYP INIPLT DRWAPR DRWLIN DRWARC
DRWINT FNDPTH CALPTH CHKPTH FNDCON ADDCON ADDLA ADDAL
ADDAA ABORTR

ICLOSE	MINIMUM DISTANCE BETWEEN TO PATHS FOR CONFLICT TO BE DETECTED (DEFAULT IS 10) [6,20]
IPAPER	
IPATH	TYPE OF PATH SELECTED (DEFAULT IS PRIMARY) [1,2] 1=PRIMARY 2=OPTION1
IPLOT	TYPE OF PLOT SELECTED (DEFAULT IS PLOT) [1,3] 1=PLOT SELECTED USING 30 INCH PAPER AND BALL POINT PEN 2=PLOT SELECTED USING 30 INCH PAPER AND INK PEN 3=NO PLOT SELECTED
ISAME	TYPE OF FRAME FOR PLOTTING SELECTED (DEFAULT IS SEPARATE) [1,2] 1=APPROACH PATHS PLOTTED ON SAME FRAME 2=APPROACH PATHS PLOTTED ON SEPARATE FRAMES
IXAPP(50)	X COORDINATE OF CENTER OF THE LANES AT THE INTERSECTION (NEGATIVE VALUE MEANS THAT THE LANE IS NOT AVAILABLE AT THE INTERSECTION) [0,2250]
IYAPP(50)	Y COORDINATE OF CENTER OF THE LANES AT THE INTERSECTION (NEGATIVE VALUE MEANS THAT THE LANE IS NOT AVAILABLE AT THE INTERSECTION) [0,2250]
RADIUS	MAXIMUM RADIUS FOR PATH BEFORE A STRAIGHT LINE WILL BE USED FOR PATH (DEFAULT IS 500) [100,900]
SCALEA	APPROACH SCALE FACTOR (INPUT OR CALCULATED) (FT/IN)
SCALEI	INTERSECTION SCALE FACTOR (INPUT OR CALCULATED) (FT/IN)

COMMON / INDEX / INDEX NUMBERS FOR CURRENT ENTITIES BEING PROCESSED
READIO READAP APPLAR FNDXYP FNDSDR WRITAP DRWAPR FNDPTH
CALPTH ADDPTH CHKPTH ABORTR

IA	ENTRY NUMBER FOR APPRO ENTITY OF APPROACH BEING PROCESSED [1,12]
IAN	INDEX NUMBER FOR LIBA/LOBA ARRAYS OF /GEOPRO/ OF APPROACH BEING PROCESSED [1,6]
IL	ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED [1,50]
ILN	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF LANE BEING PROCESSED [1,6]
JA	ENTRY NUMBER FOR APPRO ENTITY OF OTHER APPROACH BEING PROCESSED [1,12]
JAN	INDEX NUMBER FOR LIBA/LOBA ARRAYS OF /GEOPRO/ OF OTHER APPROACH BEING PROCESSED [1,6]
JL	ENTRY NUMBER FOR LANE ENTITY OF OTHER LANE BEING PROCESSED [1,50]
JLN	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF OTHER LANE BEING PROCESSED [1,6]
NLANEI	NUMBER OF LANES IN IA APPROACH [1,25]
NLANEJ	NUMBER OF LANES IN JA APPROACH [1,25]

```
COMMON / LOGICV /
                          BLKDAT

   LFALSE          LOGICAL FALSE FOR COLEASE LOGICAL ATTRIBUTES
   LTRUE           LOGICAL TRUE FOR COLEASE LOGICAL ATTRIBUTES


COMMON / NOATTB / COLEASE GENERATED NUMBER OF ATTRIBUTES FOR EACH ENTITY
                          BLKDAT READAP READAI READLI FNDPTH ABORTR ECHO

   NOATTB(7)       NUMBER OF ATTRIBUTES FOR EACH ENTITY FOR COLEASE


COMMON / OUTPUT / REGULATES PRINTING OF OUTPUT
                          BKLDAT EXEC   HEADER READIO READAP READAI READLI READSI
                          READOP WRITAL FNDXYP FNDSDR WRITAP WRITLA WRITPA WRITCO
                          ABORTR

   LINES           TOTAL NUMBER OF LINES TO BE PRINTER ON A PAGE
   MODELT          TAPE NUMBER FOR WRITING DATA FOR MODELT
   NLINE           NUMBER OF LINES ALREADY PRINTED ON THIS PAGE
   NPAGE           SERIAL PAGE NUMBER IN OUTPUT
   NTABL           SERIAL TABLE NUMBER IN OUTPUT


COMMON / PLOTTR / VARIABLES USED IN PLOTTING
                          BLKDAT FNDXYP DRWAPR DRWLIN DRWARC DRWINT DRWUTA DRWARR
                          ABORTR

   CSIZEA          CHARACTER SIZE FOR APPROACH PLOT
   CSIZEI          CHARACTER SIZE FOR INTERSECTION PLOT
   LTDIRX(50)      X COORDINATE OF LOCATION OF CENTER OF DIRECTION ARROW
   LTDIRY(50)      Y COORDINATE OF LOCATION OF CENTER OF DIRECTION ARROW
   MAXXA           MAXIMUM X COORDINATE FOR APPROACH PLOT
   MAXXI           MAXIMUM X COORDINATE FOR INTERSECTION PLOT
   MAXYA           MAXIMUM Y COORDINATE FOR APPROACH PLOT
   MAXYI           MAXIMUM Y COORDINATE FOR INTERSECTION PLOT
   MINXA           MINIMUM X COORDINATE FOR APPROACH PLOT
   MINXI           MINIMUM X COORDINATE FOR INTERSECTION PLOT
   MINYA           MINIMUM Y COORDINATE FOR APPROACH PLOT
   MINYI           MINIMUM Y COORDINATE FOR INTERSECTION PLOT
   SCALE           CURRENT PLOT SCALE FACTOR (FT/IN)
   XMAX            CURRENT MAXIMUM X COORDINATE
   XMIN            CURRENT MINIMUM X COORDINATE
   XSIZEA          SIZE ON X AXIS FOR APPROACH PLOT (INCHES)
   XSIZEI          SIZE ON X AXIS FOR INTERSECTION PLOT (INCHES)
   X0              X AXIS DISPLACEMENT FOR RE-ORIGIN
   YMAX            CURRENT MAXIMUM Y COORDINATE
   YMIN            CURRENT MINIMUM Y COORDINATE
   YSIZEA          SIZE ON Y AXIS FOR APPROACH PLOT (INCHES)
   YSIZEI          SIZE ON Y AXIS FOR INTERSECTION PLOT (INCHES)
   Y0              Y AXIS DISPLACEMENT FOR RE-ORIGIN


COMMON / RADIAN / CONSTANTS USED IN CONVERSION
                          BLKDAT READIN READAP FNDXYP FNDSDR LTOL   LDOWN  DRWARC
                          DRWUTA DRWARR CALPTH STRLFT STRSTR STRRGH UTURNL UTURNR
                          LTLTGE LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT
                          ZEROP1 ZEROP2 ZEROP3 ZEROP4 MAXVEL ADDPTH BAND   CLTOLC
                          ADDCON CLTOAC ADDLA  ICHKL  ICHKA  CATOLC ADDAL  CATOAC
                          XVAL   ADDAA  XROTX  XROTI  AZIM36 ATAN36 ABORTR

   D0P0            DOUBLE PRECISION 0.0 (ZERO)
   FPSMPH          VALUE TO CONVERT FROM MPH TO FPS (88.0/60.0)
   PI              VALUE FOR THE NUMBER OF RADIAN FOR 180 DEGREES (3.14159)
   RADIAN          VALUE FOR THE NUMBER OF RADIANS PER DEGREE (0.0174532)
   XROUND          VALUE TO ROUND TO NEAREST INTEGER (0.500001)
   ZERO            VALUE OF A VERY SMALL NUMBER (0.000001)


COMMON / SDRC /   SIGHT DISTANCE RESTRICTION COORDINATES
                          READSI FNDSDR DRWAPR DRWINT ABORTR

   IXSDRC(20)      X COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
                   [0*2250]
   IYSDRC(20)      Y COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
                   [0*2250]
   LSDRC(20)       LIST OF ENTRY NUMBERS OF SIGHT DISTANCE RESTRICTION
                   COORDINATES [0*20]
   NSDRC           TOTAL NUMBER OF SIGHT DISTANCE RESTRICTION COORDINATES
                   [0*20]


COMMON / STACK /  COLEASE GENERATED STORAGE STACK
                          GEOPRO EXTRAC REPACK STORE  FIND

   IS(3391)        COLEASE STORAGE STACK FOR CDC
   IS(6845)        COLEASE STORAGE STACK FOR IBM


COMMON / TITLE /  TITLE FOR GEOMETRY PROCESSOR
                          READIN HEADER WRITAL DRWAPR DRWINT

   ITITLE(20)      80 CHARACTER TITLE FOR GEOMETRY PROCESSOR


COMMON / ZTEMPD / TEMPORARY VARIABLES USED THROUGHOUT PROGRAM
                          READIO READAP APPLAR READAI READLI READSI READOP WRITAL
                          FNDXYP FNDSDR LTOL   LDOWN  WRITAP DRWAPR DRWBOX DRWLIN
                          DRWARC DRWINT DRWUTA DRWARR CALPTH STRLFT STRRGH LTLTGE
                          LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT MAXVEL
                          ADDPTH DRWPTH CHKPTH WRITLA FNDCON BAND   CLTOLC ADDCON
                          CLTOAC ADDLA  ICHKA  CATOLC ADDAL  CATOAC XVAL   ADDAA
                          SRTCON WRITPA NDXCON WRITCO ABORTR ECHO

   ZTEMPD(105)     TEMPORARY VARIABLES USED THROUGHOUT PROGRAM
```

6. DEFINITION OF LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL EACH ROUTINE THEM, AND THE ROUTINES THEY CALL

VARIABLES THAT ARE LOCAL WITHIN SUBROUTINES ARE LISTED BELOW, EXCEPT FOR MOST DO-LOOP INDICES

SUBROUTINE ABORTR PRINTS THE ERROR MESSAGE, PRINTS THE VALUE OF THE ATTRIBUTES IN EACH ENTITY, AND PRINTS THE VALUE OF THE VARIABLES IN SELECTED COMMON BLOCKS
(CALLED FROM EXEC FNDXYP FNDSDR FNDPTH CALPTH ADDPTH CHKPTH FNDCON ADDCON ICHKA CATOAC NDXCON SMEP)
(CALLS ECHO)

COM01	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK APPRO
COM02	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK ARC
COM03	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK CONFLT
COM04	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK LANE
COM05	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK LINE
COM06	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK PATH
COM07	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK SDR
COM08	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK GEOPRO
COM09	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK GEOVAL
COM10	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK INDEX
COM11	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK OUTPUT
COM12	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK PLOTTR
COM13	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK RADIAN
COM14	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK SDRC
D01	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO SCALEA
D02	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO SCALEI
D03	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO RADIUS
D04	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO XMIN
D05	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO YMIN
D06	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO XMAX
D07	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO YMAX
D08	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO X0
D09	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO Y0
D10	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO XSIZEA
D11	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO YSIZEA
D12	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO XSIZEI
D13	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO YSIZEI
D14	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO SCALE
D15	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO CSIZEA
D16	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO CSIZEI
D17	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO PI
D18	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO RADIAN
D19	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO XROUND
D20	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO FPSMPH
D21	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO ZERO
D22	ARRAY DIMENSIONED TO 2 WHICH IS EQUIVALENCED TO D0P0
IC(2,16)	COMMON BLOCK NAMES
ICHS	NUMBER OF CHARACTERS TO ENCODE FOR REMARK (CDC ONLY)
IRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
JRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
MSG(NWDS)	ERROR MESSAGE PRINTED
MSGPP(9)	ERROR MESSAGE FOR REMARK (CDC ONLY)
NCHS	NUMBER OF CHARACTERS IN ERROR MESSAGE
NCOM01(2,026)	VARIABLE NAMES FOR PRINTING ENTITY APPRO
NCOM02(2,006)	VARIABLE NAMES FOR PRINTING ENTITY ARC
NCOM03(2,010)	VARIABLE NAMES FOR PRINTING ENTITY CONFLT
NCOM04(2,020)	VARIABLE NAMES FOR PRINTING ENTITY LANE
NCOM05(2,004)	VARIABLE NAMES FOR PRINTING ENTITY LINE
NCOM06(2,094)	VARIABLE NAMES FOR PRINTING ENTITY PATH
NCOM07(2,040)	VARIABLE NAMES FOR PRINTING COMMON BLOCK SDR
NCOM08(2,054)	VARIABLE NAMES FOR PRINTING COMMON BLOCK GEOPRO
NCOM09(2,107)	VARIABLE NAMES FOR PRINTING COMMON BLOCK GEOVAL
NCOM10(2,010)	VARIABLE NAMES FOR PRINTING COMMON BLOCK INDEX
NCOM11(2,005)	VARIABLE NAMES FOR PRINTING COMMON BLOCK OUTPUT
NCOM12(2,121)	VARIABLE NAMES FOR PRINTING COMMON BLOCK PLOTTR
NCOM13(2,006)	VARIABLE NAMES FOR PRINTING COMMON BLOCK RADIAN
NCOM14(2,061)	VARIABLE NAMES FOR PRINTING COMMON BLOCK SDRC
NUM	NUMBER OF ATTRIBUTES FOR ENTITY BEING PRINTED
NWDS	NUMBER OF WORDS FOR ERROR MESSAGE MSG

SUBROUTINE ADDAA ADDS INTERSECTION CONFLICTS BETWEEN THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM CATOAC)
(CALLS AZIM36 ICHKA ADDCON)

AZIM11	AZIMUTH OF RADIUS OF FIRST ARC AT FIRST POINT OF INTERSECTION
AZIM12	AZIMUTH OF RADIUS OF SECOND ARC AT FIRST POINT OF INTERSECTION
AZIM21	AZIMUTH OF RADIUS OF FIRST ARC AT SECOND POINT OF INTERSECTION
AZIM22	AZIMUTH OF RADIUS OF SECOND ARC AT SECOND POINT OF INTERSECTION
AZ11	AZIMUTH OF TANGENT OF FIRST ARC AT FIRST POINT OF INTERSECTION
AZ12	AZIMUTH OF TANGENT OF SECOND ARC AT FIRST POINT OF INTERSECTION
AZ21	AZIMUTH OF TANGENT OF FIRST ARC AT SECOND POINT OF INTERSECTION
AZ22	AZIMUTH OF TANGENT OF SECOND ARC AT SECOND POINT OF INTERSECTION
CONVAR(44)	CONFLICT VARIABLES FOR CONCURRENT USAGE
DA11	ANGLE BETWEEN FIRST POINT OF INTERSECTION AND THE START OF THE FIRST ARC
DA12	ANGLE BETWEEN FIRST POINT OF INTERSECTION AND THE START OF THE SECOND ARC
DA21	ANGLE BETWEEN SECOND POINT OF INTERSECTION AND THE START OF THE FIRST ARC
DA22	ANGLE BETWEEN SECOND POINT OF INTERSECTION AND THE START OF THE SECOND ARC
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
IL1	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED TO THE POINT OF INTERSECTION CONFLICT
IL2	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED AGAINST TO THE POINT OF INTERSECTION CONFLICT
ITEST1	TEST WHETHER FIRST POINT OF CONFLICT LIES THE ARC PORTION OF THE PATH BEING CHECKED 0=YES 1=NO
ITEST2	TEST WHETHER FIRST POINT OF CONFLICT LIES THE LINE PORTION OF THE PATH BEING CHECKED AGAINST 0=YES 1=NO
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH

JTEST1	TEST WHETHER SECOND POINT OF CONFLICT LIES THE ARC PORTION OF THE PATH BEING CHECKED 0=YES 1=NO
JTEST2	TEST WHETHER SECOND POINT OF CONFLICT LIES THE LINE PORTION OF THE PATH BEING CHECKED AGAINST 0=YES 1=NO
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
NUM	NUMBER OF POINTS OF CONFLICT DETECTED FOR PATHS CONSIDERED
NUMPTS	NUMBER OF POINTS OF CONFLICT ADDED FOR PATHS CONSIDERED
X	DISTANCE BETWEEN POINTS OF INTERSECTION
XBEAR1	X BEARING OF RADIUS OF FIRST ARC AT POINT OF INTERSECTION
XBEAR2	X BEARING OF RADIUS OF SECOND ARC AT POINT OF INTERSECTION
YBEAR1	Y BEARING OF RADIUS OF FIRST ARC AT POINT OF INTERSECTION
YBEAR2	Y BEARING OF RADIUS OF SECOND ARC AT POINT OF INTERSECTION

SUBROUTINE ADDAL ADDS INTERSECTION CONFLICTS BETWEEN THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM CATOLC)
(CALLS ICHKL AZIM36 ICHKA ADDCON)

AZIM1	AZIMUTH OF RADIUS OF ARC AT FIRST POINT OF INTERSECTION
AZIM2	AZIMUTH OF RADIUS OF ARC AT SECOND POINT OF INTERSECTION
AZ11	AZIMUTH OF TANGENT OF ARC AT FIRST POINT OF INTERSECTION
AZ12	AZIMUTH OF LINE AT FIRST POINT OF INTERSECTION
AZ21	AZIMUTH OF TANGENT OF ARC AT SECOND POINT OF INTERSECTION
AZ22	AZIMUTH OF LINE AT SECOND POINT OF INTERSECTION
BEARX	X BEARING OF LINE
BEARY	Y BEARING OF LINE
CONVAR(44)	CONFLICT VARIABLES FOR CONCURRENT USAGE
DA1	ANGLE BETWEEN THE FIRST POINT OF CONFLICT AND THE START OF THE ARC
DA2	ANGLE BETWEEN THE SECOND POINT OF CONFLICT AND THE START OF THE ARC
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
IL1	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED TO THE POINT OF INTERSECTION CONFLICT
IL2	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED AGAINST TO THE POINT OF INTERSECTION CONFLICT
ITEST1	TEST WHETHER FIRST POINT OF CONFLICT LIES THE ARC PORTION OF THE PATH BEING CHECKED 0=YES 1=NO
ITEST2	TEST WHETHER FIRST POINT OF CONFLICT LIES THE LINE PORTION OF THE PATH BEING CHECKED AGAINST 0=YES 1=NO
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
JTEST1	TEST WHETHER SECOND POINT OF CONFLICT LIES ON THE ARC PORTIO OF THE PATH BEING CHECKED 0=YES 1=NO
JTEST2	TEST WHETHER SECOND POINT OF CONFLICT LIES ON THE LINE PORTION OF THE PATH BEING CHECKED AGAINST 0=YES 1=NO
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
NUM	NUMBER OF POINTS OF CONFLICT DETECTED FOR PATHS CONSIDERED
NUMPTS	NUMBER OF POINTS OF CONFLICT ADDED FOR PATHS CONSIDERED
X	DISTANCE BETWEEN POINTS OF INTERSECTION
XBEAR	X BEARING OF POINT OF INTERSECTION FROM CENTER OF ARC
YBEAR	Y BEARING OF POINT OF INTERSECTION FROM CENTER OF ARC

SUBROUTINE ADDCON ADDS INTERSECTION CONFLICTS BETWEEN TWO INTERSECTION PATHS
(CALLED FROM CLTOLC ADDLA ADDAL ADDAA)
(CALLS ABORTR EXTRAC FIND REPACK STORE)

AI	AZIMUTH OF PATH CONSIDERED AT POINT OF INTERSECTION
AJ	AZIMUTH OF CONFLICTING PATH AT POINT OF INTERSECTION
CONVAR(86)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IANGLE	ANGLE BETWEEN CONFLICTING PATHS AT POINT OF INTERSECTION
INA	APPROACH NUMBER OF PATH BEING CHECKED
INL	DISTANCE ALONG PATH BEING CHECKED TO POINT OF INTERSECTION CONFLICT
INP	PATH NUMBER OF PATH BEING CHECKED
JNA	APPROACH NUMBER OF PATH BEING CHECKED AGAINST
JNL	DISTANCE ALONG PATH BEING CHECKED AGAINST TO POINT OF INTERSECTION CONFLICT
JNP	PATH NUMBER OF PATH BEING CHECKED AGAINST
KP	INDEX INTO /CONFLT/ FOR INP PATH
LP	INDEX INTO /CONFLT/ FOR JNP PATH
MGEOCP	LOCAL NUMBER OF GEOMETRIC CONFLICT POINTS
MSG914(12)	ERROR MESSAGE
MSG915(12)	ERROR MESSAGE
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED

SUBROUTINE ADDLA ADDS INTERSECTION CONFLICTS BETWEEN THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM CLTOAC)
(CALLS ICHKL AZIM36 ICHKA ADDCON)

AZIM1	AZIMUTH OF RADIUS OF ARC AT FIRST POINT OF INTERSECTION
AZIM2	AZIMUTH OF RADIUS OF ARC AT SECOND POINT OF INTERSECTION
AZ11	AZIMUTH OF LINE AT FIRST POINT OF INTERSECTION
AZ12	AZIMUTH OF TANGENT OF ARC AT FIRST POINT OF INTERSECTION
AZ21	AZIMUTH OF LINE AT SECOND POINT OF INTERSECTION
AZ22	AZIMUTH OF TANGENT OF ARC AT SECOND POINT OF INTERSECTION
BEARX	X BEARING OF LINE
BEARY	Y BEARING OF LINE
CONVAR(44)	CONFLICT VARIABLES FOR CONCURRENT USAGE
DA1	ANGLE BETWEEN THE FIRST POINT OF CONFLICT AND THE START OF THE ARC
DA2	ANGLE BETWEEN THE SECOND POINT OF CONFLICT AND THE START OF THE ARC
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
IL1	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED TO THE POINT OF INTERSECTION CONFLICT
IL2	DISTANCE FROM THE START OF THE INTERSECTION PATH BEING CHECKED AGAINST TO THE POINT OF INTERSECTION CONFLICT
ITEST1	TEST WHETHER FIRST POINT OF CONFLICT LIES THE ARC PORTION OF THE PATH BEING CHECKED 0=YES 1=NO
ITEST2	TEST WHETHER FIRST POINT OF CONFLICT LIES THE LINE PORTION OF THE PATH BEING CHECKED AGAINST 0=YES 1=NO
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
JTEST1	TEST WHETHER SECOND POINT OF CONFLICT LIES THE ARC PORTION OF THE PATH BEING CHECKED

	0=YES
	1=NO
JTEST2	TEST WHETHER SECOND POINT OF CONFLICT LIES THE LINE PORTION OF THE PATH BEING CHECKED AGAINST
	0=YES
	1=NO
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
NUM	NUMBER OF POINTS OF CONFLICT DETECTED FOR PATHS CONSIDERED
NUMPTS	NUMBER OF POINTS OF CONFLICT ADDED FOR PATHS CONSIDERED
X	DISTANCE BETWEEN THE POINTS OF CONFLICT
XBEAR	X BEARING OF THE POINT OF CONFLICT FROM THE CENTER OF ARC
YBEAR	Y BEARING OF THE POINT OF CONFLICT FROM THE CENTER OF ARC

SUBROUTINE ADDPTH ADDS INTERSECTION PATHS FOR A LANE
(CALLED FROM FNDPTH)
(CALLS XROTI AJAZIM ABORTR REPACK STORE FIND)

JSLIM	THE LEGAL SPEED LIMIT OF THE LINKING INBOUND APPROACH
KSLIM	THE LEGAL SPEED LIMIT OF THE LINKING OUTBOUND APPROACH
MSG909(9)	ERROR MESSAGE
MSG910(11)	ERROR MESSAGE
NPINT	

SUBROUTINE AJAZIM ADDS JAZIML TO JB2OR3 AND MAKES IT FALL IN THE RANGE FROM 0 TO 360 DEGREES AND SETS IDAL TO JD2OR3 WHEN THE LENGTH OF THE ARC (L2OR3) IS GT 0
(CALLED FROM ADDPTH)

IBAL	CALCULATED BEGINING AZIMUTH OF ARC OF PATH
IDAL	CALCULATED SWEEP ANGLE OF ARC OF PATH
JAZIML	AZIMUTH OF INBOUND APPROACH
JB2OR3	BEGINING AZIMUTH OF FIRST ARC OF PATH
JD2OR3	SWEEP ANGLE OF ARC OF PATH
L2OR3	LENGTH OF ARC SEGMENT OF PATH

SUBROUTINE APPLAR FINDS THE APPROACH TO THE LEFT AND THE APPROACH TO THE RIGHT FOR EACH APPROACH ON THE LBA LIST
(CALLED FROM READIN)
(CALLS STORE FIND)

IALEFT	ENTRY NUMBER OF APPROACH TO THE LEFT [1-12]
IARGHT	ENTRY NUMBER OF APPROACH TO THE RIGHT [1-12]
IMAXAZ	MAXIMUM AZIMUTH OF AN APPROACH FROM APPROACH BEING PROCESSED
IMINAZ	MINIMUM AZIMUTH OF AN APPROACH FROM APPROACH BEING PROCESSED
JAAZIM	AZIMUTH OF APPROACH UNDER CONSIDERATION
KAAZIM	AZIMUTH OF APPROACH REQUIRED
LAAZIM	DIFFERENCE BETWEEN JAZIM AND KAZIM
LBA(NBA)	LIST OF ENTRY NUMBERS FOR APPRO ENTITY OF APPROACHES TO BE PROCESSED (LIBA OR LOBA)
NBA	NUMBER OF APPROACHES (NIBA OR NOBA)

FUNCTION ATAN36 FINDS THE ARC TANGENT OF A COORDINATE AND RETURNS THE ANGLE FROM 0 TO 360 DEGREES (EAST ZERO AND COUNTER-CLOCKWISE POSITIVE)
(CALLED FROM AZIM36)

ATAN36	ARC TANGENT OF A COORDINATE FROM 0 TO 360 DEGREES (EAST ZERO AND COUNTER CLOCKWISE POSITIVE)
X	X COORDINATE
Y	Y COORDINATE

FUNCTION AZIM36 FINDS THE ARC TANGENT OF A COORDINATE AND RETURNS THE AZIMUTH FROM 0 TO 360 DEGREES (NORTH ZERO AND CLOCKWISE POSITIVE)
(CALLED FROM BAND CLTOLC ADDLA ADDAL ADDAA)
(CALLS ATAN36)

AZIM36	AZIMUTH OF A COORDINATE FROM 0 TO 360 DEGREES (NORTH ZERO AND CLOCKWISE POSITIVE)
X	X COORDINATE
Y	Y COORDINATE

SUBROUTINE BAND BUILDS A BAND IDIST DISTANCE FROM THE MAIN INTERSECTION PATH EITHER LEFT OR RIGHT OF THE MAIN INTERSECTION PATH DEPENDING UPON ILR
(CALLED FROM FNDCON)
(CALLS AZIM36 XROTAI)

BEARX	X BEARING OF LINE OF PATH CONSIDERED
BEARY	Y BEARING OF LINE OF PATH CONSIDERED
CONVAR(12)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IAZ1	AZIMUTH OF LINE PERPENDICULAR TO FIRST LINE OF PATHS CONSIDERED
IAZ2	AZIMUTH OF LINE PERPENDICULAR TO SECOND LINE OF PATHS CONSIDERED
IB	INDEX NUMBER FOR BAND
	1=MAIN PATH
	2=BAND ONE FOOT LEFT OF MAIN PATH
	3=BAND ONE FOOT RIGHT OF MAIN PATH
	4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH
	5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IDIST	DISTANCE FROM MAIN PATH FOR BAND
ILR	WHETHER BAND IS TO THE LEFT OR RIGHT OF MAIN PATH
	-1=LEFT
	+1=RIGHT

SUBROUTINE BLKDAT INITIALIZES DATA IN LABELED COMMON BLOCKS (BLOCK DATA)

SUBROUTINE CALPTH CALCULATES AN INTERSECTION PATH WITHIN THE INTERSECTION AND CHECKS ITS LEGALITY
(CALLED FROM FNDPTH)
(CALLS IROTX STRLFT STRSTR STRRGH UTURNL UTURNR
LTLTGE LTLTLT LTGEGE LTGELT RTLTGE RTLTLT
RTGEGE RTGELT IAND ABORTR FIND)

IANGLE	DIRECTION OF PATH
ILNI	LANE NUMBER RELATIVE TO THE FIRST INBOUND LANE THAT HAS A TURN CODE THAT MATCHES TURN CODE OF THE PATH (ILN=LNI)
ILNO	LANE NUMBER RELATIVE TO THE FIRST OUTBOUND LANE THAT HAS A TURN CODE THAT MATCHES TURN CODE OF THE PATH (JLN=LNJ)
ITURN	TURN CODE OF INBOUND LANE
JTURN	TURN CODE FOR OUTBOUND LANE
KANGLE	ANGLE BETWEEN INBOUND AND OUTBOUND LANES (SUPPLEMENT OF JANGLE)
LAZIM	JAZIM + 180 (THE REVERSE OF OUTBOUND APPROACH)
LN	ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED [1-50]
LNI	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF FIRST INBOUND LANE WHICH HAS A TURN CODE THAT MATCHES THE TURN CODE OF THE PATH [1-6]
LNJ	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF FIRST OUTBOUND LANE WHICH HAS A TURN CODE THAT MATCHES THE TURN CODE OF THE PATH [1-6]
LNN	DO LOOP COUNTER USED TO COUNT LNI AND LNJ BACKWARD (RIGHT TO LEFT) FOR CHECKING RIGHT TURNS OR THE LAST LANE OF AN APPROACH
MSG907(19)	ERROR MESSAGE
MSG908(19)	ERROR MESSAGE
MAZIM	AZIMUTH OF OUTBOUND APPROACH
MTURN	TURN CODE OF LANE (SAME AS LTURN IN ENTITY LANE)
NDEGST	NUMBER OF DEGREES LEFT OR RIGHT OF STRAIGHT FOR PATH TO BE CONSIDERED STRAIGHT (DEFAULT IS 20) [0-45]
NDEGUT	NUMBER OF DEGREES LESS THAN 180 FOR PATH TO BE CONSIDERED AS A U-TURN (DEFAULT IS 10) [0-45]

SUBROUTINE CATOAC INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM FNDCON)
(CALLS XVAL ADDAA ABORTR)

A	FIRST TERM OF QUADRATIC EQUATION FOR INTERSECTION OF TWO ARCS
B	SECOND TERM OF QUADRATIC EQUATION FOR INTERSECTION OF TWO ARCS
C	THIRD TERM OF QUADRATIC EQUATION FOR INTERSECTION OF TWO ARCS
CONVAR(12)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
MSG917(8)	ERROR MESSAGE
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
RADICL	VALUE OF RADICAL FOR SQRT FUNCTION
R1	RADIUS OF ARC OF FIRST PATH
R1SQ	SQUARE OF RADIUS OF ARC OF FIRST PATH
R2	RADIUS OF ARC OF SECOND PATH
R2SQ	SQUARE OF RADIUS OF ARC OF SECOND PATH
X1	X COORDINATE OF THE CENTER OF THE ARC OF THE FIRST PATH
X2	X COORDINATE OF THE CENTER OF THE ARC OF THE SECOND PATH
X2X1SQ	SQUARE OF THE DIFFERENCE IN X COORDINATES OF THE CENTERS OF ARCS
Y1	Y COORDINATE OF THE CENTER OF THE ARC OF THE FIRST PATH
Y1SQ	SQUARE OF THE Y COORDINATE OF THE CENTER OF THE ARC OF THE FIRST PATH
Y2	Y COORDINATE OF THE CENTER OF THE ARC OF THE SECOND PATH
Y2SQ	SQUARE OF THE Y COORDINATE OF THE CENTER OF THE ARC OF THE SECOND PATH
Y2Y1SQ	SQUARE OF THE DIFFERENCE IN Y COORDINATES OF THE CENTERS OF ARCS

SUBROUTINE CATOLC CHECKS FOR CONFLICTS BETWEEN THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM FNDCON)
(CALLS ADDAL)

A	FIRST TERM OF QUADRATIC EQUATION FOR INTERSECTION OF AN ARC AND A LINE
B	SECOND TERM OF QUADRATIC EQUATION FOR INTERSECTION OF AN ARC AND A LINE
C	THIRD TERM OF QUADRATIC EQUATION FOR INTERSECTION OF AN ARC AND A LINE
CONVAR(12)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERE 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
RADICL	VALUE OF RADICAL FOR SQRT FUNCTION
X	DISTANCE BETWEEN POINTS OF CONFLICT
XB	Y INTERCEPT OF LINE
XM	SLOPE OF THE LINE

SUBROUTINE CHKPTH CHECKS EACH INBOUND LANE THAT IS AVAILABLE AT THE INTERSECTION TO SEE IF AN INTERSECTION PATH WAS CALCULATED FOR EACH TURNING MOVEMENT SPECIFIED FOR THE INBOUND LANE
(CALLED FROM EXEC)
(CALLS LSHIFT IAND ABORTR EXTRAC FIND)

IPTURN	PATH TURN CODE [1-8] 1= RIGHT 2= STRAIGHT 4= LEFT 8=U-TURN
ITURNC	TURN CODE DESCRIPTION (INDEXED BY ITEST) (1)=R (2)=S (3)=L (4)=U
JPINT	INDEX NUMBER FOR PATH ENTITY OF PATH BEING PROCESSED [1-125]
JTEST	TURN CODE FOR TESTING (WHEN ITEST=1 THEN JTEST=1=RIGHT TURN) (WHEN ITEST=2 THEN JTEST=2=STRAIGHT) (WHEN ITEST=3 THEN JTEST=4=LEFT TURN) (WHEN ITEST=4 THEN JTEST=8=U-TURN)
MSG911(12)	ERROR MESSAGE
MSG912(15)	ERROR MESSAGE

SUBROUTINE CLTOAC CHECKS FOR INTERSECTION CONFLICTS BETWEEN THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE ARC PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM FNDCON)
(CALLS ADDLA)

A	FIRST TERM OF QUADRATIC EQUATION FOR INTERSECTION OF A LINE AND AN ARC
B	SECOND TERM OF QUADRATIC EQUATION FOR INTERSECTION OF A LINE AND AN ARC
C	THIRD TERM OF QUADRATIC EQUATION FOR INTERSECTION OF A LINE AND AN ARC
CONVAR(12)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
RADICL	VALUE OF RADICAL FOR SQRT FUNCTION
X	DISTANCE BETWEEN POINTS OF CONFLICT
XB	Y INTERCEPT OF LINE
XM	SLOPE OF LINE

SUBROUTINE CLTOLC CHECKS FOR INTERSECTION CONFLICTS BETWEEN THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AND THE LINE PORTION OF THE INTERSECTION PATH BEING CHECKED AGAINST
(CALLED FROM FNDCON)
(CALLS LTOL AZIM36 ADDCON)

AZ1	AZIMUTH OF FIRST LINE
AZ2	AZIMUTH OF SECOND LINE
CONVAR(12)	CONFLICT VARIABLES FOR CONCURRENT USAGE
IBAND	INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED 1=MAIN PATH 2=BAND ONE FOOT LEFT OF MAIN PATH 3=BAND ONE FOOT RIGHT OF MAIN PATH 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH
IFS	WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH
IL1	DISTANCE OF POINT OF CONFLICT FROM START OF FIRST PATH

IL2	DISTANCE OF POINT OF CONFLICT FROM START OF SECOND PATH
ITEST	TEST WHETHER THE POINT OF INTERSECTION LIES ON THE LINES
JFS	WHETHER FIRST OR SECOND LINE OR ARC OF SECOND PATH
NC	NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED
X1	X COORDINATE OF START OF LINE OF FIRST PATH
X2	X COORDINATE OF END OF LINE OF FIRST PATH
X3	X COORDINATE OF START OF LINE OF SECOND PATH
X4	X COORDINATE OF END OF LINE OF SECOND PATH
Y1	Y COORDINATE OF START OF LINE OF FIRST PATH
Y2	Y COORDINATE OF END OF LINE OF FIRST PATH
Y3	Y COORDINATE OF START OF LINE OF SECOND PATH
Y4	Y COORDINATE OF END OF LINE OF SECOND PATH

SUBROUTINE DRWAPR DRAWS THE APPROACH PLOT
(CALLED FROM INIPLT)
(CALLS DRWBOX DRWARC DRWLIN EXTRAC)

IARC	ENTRY NUMBER FOR ARC ENTITY OF ARC CURRENTLY BEING PROCESSED
IARCN	INDEX NUMBER FOR LARCS ARRAY OF /GEOPRO/ FOR ARC BEING PROCESSED
ILINE	ENTRY NUMBER FOR LINE ENTITY OF LINE CURRENTLY BEING PROCESSED
ILINEN	INDEX NUMBER FOR LLINES ARRAY OF /GEOPRO/ FOR LINE BEING PROCESSED
ISCALE(9)	MESSAGE FOR SCALE FACTOR USED FOR PLOTTING
ISDRC	INDEX NUMBER FOR SDRC COMMON BLOCK OF SIGHT DISTANCE RESTRICTION CURRENTLY BEING PROCESSED
ISDRCN	INDEX NUMBER FOR LSDRC ARRAY OF /GEOPRO/ FOR SIGHT DISTANCE RESTRICTION BEING PROCESSED
IX1	DISTANCE FROM CENTER LINE OF APPROACH TO INSIDE EDGE OF LANE
IX2	DISTANCE FROM CENTER LINE OF APPROACH TO OUTSIDE EDGE OF LANE
JSCALE(4)	MESSAGE FOR SCALE FACTOR USED FOR PLOTTING (CDC ONLY)
JTITLE(8)	80 CHARACTER TITLE FOR GEOMETRY PROCESSOR (CDC ONLY)
NLEFTD	NUMBER OF DIGITS TO THE LEFT OF THE DECIMAL POINT
X	X COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
XBRDR	BORDER OF PLOT IN X COORDINATE DIRECTION (INCHES)
XPAGE	X COORDINATE OF POINT (INCHES)
X1	X COORDINATE OF BEGINING OF LINE
X2	X COORDINATE OF END OF LINE
Y	Y COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
YBRDR	BORDER OF PLOT IN Y COORDINATE DIRECTION (INCHES)
YPAGE	Y COORDINATE OF POINT (INCHES)
Y1	Y COORDINATE OF BEGINING OF LINE
Y2	Y COORDINATE OF END OF LINE

SUBROUTINE DRWARC DRAWS AN ARC ON THE PLOT PAGE
(CALLED FROM DRWAPR DRWINT DRWUTA DRWPTH)

ADD	NUMBER OF DEGREES TO ADD TO BEGINING AZIMUTH TO GET CURRENT AZIMUTH
ADDAZ	NUMBER OF DEGREES TO ADD EACH INCREMENT (MINIMUM OF 1/10 OF TOTAL AND 10 DEGREES)
DEG	ANGLE OF ARC (EAST ZERO AND COUNTER-CLOCKWISE POSITIVE)
DRWVAR(72)	DRAW VARIABLE FOR CONCURRENT USAGE
IADD	ABSOLUTE ROUNDED VALUE OF ADD
IAZARC	BEGINING AZIMUTH OF ARC
IPEN	PEN POSITIONING 2=PEN UP 3=PEN DOWN
IRARC	RADIUS OF ARC
ISWARC	SWEEP ANGLE OF ARC
IXARC	X COORDINATE OF CENTER OF ARC
IYARC	Y COORDINATE OF CENTER OF ARC
X	X COORDINATE OF ARC
XPAGE	X COORDINATE OF POINT (INCHES)
Y	Y COORDINATE OF ARC
YPAGE	Y COORDINATE OF POINT (INCHES)

SUBROUTINE DRWARR DRAWS AN ARROW POINTING IN THE IANGLE DIRECTION
(CALLED FROM DRWINT)
(CALLS XROTAX DRWLIN)

DRWVAR(72)	DRAW VARIABLE FOR CONCURRENT USAGE
D0P5	DOUBLE PRECISION 0.5
D2P5	DOUBLE PRECISION 2.5
D3P5	DOUBLE PRECISION 3.5
IANGLE	DIRECTION ARROW POINTS (AZIMUTH)
ICX	X COORDINATE OF CENTER OF ARROW
ICY	Y COORDINATE OF CENTER OF ARROW
ILANE	ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED (1-50)
XBOT	X COORDINATE OF BOTTOM OF ARROW
XLEFT	X COORDINATE OF LEFT POINT OF ARROW
XRGHT	X COORDINATE OF RIGHT POINT OF ARROW
XTOP	X COORDINATE OF TOP OF ARROW
YBOT	Y COORDINATE OF BOTTOM OF ARROW
YLEFT	Y COORDINATE OF LEFT POINT OF ARROW
YRGHT	Y COORDINATE OF RIGHT POINT OF ARROW
YTOP	Y COORDINATE OF TOP OF ARROW

SUBROUTINE DRWBOX DRAWS A BOX FROM IL1 TO IL2 FOR A LANE
(CALLED FROM DRWAPR)
(CALLS IROTAX DRWLIN)

DRWVAR(72)	DRAW VARIABLE FOR CONCURRENT USAGE
IL1	DISTANCE DOWN APPROACH FOR START OF BOX
IL2	DISTANCE DOWN APPROACH FOR END OF BOX
IX1	DISTANCE FROM MEDIAN TO LEFT SIDE OF BOX
IX2	DISTANCE FROM MEDIAN TO RIGHT SIDE OF BOX
X1	X COORDINATE OF LEFT STARTING CORNER OF BOX
X2	X COORDINATE OF RIGHT STARTING CORNER OF BOX
X3	X COORDINATE OF RIGHT ENDING CORNER OF BOX
X4	X COORDINATE OF LEFT ENDING CORNER OF BOX
Y1	Y COORDINATE OF LEFT STARTING CORNER OF BOX
Y2	Y COORDINATE OF RIGHT STARTING CORNER OF BOX
Y3	Y COORDINATE OF RIGHT ENDING CORNER OF BOX
Y4	Y COORDINATE OF LEFT ENDING CORNER OF BOX

SUBROUTINE DRWINT DRAWS THE INTERSECTION PLOT
(CALLED FROM INIPLT FNDPTH)
(CALLS IAND DRWUTA DRWARR DRWARC DRWLIN EXTRAC)

IAL	AZIMUTH FOR LEFT ARROW
IAR	AZIMUTH FOR RIGHT ARROW
IARC	ENTRY NUMBER FOR ARC ENTITY OF ARC CURRENTLY BEING PROCESSED
IARCN	INDEX NUMBER FOR LARCS ARRAY OF /GEOPRO/ FOR ARC BEING PROCESSED
IAS	AZIMUTH FOR STRAIGHT ARROW
ILINE	ENTRY NUMBER FOR LINE ENTITY OF LINE CURRENTLY BEING PROCESSED
ILINEN	INDEX NUMBER FOR LLINES ARRAY OF /GEOPRO/ FOR LINE BEING PROCESSED
ISCALE(9)	MESSAGE FOR SCALE FACTOR USED FOR PLOTTING
ISDRC	INDEX NUMBER FOR SDRC COMMON BLOCK OF SIGHT DISTANCE RESTRICTION CURRENTLY BEING PROCESSED
ISDRCN	INDEX NUMBER FOR LSDRC ARRAY OF /GEOPRO/ FOR SIGHT DISTANCE RESTRICTION BEING PROCESSED
IX1	DISTANCE FROM CENTER LINE OF APPROACH TO INSIDE EDGE OF LANE
IX2	DISTANCE FROM CENTER LINE OF APPROACH TO OUTSIDE EDGE OF LANE
JSCALE(4)	MESSAGE FOR SCALE FACTOR USED FOR PLOTTING (CDC ONLY)
JTITLE(8)	80 CHARACTER TITLE FOR GEOMETRY PROCESSOR (CDC ONLY)
KA	ENTRY NUMBER FOR APPRO ENTITY OF APPROACH BEING PROCESSED (1-12)
KAN	INDEX NUMBER FOR LIBA/LOBA ARRAYS OF /GEOPRO/ OF APPROACH BEING PROCESSED (1-6)
KL	ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED (1-50)
KLN	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF LANE

BEING PROCESSED (1-6)
NLEFTD NUMBER OF DIGITS TO THE LEFT OF THE DECIMAL POINT
X X COORDINATE OF SIGHT DISTANCE RESTRICTION
XBRDR BORDER OF PLOT IN X COORDINATE DIRECTION (INCHES)
XPAGE X COORDINATE OF POINT (INCHES)
X1 X COORDINATE OF BEGINING OF LINE
X2 X COORDINATE OF END OF LINE
Y Y COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
YBRDR BORDER OF PLOT IN Y COORDINATE DIRECTION (INCHES)
YPAGE Y COORDINATE OF POINT (INCHES)
Y1 Y COORDINATE OF BEGINING OF LINE
Y2 Y COORDINATE OF END OF LINE

SUBROUTINE DRWLIN DRAWS A LINE ON THE PLOT PAGE
(CALLED FROM DRWAPR DRWBOX DRWINT DRWUTA DRWAPR DRWPTH)

D DUMMY VARIABLE FOR CALLS TO LTOL
DIST DISTANCE FROM POINT OF INTERSECTION OF LINE WITH BORDER AND OR END POINT
DMIN MINIMUM DISTANCE FROM POINT OF INTERSECTION OF LINE WITH BORDER AND OR END POINT
DRWVAR(72) DRAW VARIABLE FOR CONCURRENT USAGE
XDMIN X COORDINATE OF POINT OF INTERSECTION OF LINE WITH BORDER ASSOCIATED WITH DMIN
XINT X COORDINATE OF POINT OF INTERSECTION OF LINE WITH BORDER
XPAGE X COORDINATE OF POINT (INCHES)
X1 X COORDINATE OF BEGINING OF LINE
X2 X COORDINATE OF END OF LINE
YDMIN Y COORDINATE OF POINT OF INTERSECTION OF LINE WITH BORDER ASSOCIATED WITH DMIN
YINT Y COORDINATE OF POINT OF INTERSECTION OF LINE WITH BORDER
YPAGE Y COORDINATE OF POINT (INCHES)
Y1 Y COORDINATE OF BEGINING OF LINE
Y2 Y COORDINATE OF END OF LINE

SUBROUTINE DRWPTH DRAWS AN INTERSECTION PATH ON THE PLOT PAGE
(CALLED FROM FNDPTH)
(CALLS XROTX DRWLIN DRWARC)

X1 X COORDINATE OF THE BEGINING OF LINE OF PATH
X2 X COORDINATE OF THE END OF LINE OF PATH
Y1 Y COORDINATE OF THE BEGINING OF LINE OF PATH
Y2 Y COORDINATE OF THE END OF LINE OF PATH

SUBROUTINE DRWUTA DRAWS A U-TURN ARROW FOR A LANE
(CALLED FROM DRWINT)
(CALLS XROTAX DRWLIN DRWARC)

DRWVAR(46) DRAW VARIABLE FOR CONCURRENT USAGE
D1P5 DOUBLE PRECISION 1.5
D2P0 DOUBLE PRECISION 2.0
D2P5 DOUBLE PRECISION 2.5
D3P0 DOUBLE PRECISION 3.0
ICX X COORDINATE OF CENTER OF U-TURN ARROW
ICY Y COORDINATE OF CENTER OF U-TURN ARROW
ILANE INDEX NUMBER FOR LTDIRX/LTDIRY ARRAY OF /PLOTTR/ OF LANE TO DRAW U-TURN ARROW
UX1 X COORDINATE OF BEGINING OF FIRST LINE OF ARROW
UX2 X COORDINATE OF END OF FIRST LINE AND BEGINING OF ARC OF ARROW
UX3 X COORDINATE OF END OF ARC AND BEGINING OF SECOND LINE OF ARROW
UX4 X COORDINATE OF END OF SECOND LINE AND POINT OF ARROW HEAD
UX5 X COORDINATE OF END OF FIRST ARROW HEAD
UX6 X COORDINATE OF END OF SECOND ARROW HEAD
UY1 Y COORDINATE OF BEGINING OF FIRST LINE OF ARROW
UY2 Y COORDINATE OF END OF FIRST LINE AND BEGINING OF ARC OF ARROW
UY3 Y COORDINATE OF END OF ARC AND BEGINING OF SECOND LINE OF ARROW
UY4 Y COORDINATE OF END OF SECOND LINE AND POINT OF ARROW HEAD
UY5 Y COORDINATE OF END OF FIRST ARROW HEAD
UY6 Y COORDINATE OF END OF SECOND ARROW HEAD

FUNCTION DTAN FINDS THE DOUBLE PRECISION TANGENT OF VAL
(CALLED FROM LTLTLT LTGELT RTLTLT RTGELT)

DTAN DOUBLE PRECISION TANGENT OF VAL
VAL OPERAND FOR FUNCTION

SUBROUTINE ECHO ECHO-PRINTS THE VALUE OF THE ATTRIBUTES IN EACH ENTRY OF EACH ENTITY
(CALLED FROM ABORTR)
(CALLS EXTRAC)

IENT1 DETAILS OF ARC
IENT2 DETAILS OF LINE
IENT3 DETAILS OF APPROACH
IENT4 DETAILS OF LANE
IENT5 DETAILS OF SIGHT DISTANCE RESTRICTIONS
IENT6 DETAILS OF PATH
IENT7 DETAILS OF CONFLICTS
NUM NUMBER OF ATTRIBUTES IN ENTITY
NUMLAN NUMBER OF INBOUND LANES PLUS NUMBER OF OUTBOUND LANES

SUBROUTINE EXEC CONTROLS THE CALLING OF THE OTHER SUBROUTINES TO PROCESS THE INTERSECTION
(CALLED FROM GEOPRO)
(CALLS ISLCPF READIN WRITAL FNDXYP FNDSDR WRITAP
INIPLT FNDPTH CHKPTH WRITLA FNDCON SRTCON
WRITPA NDXCON WRITCO ABORTR)

IBUF(513) BUFFER FOR TKPLOT FILE (CDC ONLY)
IFET(8) FILE ENVIRONMENT TABLE FOR TKPLOT FILE (CDC ONLY)
IRET RETURN FLAG FOR ISLCPF (CDC ONLY)
0=OK
1=FILE ALREADY ASSIGNED
2=LOW CORE POINTER AREA FULL
MSG(6) ERROR MESSAGE THAT IS PRINTED WHEN SYSTEM ERROR DETECTED (CDC ONLY)
MSGERR(2) ERROR MESSAGE IF ISLCPF ERROR (CDC ONLY)
NRECAD RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)

SUBROUTINE EXTRAC EXTRACTS THE ATTRIBUTES FOR ENTRY IN OF ENTITY IY FROM THE STORAGE STACK AND PUTS THEM IN THE COMMON BLOCK FOR ENTITY IY
(CALLED FROM WRITAL FNDXYP FNDSDR WRITAP DRWAPR
DRWINT CHKPTH WRITLA FNDCON SRTCON
WRITPA NDXCON WRITCO ECHO)
(CALLS LSHIFT IAND SMEP)

IBA LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/ FOR ENTITY IY
ID SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES IN ALL THE ENTITIES
IEA LOCATION OF THE LAST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/ FOR ENTITY IY
IFW LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
IIAT SINGLE INDEX FOR IAT ARRAY OF /ATTB/
IIEN SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
IN ENTRY NUMBER FOR ENTITY IY
ISNAME(2) SUBROUTINE NAME FOR PRINTING (EXTRAC)
IWD LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR

ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN OF ENTITY IY

IY — ENTITY NUMBER
- 1=APPRO
- 2=ARC
- 3=CONFLT
- 4=LANE
- 5=LINE
- 6=PATH
- 7=SDR

NBITS — NUMBER OF BITS PER COMPUTER WORD

NWE — NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE FIND FINDS THE VALUE OF ATTRIBUTE IV OF ENTRY IN OF ENTITY IY IN THE STORAGE STACK AND PUTS IT INTO LOCAL INTEGER IR
(CALLED FROM APPLAR FNDXYP FNDPTH CALPTH ADDPTH CHKPTH ADDCON)
(CALLS LSHIFT IAND SMEP)

I — ABSOLUTE ATTRIBUTE NUMBER

IBA — LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/ FOR ENTITY IY

IE — SMEP ERROR NUMBER

IFW — LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY

IIAT — SINGLE INDEX FOR IAT ARRAY OF /ATTB/

IIEN — SINGLE INDEX FOR IEN ARRAY OF /ENTITY/

IN — ENTRY NUMBER FOR ENTITY IY

IR — LOCAL INTEGER TO BE SET TO THE VALUE OF ATTRIBUTE IV OF ENTRY IN OF ENTITY IY

ISNAME(2) — SUBROUTINE NAME FOR PRINTING (FIND)

IV — ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)

IWD — LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN OF ENTITY IY

IY — ENTITY NUMBER
- 1=APPRO
- 2=ARC
- 3=CONFLT
- 4=LANE
- 5=LINE
- 6=PATH
- 7=SDR

NWE — NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE FNDCON FINDS THE INTERSECTION CONFLICTS BETWEEN THE INTERSECTION PATHS
(CALLED FROM EXEC)
(CALLS BAND CLTOLC CLTOAC CATOLC CATOAC ABORTR EXTRAC)

IBAND — INDEX NUMBER OF THE BAND FOR THE PATH BEING CONSIDERED
- 1=MAIN PATH
- 2=BAND ONE FOOT LEFT OF MAIN PATH
- 3=BAND ONE FOOT RIGHT OF MAIN PATH
- 4=BAND ICLOSE DISTANCE LEFT OF MAIN PATH
- 5=BAND ICLOSE DISTANCE RIGHT OF MAIN PATH

IFS — WHETHER FIRST OR SECOND LINE OR ARC OF FIRST PATH

JGEOCP — SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN GEOCP

MIBL — LINKING INBOUND LANE NUMBER OF PATH BEING CHECKED (1-50)

MLCH — ILCH OF PATH BEING CHECKED

MOA — IOA OF PATH BEING CHECKED

MOBL — LOBL OF PATH BEING CHECKED

MPTHP1 — MPTH PLUS ONE

MPTURN — IPTURN OF PATH BEING CHECKED

MSG913(11) — ERROR MESSAGE

NC — NUMBER OF POINTS OF CONFLICT BETWEEN THE INTERSECTION PATHS BEING CHECKED

NPM1 — TOTAL NUMBER OF PATHS IN THE INTERSECTION MINUS ONE

SUBROUTINE FNDPTH FINDS THE INTERSECTION PATHS WITHIN THE INTERSECTION
(CALLED FROM EXEC)
(CALLS DRWINT CALPTH ADDPTH DRWPTH ABORTR FIND)

IENT6 — SINGLE DIMENSIONAL ARRAY EQUIVALENCED TO ALL ATTRIBUTES IN ENTITY PATH

MSG906(9) — ERROR MESSAGE

NUM — NUMBER OF ATTRIBUTES IN ENTITY PATH

SUBROUTINE FNDSDR FINDS THE SIGHT DISTANCE RESTRICTIONS BETWEEN THE INBOUND APPROACHES
(CALLED FROM EXEC)
(CALLS XROTAX LTOL LDOWN HEADER ABORTR EXTRAC REPACK STORE)

DUM — DUMMY VARIABLE FOR CALL TO LTOL

DX1 — DISTANCE FROM CENTER LINE OF APPROACH TO CENTER OF APPROACH (CENTER OF ALL APPROACH LANES) BEING CHECKED

DX2 — DISTANCE FROM CENTER LINE OF APPROACH TO THE CENTER OF APPROACH (CENTER OF ALL APPROACH LANES) BEING CHECKED AGAINST

DY1 — DISTANCE DOWN APPROACH BEING CHECKED

IAZIM — AZIMUTH OF APPROACH BEING CHECKED

IMAXL — MAXIMUM LENGTH OF LANE FOR APPROACH BEING CHECKED

INDEX — COUNTER FOR POINTS ALONG APPROACH BEING CHECKED

ISDRC — INDEX NUMBER FOR SDRC COMMON BLOCK OF SIGHT DISTANCE RESTRICTION CURRENTLY BEING PROCESSED

ISDRCN — INDEX NUMBER FOR LSDRC ARRAY OF /GEOPRO/ FOR SIGHT DISTANCE RESTRICTION BEING PROCESSED

ISEE — DISTANCE VISIBLE DOWN APPROACH BEING CHECKED

ISTART — BEGINING POINT FOR AREA ON LANE FROM WHICH OTHER LANE IS OBSERVED

ISTOP — END POINT FOR AREA ON LANE FROM WHICH OTHER LANE IS OBSERVED

ITEST — TEST TO CHECK IF LINE FROM (X1,Y1) TO (X4,Y4) INTERSECTS WITH LINE FROM (X2,Y2) TO (X3,Y3)
- 0=YES
- 1=NO

IXCLAP — IAPX FOR APPROACH BEING CHECKED

IYCLAP — IAPY FOR APPROACH BEING CHECKED

JMAXL — MAXIMUM LENGTH OF LANE FOR APPROACH BEING CHECKED AGAINST

MAXSEE — MAXIMUM DISTANCE VISIBLE DOWN APPROACH BEING CHECKED AGAINST

MSG903(10) — ERROR MESSAGE

MSG904(17) — ERROR MESSAGE

MSG905(19) — ERROR MESSAGE

NSDRAP — NUMBER OF SIGHT DISTANCE RESTRICTIONS FOR APPROACH BEING CHECKED

XBIG — VALUE FOR A VERY LONG DISTANCE DOWN AN APPROACH (2000 FEET)

XFROM — X COORDINATE OF THE POINT WHERE THE DRIVER WILL BE LOOKING FROM

XINT — X COORDINATE OF POINT OF INTERSECTION OF APPROACHES

XSDR — X COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION

X1 — X COORDINATE OF BEGINING OF APPROACH BEING CHECKED

X2 — X COORDINATE OF BEGINING OF APPROACH BEING CHECKED AGAINST

X3 — X COORDINATE OF A POINT 2000 FEET DOWN APPROACH BEING CHECKED AGAINST

X4 — X COORDINATE OF A POINT 2000 FEET DOWN APPROACH BEING CHECKED

YFROM — Y COORDINATE OF THE POINT WHERE THE DRIVER WILL BE LOOKING FROM

YINT — Y COORDINATE OF POINT OF INTERSECTION OF APPROACHES

YSDR — Y COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION

Y1 — Y COORDINATE OF BEGINING OF APPROACH BEING CHECKED

Y2 — Y COORDINATE OF BEGINING OF APPROACH BEING CHECKED AGAINST

Y3 — Y COORDINATE OF A POINT 2000 FEET DOWN APPROACH BEING CHECKED AGAINST

Y4 — Y COORDINATE OF A POINT 2000 FEET DOWN APPROACH BEING CHECKED

SUBROUTINE FNDXYP FINDS THE X AND Y COORDINATES FOR A POINT AT THE MIDDLE AND END OF EACH INBOUND LANE AND AT THE MIDDLE AND START OF EACH OUTBOUND LANE THAT IS AVAILABLE AT THE INTERSECTION,

```
                FINDS THE BOUNDARIES FOR PLOTTING, AND FINDS THE PLOT
                SCALE FACTORS
                (CALLED FROM EXEC)
                (CALLS XROTAI HEADER ABORTR EXTRAC STORE  FIND)

   DAI          DISTANCE DOWN THE APPROACH TO THE CENTER OF DIRECTION ARROW
   DW           HALF THE WIDTH OF LANE
   DXI          DISTANCE FROM THE CENTER LINE OF THE APPROACH TO THE
                CENTER OF THE LANE BEING PROCESSED
   DYI          DISTANCE DOWN APPROACH FOR END OF LANE FOR INBOUND LANES
                AND START OF LANE FOR OUTBOUND LANES
   IDX          DISTANCE FROM MEDIAN TO CENTER OF LANE (FEET) [0,90]
   IX           X COORDINATE OF INSIDE AND OUTSIDE EDGE OF END OF LANE FOR
                INBOUND LANES AND BEGINING OF LANE FOR OUTBOUND LANES
   IY           Y COORDINATE OF INSIDE AND OUTSIDE EDGE OF END OF LANE FOR
                INBOUND LANES AND BEGINING OF LANE FOR OUTBOUND LANES
   LGEOM1       LGEOM(1) FOR LANE
   LGEOM2       LGEOM(2) FOR LANE
   LGEOM3       LGEOM(3) FOR LANE
   LGEOM4       LGEOM(4) FOR LANE
   LWID         WIDTH OF LANE (FEET) [8,15]
   MSG901(20)   ERROR MESSAGE
   MSG902(21)   ERROR MESSAGE
   NSCALE       NUMBER OF PLOT SCALE FACTORS
   PWID         PLOT PAPER WIDTH
   SA           PLOT SCALE FACTOR FOR APPROACH
   SCALEF(11)   PLOT SCALE FACTORS (FT/IN)
   SI           PLOT SCALE FACTOR FOR INTERSECTION

PROGRAM    GEOPRO GEOMETRY PROCESSER FOR THE TEXAS TRAFFIC SIMULATION PACKAGE
                  (COLEASE GENERATED)
                  (CALLS LSHIFT EXEC)

   IB             NUMBER OF BITS TO LEFT SHIFT ATTRIBUTE MASK FOR PROPER
                  POSITIONING
   NBITS          NUMBER OF BITS IN COMPUTER WORD

SUBROUTINE HEADER SKIPS TO THE TOP OF A NEW PAGE, PRINTS THE HEADER MESSAGE,
                  AND PRINTS THE TITLE FOR GEOPRO
                  (CALLED FROM READIN FNDXYP FNDSDR WRITPA WRITCO)

FUNCTION    IAND  FINDS THE LOGICAL PRODUCT (AND) OF THE PARAMETERS
                  (CALLED FROM DRWINT CALPTH CHKPTH EXTRAC REPACK STORE FIND)

FUNCTION    ICHKA CHECKS TO SEE IF AZIM LIES BETWEEN NBA AND NBA+NDA AND
                  RETURNS DA
                  (CALLED FROM ADDLA ADDAL ADDAA)
                  (CALLS ABORTR)

   AZIM           AZIMUTH OF LINE TANGENT TO ARC AT CONFLICT
   BZIM           LOCAL VALUE OF AZIM
   CONVAR(86)     CONFLICT VARIABLES FOR CONCURRENT USAGE
   DA             ANGLE BETWEEN LINE TANGENT TO ARC AND INITIAL AZIMUTH OF ARC
   ICHKA          DOES AZIM LIE BETWEEN IBA AND IBA+IDA
                      0=YES
                      1=NO
   MSG916(13)     ERROR MESSAGE
   NBA            BEGINING AZIMUTH OF ARC
   NDA            SWEEP ANGLE OF ARC

FUNCTION    ICHKL CHECKS TO SEE IF (XINT,YINT) LIES BETWEEN (IX1,IY1) AND
                  (IX2,IY2)
                  (CALLED FROM ADDLA ADDAL)
   CONVAR(86)     CONFLICT VARIABLES FOR CONCURRENT USAGE

   ICHKL          DOES (XINT,YINT) LIE ON LINE FROM (IX1,IY1) TO (IX2,IY2)
                      0=YES
                      1=NO
   IX1            X COORDINATE OF BEGINING OF LINE OF PATH
   IX2            X COORDINATE OF END OF LINE OF PATH
   IY1            Y COORDINATE OF BEGINING OF LINE OF PATH
   IY2            Y COORDINATE OF END OF LINE OF PATH
   XINT           X COORDINATE OF POINT OF INTERSECTION
   YINT           Y COORDINATE OF POINT OF INTERSECTION

SUBROUTINE INIPLT INITIALIZES PLOTTING
                  (CALLED FROM EXEC)
                  (CALLS DRWAPR DRWINT)

   IBUF           BUFFER FOR PLOTS (IBM ONLY)

FUNCTION    INOT  FINDS THE COMPLIMENT (NOT) OF THE PARAMETER
                  (CALLED FROM REPACK STORE)

FUNCTION    IOR   FINDS THE LOGICAL SUM (OR) OF THE PARAMETERS
                  (CALLED FROM REPACK STORE)

SUBROUTINE IROTAX ROTATES AN INTEGER VECTOR BY AN AZIMUTH, ADDS AN INTEGER
                  COORDINATE, AND RETURNS A REAL COORDINATE
                  (CALLED FROM DRWBOX)
                  (CALLS IROTX)

   IAX            X COORDINATE OF POINT TO BE ADDED
   IAY            Y COORDINATE OF POINT TO BE ADDED
   IAZIM          AZIMUTH FOR ROTATION
   IX             X VECTOR BEFORE ROTATION
   IY             Y VECTOR BEFORE ROTATION
   RX             X COORDINATE AFTER ROTATION AND ADDITION
   RY             Y COORDINATE AFTER ROTATION AND ADDITION

SUBROUTINE IROTX  ROTATES AN INTEGER VECTOR BY AN AZIMUTH AND RETURNS A REAL
                  VECTOR
                  (CALLED FROM CALPTH IROTAX)
                  (CALLS XROTX)

   IAZIM          AZIMUTH FOR ROTATION
   IX             X VECTOR BEFORE ROTATION
   IY             Y VECTOR BEFORE ROTATION
   RX             X VECTOR AFTER ROTATION
   RY             Y VECTOR AFTER ROTATION
   X              FLOATING POINT VALUE OF INTEGER IX
   Y              FLOATING POINT VALUE OF INTEGER IY

FUNCTION    ISLCPF SETS UP THE LOW CORE POINTERS AND FILE ENVIRONMENT TABLE
                  FOR A FILE AT EXECUTION TIME (CDC ONLY)
                  (CALLED FROM EXEC)

FUNCTION    LDOWN FINDS THE DISTANCE FROM (X2,Y2) TO (XINT,YINT) IF LINE A
                  FROM (X1,Y1) THROUGH (XSDR,YSDR) INTERSECTS WITH LINE B
                  FROM (X2,Y2) TO (X3,Y3)
                  (CALLED FROM FNDSDR)

   LDOWN          DISTANCE A FROM (X2,Y2) TO (XINT,YINT) IF LINE A FROM (X1,Y1)
                  THROUGH (XSDR,YSDR) INTERSECTS LINE B FROM (X2,Y2) TO (X3,Y3)
                  ( (XSDR,YSDR) MUST LIE BETWEEN (X1,Y1) AND (XINT,YINT)
                  AND (XINT,YINT) MUST LIE BETWEEN (X2,Y2) AND (X3,Y3) )
                      0=NO INTERSECTION
   XBA            Y INTERCEPT OF LINE A
   XBB            Y INTERCEPT OF LINE B
   XINT           X COORDINATE OF POINT OF INTERSECTION (POINT JUST VISIBLE)
```

```
XMA            SLOPE OF LINE A
XMB            SLOPE OF LINE B
XSDR           X COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
X1             X COORDINATE OF POINT OF OBSERVATION
X2             X COORDINATE OF BEGINING OF OBSERVED APPROACH
X3             X COORDINATE OF END OF OBSERVED APPROACH
YINT           Y COORDINATE OF POINT OF INTERSECTION (POINT JUST VISIBLE)
YSDR           Y COORDINATE OF POINT OF SIGHT DISTANCE RESTRICTION
Y1             Y COORDINATE OF POINT OF OBSERVATION
Y2             Y COORDINATE OF BEGINING OF OBSERVED APPROACH
Y3             Y COORDINATE OF END OF OBSERVED APPROACH


FUNCTION   LSHIFT LEFT OR RIGHT SHIFT A COMPUTER WORD
                 (CALLED FROM GEOPRO CHKPTH EXTRAC REPACK STORE FIND)


SUBROUTINE LTGEGE CALCULATES AN INTERSECTION PATH THAT IS A LEFT TURN GE 90
                 DEGREES AND ADY GE YC WITH RADIUS RC
                 (CALLED FROM CALPTH)
                 (CALLS ZEROP3 ZEROP4 MAXVEL ZEROP1)

  DY             DIFFERENCE BETWEEN ADY AND YC


SUBROUTINE LTGELT CALCULATES AN INTERSECTION PATH THAT IS A LEFT TURN GE 90
                 DEGREES AND ADY LT YC
                 (CALLED FROM CALPTH)
                 (CALLS ZEROP3 DTAN   ZEROP1 MAXVEL ZEROP4)

  DY             DIFFERENCE BETWEEN ADY AND YC
  KANGLE


SUBROUTINE LTLTGE CALCULATES AN INTERSECTION PATH THAT IS A LEFT TURN LT 90
                 DEGREES AND ADY GE YC WITH RADIUS RC
                 (CALLED FROM CALPTH)
                 (CALLS ZEROP3 ZEROP4 MAXVEL ZEROP1)

  DY             DIFFERENCE BETWEEN ADY AND YC


SUBROUTINE LTLTLT CALCULATES AN INTERSECTION PATH THAT IS A LEFT TURN LT 90
                 DEGREES AND ADY LT YC
                 (CALLED FROM CALPTH)
                 (CALLS ZEROP3 DTAN   ZEROP1 MAXVEL ZEROP4)

  A              FIRST TERM OF QUADRATIC EQUATION FOR RADIUS
  ANGLE2         CALCULATED ANGLE OF ROTATION OF FIRST ARC
  ANGLE3         CALCULATED ANGLE OF ROTATION OF SECOND ARC
  B              SECOND TERM OF QUADRATIC EQUATION FOR RADIUS
  C              THIRD TERM OF QUADRATIC EQUATION FOR RADIUS
  COSJA          COSINE OF THE ANGLE THE VEHICLE TURNS THROUGH
  DY             DIFFERENCE BETWEEN ADY AND YC
  KANGLE         COMPLEMENT OF JANGLE TO FIND DISTANCE ALONG X AXIS
  KANGLE2        ANGLE OF ROTATION OF FIRST ARC OF PATH
  KANGLE3        ANGLE OF ROTATION OF SECOND ARC OF PATH
  RADICL         VALUE UNDER SQUARE ROOT FOR QUADRATIC
  SINJA          SINE OF THE ANGLE THE VEHICLE TURNS THROUGH


FUNCTION   LTOL   TESTS IF LINE A FROM (X1,Y1) TO (X2,Y2) INTERSECTS WITH
                 LINE B FROM (X3,Y3) TO (X4,Y4)
                 (CALLED FROM FNDSDR CLTOLC)

  CLOSE          VALUE USED FOR TESTING IF TWO LINES ARE THE SAME IF
                 PARALLEL
  DRWVAR(96)     DRAW VARIABLE FOR CONCURRENT USAGE
  LTOL           DOES LINE A FROM (X1,Y1) TO (X2,Y2) INTERSECT WITH LINE B
                 FROM (X3,Y3) TO (X4,Y4)
                 ( (XINT,YINT) MUST LIE BETWEEN (X1,Y1) AND (X2,Y2)
                 AND (XINT,YINT) MUST LIE BETWEEN (X3,Y3) AND (X4,Y4) )
                    0=YES
                    1=NO
  XBA            Y INTERCEPT OF LINE A
  XBB            Y INTERCEPT OF LINE B
  XI1            X COORDINATE OF FIRST POINT OF INTERSECTION
  XI2            X COORDINATE OF SECOND POINT OF INTERSECTION
                 (IF PARALLEL AND CLOSE)
  XMA            SLOPE OF LINE A
  XMB            SLOPE OF LINE B
  X1             X COORDINATE OF BEGINING OF LINE A
  X2             X COORDINATE OF END OF LINE A
  X3             X COORDINATE OF BEGINING OF LINE B
  X4             X COORDINATE OF END OF LINE B
  YI1            Y COORDINATE OF FIRST POINT OF INTERSECTION
  YI2            Y COORDINATE OF SECOND POINT OF INTERSECTION
                 (IF PARALLEL AND CLOSE)
  Y1             Y COORDINATE OF BEGINING OF LINE A
  Y2             Y COORDINATE OF END OF LINE A
  Y3             Y COORDINATE OF BEGINING OF LINE B
  Y4             Y COORDINATE OF END OF LINE B


FUNCTION   MAXVEL FINDS THE MAXIMUM VELOCITY FOR AN INTERSECTION PATH BASED ON
                 THE MAXIMUM SAFE SIDE FRICTION AND THE RADIUS OF THE
                 INTERSECTION PATH
                 (CALLED FROM STRLFT STRRGH UTURNL UTURNR LTLTGE LTLTLT
                             LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT)

  A              FIRST TERM OF QUADRATIC EQUATION FOR VELOCITY
  AL             FIRST TERM CONSTANT OF EQUATION FOR LINEAR SEGMENT
                 OF SIDE FRICTION (F = AL + BL*V)
  AP             FIRST TERM CONSTANT OF EQUATION FOR PARABOLIC SEGMENT
                 OF SIDE FRICTION (F = AP + BP*V + CP*V**2)
  B              SECOND TERM OF QUADRATIC EQUATION FOR VELOCITY
  BL             SECOND TERM CONSTANT OF EQUATION FOR LINEAR SEGMENT
                 OF SIDE FRICTION (F = AL + BL*V)
  BP             SECOND TERM CONSTANT OF EQUATION FOR PARABOLIC SEGMENT
                 OF SIDE FRICTION (F = AP + BP*V + CP*V**2)
  C              THIRD TERM OF QUADRATIC EQUATION FOR VELOCITY
  CALVEL(38)     TEMPORARY /ZTEMPD/ STORAGE
  CP             THIRD TERM CONSTANT OF EQUATION FOR PARABOLIC SEGMENT
                 OF SIDE FRICTION (F = AP + BP*V + CP*V**2)
  R              RADIUS OF PATH
  VELMPH         VELOCITY IN MILES PER HOUR


SUBROUTINE NDXCON CROSS INDEXES THE INTERSECTION CONFLICTS WITH THE
                 INTERSECTION PATHS
                 (CALLED FROM EXEC)
                 (CALLS ABORTR EXTRAC REPACK)

  IPTH           ENTRY NUMBER FOR PATH ENTITY OF PATH BEING PROCESSED
  MSG918(14)     ERROR MESSAGE


SUBROUTINE READAI READS THE ARC INFORMATION AND CHECKS FOR ERRORS
                 (CALLED FROM READIN)

  IENT2          SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLE
                 IN ENTITY ARC
  IUSED(20)      TEST TO CHECK IF DATA IS REPEATED
                    0=NOT USED
                    1=USED
  LTEST          TEMPORARY TEST FOR NUMBER OF LINES PRINTED ON PAGE
  NUM            NUMBER OF ATTRIBUTES IN ENTITY


SUBROUTINE READAP READS THE APPROACH INFORMATION AND CHECKS FOR ERRORS
                 (CALLED FROM READIN)
```

```
IENT1         SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLE
              IN ENTITY APPRO
IENT4         SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLE
              IN ENTITY LANE
ILT           LEFT TURN LEGAL FROM LANE (READ IN)
IRT           RIGHT TURN LEGAL FROM LANE (READ IN)
IST           STRAIGHT THROUGH LEGAL FROM LANE (READ IN)
ITEST         TEST FOR INBOUND OR OUTBOUND APPROACH DATA
              (HEADWAY DISTRIBUTION NAME) (READ IN)
IUSED(12)     TEST TO CHECK IF DATA IS REPEATED
                  0=NOT USED
                  1=USED
IUT           U-TURN LEGAL FROM LANE (READ IN)
IYES          YES/NO FOR USER SUPPLIED PERCENT OF EACH VEHICLE CLASS
              MAKING UP TRAFFIC STREAM (INBOUND ONLY) (READ IN)
JBLN          SERIAL NUMBER OF INBOUND LANE NUMBER
LLTYPE        TYPE OF LANE
                  1=INBOUND
                  2=OUTBOUND
LTEST         TEMPORARY TEST FOR NUMBER OF LINES PRINTED ON PAGE
NBLANK        CHECKING VALUE FOR BLANK (     )
NEXTL(9)      TEMPORARY STORAGE FOR SECOND LANE DATA
NL            CHECKING VALUE FOR LEFT TURN (L)
NR            CHECKING VALUE FOR RIGHT TURN (R)
NS            CHECKING VALUE FOR STRAIGHT THROUGH (S)
NU            CHECKING VALUE FOR U-TURN (U)
NUM           NUMBER OF ATTRIBUTES IN ENTITY
NYES          CHECKING VALUE FOR YES (YES)
```

```
SUBROUTINE READIN READS INPUT DATA AND CHECKS FOR ERRORS
              (CALLED FROM EXEC)
              (CALLS HEADER APPLAR READAI READAP READIO READLI READ
                     READSI REPACK)
```

```
SUBROUTINE READIO READS THE NUMBER AND LIST OF INBOUND AND OUTBOUND APPROACHES
              AND CHECKS FOR ERRORS
              (CALLED FROM READIN)

IANP1         IAN PLUS 1
NTEST         NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND
              APPROACHES
```

```
SUBROUTINE READLI READS THE LINE INFORMATION AND CHECKS FOR ERRORS
              (CALLED FROM READIN)

IENT5         SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLE
              IN ENTITY LINE
IUSED(20)     TEST TO CHECK IF DATA IS REPEATED
                  0=NOT USED
                  1=USED
LTEST         TEMPORARY TEST FOR NUMBER OF LINES PRINTED ON PAGE
```

```
SUBROUTINE READOP READS THE GEOMETRY PROCESSOR OPTIONS AND CHECKS FOR ERRORS
              (CALLED FROM READIN)

JPATH(2)      PATH OPTION (READ IN)
JPLOT(2)      PLOT OPTION (READ IN)
JSAME(2)      PATH PLOT OPTION (READ IN)
NBLANK        CHECKING VALUE FOR BLANK (     )
NNOPLOT(2)    CHECKING VALUE FOR NO PLOT (NOPLOT  )
NOPT1(2)      CHECKING VALUE FOR OPTION1 PATHS (OPTION1 )
NPLT          CHECKING VALUE FOR PLOT ON 30 INCH PAPER AND BALL POINT PEN
              (PLOT)
NPLTI(2)      CHECKING VALUE FOR PLOT ON 30 INCH PAPER AND INK PEN
              (PLOTI    )
NPRIM(2)      CHECKING VALUE FOR PRIMARY PATHS (PRIMARY )
NSAME         CHECKING VALUE FOR SAME FRAME (SAME)
NSEPAR(2)     CHECKING VALUE FOR SEPARATE FRAME (SEPARATE)
R             MAXIMUM RADIUS FOR PATH BEFORE A STRAIGHT LINE WILL BE
              USED FOR PATH (READ IN)
SA            PLOT SCALE FACTOR FOR APPROACH (READ IN)
SI            PLOT SCALE FACTOR FOR INTERSECTION (READ IN)
```

```
SUBROUTINE READSI READS THE SIGHT DISTANCE RESTRICTION COORDINATE INFORMATION
              AND CHECKS FOR ERRORS
              (CALLED FROM READIN)

IUSED(20)     TEST TO CHECK IF DATA IS REPEATED
                  0=NOT USED
                  1=USED
LTEST         TEMPORARY TEST FOR NUMBER OF LINES PRINTED ON PAGE
```

```
SUBROUTINE REPACK REPACKS THE VALUES OF THE ATTRIBUTES FROM THE COMMON BLOCK
              FOR ENTITY IY INTO ENTRY IN OF ENTITY IY IN THE STORAGE STACK
              (CALLED FROM READIN FNDSDR ADDPTH ADDCON SRTCON NDXCON)
              (CALLS LSHIFT IAND   INOT   IOR    SMEP)

IBA           LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
              FOR ENTITY IY
ID            SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES
              IN ALL THE ENTITIES
IE            SMEP ERROR NUMBER
IEA           LOCATION OF THE LAST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
              FOR ENTITY IY
IFW           LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
              ENTRY IN OF ENTITY IY
IIAT          SINGLE INDEX FOR IAT ARRAY OF /ATTB/
IIEN          SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
ILW           LOCATION OF THE LAST COMPUTER WORD IN THE STORAGE STACK FOR
              ENTRY IN OF ENTITY IY
IN            ENTRY NUMBER FOR ENTITY IY
IR            VALUE OF CURRENT ATTRIBUTE BEING REPACKED
ISNAME(2)     SUBROUTINE NAME FOR PRINTING (REPACK)
IT            ATTRIBUTE I LEFT SHIFTED TO ITS PROPER POSITION FOR STORING
              IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
IV            INDEX NUMBER OF CURRENT ATTRIBUTE BEING REPACKED
IWD           LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
              ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
              OF ENTITY IY
IX            TEST IF ATTRIBUTE I IS OUT OF RANGE FOR ENTITY IY
                 <0=OUT OF RANGE
                  0=OK
                 >0=OUT OF RANGE
IY            ENTITY NUMBER
                  1=APPRO
                  2=ARC
                  3=CONFLT
                  4=LANE
                  5=LINE
                  6=PATH
                  7=SDR
NWE           NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY
```

```
SUBROUTINE RTGEGE CALCULATES AN INTERSECTION PATH THAT IS A RIGHT TURN GE 90
              DEGREES AND ADY GE YC WITH RADIUS RC
              (CALLED FROM CALPTH)
              (CALLS ZEROP3 ZEROP4 MAXVEL ZEROP1)

DY            DIFFERENCE BETWEEN ADY AND YC
```

```
SUBROUTINE RTGELT CALCULATES AN INTERSECTION PATH THAT IS A RIGHT TURN GE 90
              DEGREES AND ADY LT YC
              (CALLED FROM CALPTH)
              (CALLS ZEROP3 DTAN   ZEROP1 MAXVEL ZEROP4)
```

```
   DY              DIFFERENCE BETWEEN ADY AND YC
   KANGLE          ANGLE OF ROTATION OF PATH

SUBROUTINE RTLTGE CALCULATES AN INTERSECTION PATH THAT IS A RIGHT TURN LT 90
                  DEGREES AND ADY GE YC WITH RADIUS RC
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP3 ZEROP4 MAXVEL ZEROP1)

   DY              DIFFERENCE BETWEEN ADY AND YC

SUBROUTINE RTLTLT CALCULATES AN INTERSECTION PATH THAT IS A RIGHT TURN LT 90
                  DEGREES AND ADY LT YC
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP3 DTAN   ZEROP1 MAXVEL ZEROP4)

   A               FIRST TERM OF QUADRATIC EQUATION FOR RADIUS
   ANGLE2          CALCULATED ANGLE OF ROTATION OF FIRST ARC
   ANGLE3          CALCULATED ANGLE OF ROTATION OF SECOND ARC
   B               SECOND TERM OF QUADRATIC EQUATION FOR RADIUS
   C               THIRD TERM OF QUADRATIC EQUATION FOR RADIUS
   COSJA           COSINE OF THE ANGLE THE VEHICLE TURNS THROUGH
   DY              DIFFERENCE BETWEEN ADY AND YC
   KANGLE          COMPLEMENT OF JANGLE TO FIND DISTANCE ALONG X AXIS
   KANGLE2         ANGLE OF ROTATION OF FIRST ARC OF PATH
   KABGLE3         ANGLE OF ROTATION OF SECOND ARC OF PATH
   RADICL          VALUE UNDER SQUARE ROOT FOR QUADRATIC
   SINJA           SINE OF THE ANGLE THE VEHICLE TURNS THROUGH

SUBROUTINE SMEP   SYSTEM MESSAGE ERROR PROCESSOR FOR COLEASE SUBROUTINES
                  (CALLED FROM EXTRAC FIND   REPACK STORE)
                  (CALLS ABORTR)

   IE              SMEP ERROR NUMBER
   IERROR(8)       ERROR MESSAGE FOR ABORTR
   IN              ENTRY NUMBER FOR ENTITY IY
   IR              VALUE OF ATTRIBUTE BEING PROCESSED
   ISNAME(2)       SUBROUTINE NAME FOR PRINTING
   IV              ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)
   IY              ENTITY NUMBER
                       1=APPRO
                       2=ARC
                       3=CONFLT
                       4=LANE
                       5=LINE
                       6=PATH
                       7=SDR

SUBROUTINE SRTCON SORTS THE INTERSECTION CONFLICTS FOR EACH INTERSECTION PATH
                  BY THE DISTANCE DOWN THE INTERSECTION PATH TO THE
                  INTERSECTION CONFLICT
                  (CALLED FROM EXEC)
                  (CALLS EXTRAC REPACK)

   I               INDEX NUMBER FOR IDIST/IGEOCP ARRAYS OF PATH FOR SORTING
   IDIST(60)       DISTANCE TO POINT OF CONFLICT ALONG PATH
   IPN             INDEX NUMBER FOR ICONP/ICOND ARRAY OF ENTITY CONFLT OF PATH
                   BEING PROCESSED
   ITEMP           TEMPORARY STORAGE FOR SORTING
   J               INDEX NUMBER FOR IDIST/IGEOCP ARRAYS OF NEXT PATH NOT SORTED
   JCON            ENTRY NUMBER FOR CONFLT ENTITY OF CONFLICT BEING PROCESSED

SUBROUTINE STORE  STORES THE VALUE OF LOCAL INTEGER IR INTO ATTRIBUTE IV OF
                  ENTRY IN OF ENTITY IY IN THE STORAGE STACK
                  (CALLED FROM APPLAR FNDXYP FNDSDR ADDPTH ADDCON)
                  (CALLS LSHIFT IAND   INOT   IOR    SMEP)

   IBA             LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
                   FOR ENTITY IY
   ID              SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES
                   IN ALL THE ENTITIES
   IE              SMEP ERROR NUMBER
   IFW             LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
                   ENTRY IN OF ENTITY IY
   IIAT            SINGLE INDEX FOR IAT ARRAY OF /ATTB/
   IIEN            SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
   IN              ENTRY NUMBER FOR ENTITY IY
   IR              LOCAL INTEGER TO BE STORED IN ATTRIBUTE IV OF ENTRY IN OF
                   ENTITY IY
   ISNAME(2)       SUBROUTINE NAME FOR PRINTING (STORE)
   IT              ATTRIBUTE I LEFT SHIFTED TO ITS PROPER POSITION FOR STORING
                   IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
   IV              ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)
   IWD             LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
                   ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
                   OF ENTITY IY
   IX              TEST IF ATTRIBUTE I IS OUT OF RANGE FOR ENTITY IY
                     <0=OUT OF RANGE
                      0=OK
                     >0=OUT OF RANGE
   IY              ENTITY NUMBER
                       1=APPRO
                       2=ARC
                       3=CONFLT
                       4=LANE
                       5=LINE
                       6=PATH
                       7=SDR
   NWE             NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE STRLFT CALCULATES AN INTERSECTION PATH THAT IS A STRAIGHT THROUGH
                  MOVEMENT THAT VEERS LEFT (EXACTLY 0 DEGREES)
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP1 ZEROP4 MAXVEL)

   ANGLE           ANGLE OF ROTATION OF PATH

SUBROUTINE STRRGH CALCULATES AN INTERSECTION PATH THAT IS A STRAIGHT THROUGH
                  MOVEMENT THAT VEERS RIGHT (EXACTLY 0 DEGREES)
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP1 ZEROP4 MAXVEL)

   ANGLE           ANGLE OF ROTATION OF PATH

SUBROUTINE STRSTR CALCULATES AN INTERSECTION PATH THAT IS A STRAIGHT THROUGH
                  MOVEMENT THAT GOES STRAIGHT FROM THE INBOUND LANE TO THE
                  OUTBOUND LANE
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP2 ZEROP3 ZEROP4)

SUBROUTINE UTURNL CALCULATES AN INTERSECTION PATH THAT IS A U-TURN THAT GOES
                  LEFT (EXACTLY 180 DEGREES)
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP3 MAXVEL ZEROP1 ZEROP4)

SUBROUTINE UTURNR CALCULATES AN INTERSECTION PATH THAT IS A U-TURN THAT GOES
                  RIGHT (EXACTLY 180 DEGREES)
                  (CALLED FROM CALPTH)
                  (CALLS ZEROP3 MAXVEL ZEROP1 ZEROP4)

SUBROUTINE WRITAL WRITES THE TITLE FOR GEOPRO, THE ARC INFORMATION,
                  AND THE LINE INFORMATION ONTO TAPE MODELT FOR SIMPRO
```

(CALLED FROM EXEC)
(CALLS EXTRAC)

IARC — ENTRY NUMBER FOR ARC ENTITY OF ARC BEING PROCESSED
IARCN — INDEX NUMBER FOR LARCS ARRAY OF /GEOPRO/ FOR ARC BEING PROCESSED
ILINE — ENTRY NUMBER FOR LINE ENTITY OF LINE BEING PROCESSED
ILINEN — INDEX NUMBER FOR LLINES ARRAY OF /GEOPRO/ FOR LINE BEING PROCESSED

SUBROUTINE WRITAP WRITES THE APPROACH INFORMATION ONTO TAPE MODELT FOR SIMPRO
(CALLED FROM EXEC)
(CALLS EXTRAC)

SUBROUTINE WRITCO WRITES THE INTERSECTION CONFLICT INFORMATION ONTO TAPE MODELT FOR SIMPRO
(CALLED FROM EXEC)
(CALLS HEADER EXTRAC)

IADD — ADDS LINES DEPENDING ON CONFLICTS TO SKIP TO NEW PAGE

SUBROUTINE WRITLA WRITES THE LANE INFORMATION AND THE SIGHT DISTANCE RESTRICTION INFORMATION ONTO TAPE MODELT FOR SIMPRO
(CALLED FROM EXEC)
(CALLS EXTRAC)

ILANE — ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED [1⌀50]
NUMLAN — NUMBER OF INBOUND PLUS OUTBOUND LANES

SUBROUTINE WRITPA WRITES THE INTERSECTION PATH INFORMATION ONTO TAPE MODELT FOR SIMPRO
(CALLED FROM EXEC)
(CALLS HEADER EXTRAC)

LTEST — TEMPORARY TEST FOR NUMBER OF LINES PRINTED ON PAGE

SUBROUTINE XROTAI ROTATES A REAL VECTOR BY AN AZIMUTH, ADDS AN INTEGER COORDINATE, AND RETURNS AN INTEGER COORDINATE
(CALLED FROM FNDXYP BAND)
(CALLS XROTI)

IAX — X COORDINATE TO BE ADDED
IAY — Y COORDINATE TO BE ADDED
IAZIM — AZIMUTH OF ROTATION
IRX — X VECTOR OF POINT AFTER ROTATION AND ADDITION
IRY — Y VECTOR OF POINT AFTER ROTATION AND ADDITION
X — X VECTOR BEFORE ROTATION AND ADDITION
Y — Y VECTOR BEFORE ROTATION AND ADDITION

SUBROUTINE XROTAX ROTATES A REAL VECTOR BY AN AZIMUTH, ADDS AN INTEGER COORDINATE, AND RETURNS A REAL COORDINATE
(CALLED FROM FNDSDR DRWUTA DRWARR)
(CALLS XROTX)

IAX — X COORDINATE TO BE ADDED
IAY — Y COORDINATE TO BE ADDED
IAZIM — AZIMUTH OF ROTATION
RX — X VECTOR AFTER ROTATION AND ADDITION
RY — Y VECTOR AFTER ROTATION AND ADDITION
X — X VECTOR BEFORE ROTATION AND ADDITION
Y — Y VECTOR BEFORE ROTATION AND ADDITION

SUBROUTINE XROTI ROTATES A REAL VECTOR BY AN AZIMUTH AND RETURNS AN INTEGER VECTOR
(CALLED FROM ADDPTH XROTAI)
(CALLS XROTX)

IAZIM — AZIMUTH OF ROTATION
IRX — X VECTOR AFTER ROTATION
IRY — Y VECTOR AFTER ROTATION
RX — X VECTOR AFTER ROTATION
RY — Y VECTOR AFTER ROTATION
X — X VECTOR BEFORE ROTATION
Y — Y VECTOR BEFORE ROTATION

SUBROUTINE XROTX ROTATES A REAL VECTOR BY AN AZIMUTH AND RETURNS A REAL VECTOR
(CALLED FROM DRWPTH XROTI IROTX XROTAX)

COSA — COSINE OF ANGLE OF ROTATION
IALAST — LAST VALUE OF AZIMUTH OF ROTATION
IAZIM — AZIMUTH OF ROTATION
RX — X VECTOR AFTER ROTATION
RY — Y VECTOR AFTER ROTATION
SINA — SINE OF ANGLE OF ROTATION
X — X VECTOR BEFORE ROTATION
Y — Y VECTOR BEFORE ROTATION

FUNCTION XVAL FINDS THE X COORDINATE OF THE INTERSECTION OF TWO ARCS FOR A GIVEN YVAL COORDINATE
(CALLED FROM CATOAC)

CONVAR(44) — CONFLICT VARIABLES FOR CONCURRENT USAGE
RA — DISTANCE BETWEEN POINT OF CONFLICT AND CENTER OF FIRST ARC
RADICL — LOCATION OF X COORDINATE OF INTERSECTION WITH RESPECT TO X COORDINATE OF CENTER OF ARC
RB — DISTANCE BETWEEN POINT OF CONFLICT AND CENTER OF SECOND ARC
R1 — RADIUS OF ARC OF FIRST ARC
R2 — RADIUS OF ARC OF SECOND ARC
XVAL — X COORDINATE OF POINT OF INTERSECTION OF TWO ARCS
X1 — X COORDINATE OF CENTER OF ARC OF FIRST ARC
X2 — X COORDINATE OF CENTER OF ARC OF SECOND ARC
YVAL — Y COORDINATE OF POINT OF INTERSECTION OF TWO ARCS
Y1 — Y COORDINATE OF CENTER OF ARC OF FIRST ARC
Y2 — Y COORDINATE OF CENTER OF ARC OF SECOND ARC

SUBROUTINE ZEROP1 ZEROES OUT THE PARAMETERS FOR SECTION 1 OF THE INTERSECTION PATH (LINE 1)
(CALLED FROM STRLFT STRRGH UTURNL UTURNR LTLTGE LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT)

CALVAL(38) — CALCULATE INTERSECTION PATHS VARIABLES FOR CONCURRENT USAGE

SUBROUTINE ZEROP2 ZEROES OUT THE PARAMETERS FOR SECTION 2 OF THE INTERSECTION PATH (ARC 1)
(CALLED FROM STRSTR)

CALVAL(38) — CALCULATE INTERSECTION PATHS VARIABLES FOR CONCURRENT USAGE

SUBROUTINE ZEROP3 ZEROES OUT THE PARAMETERS FOR SECTION 3 OF THE INTERSECTION PATH (ARC 2)
(CALLED FROM STRSTR UTURNL UTURNR LTLTGE LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT)

CALVAL(38) — CALCULATE INTERSECTION PATHS VARIABLES FOR CONCURRENT USAGE

SUBROUTINE ZEROP4 ZEROES OUT THE PARAMETERS FOR SECTION 4 OF THE INTERSECTION PATH (LINE 2)
(CALLED FROM STRLFT STRSTR STRRGH UTURNL UTURNR LTLTGE LTLTLT LTGEGE LTGELT RTLTGE RTLTLT RTGEGE RTGELT)

CALVAL(38)   CALCULATE INTERSECTION PATHS VARIABLES FOR CONCURRENT USAGE

7. ALPHABETICAL LISTING OF ALL ROUTINES AND THE ROUTINES WHICH CAN CALL THEM

```
ABORTR = ADDCON  ADDPTH  CALPTH  CATOAC  CHKPTH  EXEC    FNDCON  FNDPTH
         FNDSDR  FNDXYP  ICHKA   NDXCON  SMEP
ADDAA  = CATOAC
ADDAL  = CATOLC
ADDCON = ADDAA   ADDAL   ADDLA   CLTOLC
ADDLA  = CLTOAC
ADDPTH = FNDPTH
AJAZIM = ADDPTH
APPLAR = READIN
ATAN36 = AZIM36
AZIM36 = ADDAA   ADDAL   ADDLA   BAND    CLTOLC
BAND   = FNDCON
BLKDAT =
CALPTH = FNDPTH
CATOAC = FNDCON
CATOLC = FNDCON
CHKPTH = EXEC
CLTOAC = FNDCON
CLTOLC = FNDCON
DRWAPR = INIPLT
DRWARC = DRWAPR  DRWINT  DRWPTH  DRWUTA
DRWARR = DRWINT
DRWBOX = DRWAPR
DRWINT = FNDPTH  INIPLT
DRWLIN = DRWAPR  DRWARR  DRWBOX  DRWINT  DRWPTH  DRWUTA
DRWPTH = FNDPTH
DRWUTA = DRWINT
DTAN   = LTGELT  LTLTLT  RTGELT  RTLTLT
ECHO   = ABORTR
EXEC   = GEOPRO
EXTRAC = CHKPTH  DRWAPR  DRWINT  ECHO    FNDCON  FNDSDR  FNDXYP
         NDXCON  SRTCON  WRITAL  WRITAP  WRITCO  WRITLA  WRITPA
FIND   = ADDCON  ADDPTH  APPLAR  CALPTH  CHKPTH  FNDPTH  FNDXYP
FNDCON = EXEC
FNDPTH = EXEC
FNDSDR = EXEC
FNDXYP = EXEC
GEOPRO =
HEADER = FNDSDR  FNDXYP  READIN  WRITCO  WRITPA
IAND   = CALPTH  CHKPTH  DRWINT  EXTRAC  FIND    REPACK  STORE
ICHKA  = ADDAA   ADDAL   ADDLA
ICHKL  = ADDAL   ADDLA
INIPLT = EXEC
INOT   = REPACK  STORE
IOR    = REPACK  STORE
IROTAX = DRWBOX
IROTX  = CALPTH  IROTAX
ISLCPF = EXEC
LDOWN  = FNDSDR
LSHIFT = CHKPTH  EXTRAC  FIND    GEOPRO  REPACK  STORE
LTGEGE = CALPTH
LTGELT = CALPTH
LTLTGE = CALPTH
LTLTLT = CALPTH
LTOL   = CLTOLC  FNDSDR
MAXVEL = LTGEGE  LTGELT  LTGEGE  LTLTLT  RTGEGE  RTGELT
         RTLTGE  RTLTLT  STRLFT  STRRGH  UTURNL  UTURNR
NDXCON = EXEC
READAI = READIN
READAP = READIN
READIN = EXEC
READIO = READIN
READLI = READIN
READOP = READIN
READSI = READIN
REPACK = ADDCON  ADDPTH  FNDSDR  NDXCON  READIN  SRTCON
RTGEGE = CALPTH
```

```
RTGELT = CALPTH
RTLTGE = CALPTH
RTLTLT = CALPTH
SMEP   = EXTRAC   FIND     REPACK   STORE
SRTCON = EXEC
STORE  = ADDCON   ADDPTH   APPLAR   FNDSDR   FNDXYP
STRLFT = CALPTH
STRRGH = CALPTH
STRSTR = CALPTH
UTURNL = CALPTH
UTURNR = CALPTH
WRITAL = EXEC
WRITAP = EXEC
WRITCO = EXEC
WRITLA = EXEC
WRITPA = EXEC
XROTAI = BAND     FNDXYP
XROTAX = DRWARR   DRWUTA   FNDSDR
XROTI  = ADDPTH   XROTAI
XROTX  = DRWPTH   IROTX    XROTAX   XROTI
XVAL   = CATOAC
ZEROP1 = LTGEGE   LTGELT   LTLTGE   LTLTLT   RTGEGE   RTGELT
         RTLTGE   RTLTLT   STRLFT   STRRGH   UTURNL   UTURNR
ZEROP2 = STRSTR
ZEROP3 = LTGEGE   LTGELT   LTLTGE   LTLTLT   RTGEGE   RTGELT
         RTLTGE   RTLTLT   STRSTR   UTURNL   UTURNR
ZEROP4 = LTGEGE   LTGELT   LTLTGE   LTLTLT   RTGEGE   RTGELT   RTLTGE
         RTLTLT   STRLFT   STRRGH   STRSTR   UTURNL   UTURNR
```


8. ALPHABETICAL LISTING OF ALL VARIABLES, THEIR STORAGE TYPE,
   AND THE ROUTINES IN WHICH THEY ARE USED

```
A       =          = CATOAC CATOLC CLTOAC LTLTLT MAXVEL RTLTLT
ADD     =          = DRWARC
ADDAZ   =          = DRWARC
ADX     / DATA     / CALPTH LTLTLT RTLTLT STRLFT STRRGH STRSTR UTURNL UTURNR
ADY     / DATA     / CALPTH LTGEGE LTLTGE LTLTLT RTGEGE RTLTGE RTLTLT STRLFT
                     STRRGH STRSTR UTURNL UTURNR
AI      =          = ADDCON
AJ      =          = ADDCON
AL      =          = MAXVEL
ANGLE   =          = STRLFT STRRGH
ANGLE2  =          = LTLTLT RTLTLT
ANGLE3  =          = LTLTLT RTLTLT
AP      =          = MAXVEL
AZIM    =          = ICHKA
AZIM1   =          = ADDAL  ADDLA
AZIM11  =          = ADDAA
AZIM12  =          = ADDAA
AZIM2   =          = ADDAL  ADDLA
AZIM21  =          = ADDAA
AZIM22  =          = ADDAA
AZ1     =          = CLTOLC
AZ11    =          = ADDAA  ADDAL  ADDLA
AZ12    =          = ADDAA  ADDAL  ADDLA
AZ2     =          = CLTOLC
AZ21    =          = ADDAA  ADDAL  ADDLA
AZ22    =          = ADDAA  ADDAL  ADDLA
B       =          = CATOAC CATOLC CLTOAC LTLTLT MAXVEL RTLTLT
BEARX   =          = ADDAL  ADDLA  BAND
BEARY   =          = ADDAL  ADDLA  BAND
BL      =          = MAXVEL
BP      =          = MAXVEL
BZIM    =          = ICHKA
C       =          = CATOAC CATOLC CLTOAC LTLTLT MAXVEL RTLTLT
CLOSE   =          = LTOL
COM01   =          = ABORTR
COM02   =          = ABORTR
COM03   =          = ABORTR
COM04   =          = ABORTR
COM05   =          = ABORTR
COM06   =          = ABORTR
COM07   =          = ABORTR
COM08   =          = ABORTR
COM09   =          = ABORTR
COM10   =          = ABORTR
COM11   =          = ABORTR
COM12   =          = ABORTR
COM13   =          = ABORTR
COM14   =          = ABORTR
COSA    =          = XROTX
COSJA   =          = LTLTLT RTLTLT
CP      =          = MAXVEL
CSIZEA  / PLOTTR   / ABORTR DRWAPR FNDXYP
CSIZEI  / PLOTTR   / ABORTR DRWINT FNDXYP
D       =          = DRWLIN
DA      =          = ICHKA
DAI     =          = FNDXYP
DA1     =          = ADDAL  ADDLA
DA11    =          = ADDAA
DA12    =          = ADDAA
DA2     =          = ADDAL  ADDLA
DA21    =          = ADDAA
DA22    =          = ADDAA
DEG     =          = DRWARC
DIST    =          = DRWLIN
DMIN    =          = DRWLIN
DUM     =          = FNDSDR
DW      =          = FNDXYP
```

```
DXI    =          = FNDXYP
DX1    =          = FNDSDR
DX2    =          = FNDSDR
DY     =          = LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE RTLTLT
DYI    =          = FNDXYP
DY1    =          = FNDSDR
D0P0   / RADIAN / ABORTR ATAN36 BAND   BLKDAT CATOAC DRWARR DRWUTA FNDSDR
                  FNDXYP ICHKA  LTLTLT MAXVEL RTLTLT ZEROP1 ZEROP2 ZEROP3
                  ZEROP4
D0P5   =          = DRWARR
D01    =          = ABORTR
D02    =          = ABORTR
D03    =          = ABORTR
D04    =          = ABORTR
D05    =          = ABORTR
D06    =          = ABORTR
D07    =          = ABORTR
D08    =          = ABORTR
D09    =          = ABORTR
D1P5   =          = DRWUTA
D10    =          = ABORTR
D11    =          = ABORTR
D12    =          = ABORTR
D13    =          = ABORTR
D14    =          = ABORTR
D15    =          = ABORTR
D16    =          = ABORTR
D17    =          = ABORTR
D18    =          = ABORTR
D19    =          = ABORTR
D2P0   =          = DRWUTA
D2P5   =          = DRWARR DRWUTA
D20    =          = ABORTR
D21    =          = ABORTR
D22    =          = ABORTR
D3P0   =          = DRWUTA
D3P5   =          = DRWARR
FPSMPH / RADIAN / ABORTR MAXVEL READAP READIN
I      =          = ABORTR ECHO   EXTRAC FIND   FNDSDR FNDXYP GEOPRO READAI
                  READAP READLI READSI REPACK SRTCON STORE  WRITAP WRITLA
                  WRITPA
IA     / INDEX  / ADDPTH APPLAR CALPTH CHKPTH DRWAPR FNDPTH FNDSDR FNDXYP
                  READAP WRITAP
IAAZIM / APPRO  / DRWBOX DRWINT DRWUTA FNDSDR FNDXYP READAP WRITAP
IADD   =          = DRWARC WRITCO
IAL    =          = DRWINT
IALAST =          = XROTX
IALEFT / APPRO  / ABORTR APPLAR ECHO   GEOPRO READAP WRITAP
IAN    / INDEX  / ABORTR APPLAR CHKPTH DRWAPR FNDPTH FNDSDR FNDXYP READAP
                  READIO WRITAP
IANGLE =          = ADDCON CALPTH DRWARR
IANP1  =          = READIO
IAPX   / APPRO  / DRWBOX FNDSDR FNDXYP READAP WRITAP
IAPY   / APPRO  / DRWBOX FNDSDR FNDXYP READAP WRITAP
IAR    =          = DRWINT
IARC   =          = DRWAPR DRWINT WRITAL
IARCAZ / ARC    / DRWAPR DRWINT READAI WRITAL
IARCN  =          = DRWAPR DRWINT WRITAL
IARCR  / ARC    / DRWAPR DRWINT READAI WRITAL
IARCSW / ARC    / DRWAPR DRWINT ECHO   READAI WRITAL
IARCX  / ARC    / ABORTR DRWAPR DRWINT ECHO   READAI WRITAL
IARCY  / ARC    / DRWAPR DRWINT READAI WRITAL
IARGHT / APPRO  / APPLAR WRITAP
IAS    =          = DRWINT
IAT    =          = ABORTR EXTRAC FIND   GEOPRO REPACK STORE
IAT1   =          = BLKDAT
IAT2   =          = BLKDAT
IAX    =          = IROTAX XROTAI XROTAX
IAY    =          = IROTAX XROTAI XROTAX
IAZARC =          = DRWARC
IAZIM  =          = FNDSDR IROTAX IROTX  XROTAI XROTAX XROTI  XROTX
```

```
IAZ1   =          = BAND
IAZ2   =          = BAND
IB     =          = BAND
IBA    / PATH   / ADDAA  ADDLA  ADDPTH DRWPTH EXTRAC FIND   FNDCON REPACK
                  STORE  WRITPA
IBAL   =          = AJAZIM
IBAND  =          = ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC FNDCON
IBLN   / LANE   / READAP WRITLA
IBUF   =          = EXEC   INIPLT
IC     =          = ABORTR
ICANSE / SDR    / ABORTR ECHO   FNDSDR WRITLA
ICHS   =          = ABORTR
ICLOSE / GEOVAL / ADDAA  ADDAL  ADDCON ADDLA  FNDCON READOP
ICON   =          = ADDCON NDXCON SRTCON WRITCO
ICONA  / CONFLT / ADDCON WRITCO
ICONAN / CONFLT / ADDCON WRITCO
ICOND  / CONFLT / ADDCON SRTCON WRITCO
ICONI  / CONFLT / ADDCON NDXCON WRITCO
ICONP  / CONFLT / ABORTR ADDCON ECHO   NDXCON SRTCON WRITCO
ICX    =          = DRWARR DRWUTA
ICY    =          = DRWARR DRWUTA
ID     =          = EXTRAC REPACK
IDA    / PATH   / ADDAA  ADDLA  ADDPTH DRWPTH ECHO   FNDCON WRITPA
IDAL   =          = AJAZIM
IDIST  =          = BAND   SRTCON
IDX    / LANE   / FNDXYP WRITLA
IE     =          = FIND   REPACK SMEP   STORE
IEA    =          = EXTRAC REPACK
IEN    =          = ABORTR BLKDAT EXTRAC FIND   REPACK STORE
IENT1  =          = ECHO   READAP
IENT2  =          = ECHO   READAI
IENT3  =          = ECHO
IENT4  =          = ECHO   READAP
IENT5  =          = ECHO   READLI
IENT6  =          = ECHO   FNDPTH
IENT7  =          = ECHO
IERROR =          = SMEP
IFET   =          = EXEC
IFLAG  / DATA   / CALPTH CATOAC FNDPTH LTGELT LTLTLT MAXVEL RTGELT RTLTLT
                  XVAL
IFS    =          = ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC FNDCON
IFW    =          = EXTRAC FIND   REPACK STORE
IGEOCP / PATH   / ABORTR ECHO   FNDPTH NDXCON SRTCON WRITPA
IIA    / PATH   / ADDAA  ADDAL  ADDLA  ADDPTH CLTOLC FNDCON WRITPA
IIAT   =          = EXTRAC FIND   REPACK STORE
IIEN   =          = EXTRAC FIND   REPACK STORE
IIL    / PATH   / ADDPTH WRITPA
IISIGN =          = XVAL
IL     / INDEX  / ADDPTH CALPTH CHKPTH DRWAPR FNDPTH FNDSDR FNDXYP READAP
ILANE  =          = DRWARR DRWUTA WRITLA
ILCH   / PATH   / ADDPTH FNDCON WRITPA
ILINE  =          = DRWAPR DRWINT WRITAL
ILINEN =          = DRWAPR DRWINT WRITAL
ILN    / INDEX  / ADDPTH CALPTH CHKPTH DRWAPR FNDPTH FNDSDR FNDXYP READAP
                  WRITAP
ILNI   =          = CALPTH
ILNO   =          = CALPTH
ILR    =          = BAND
ILT    =          = READAP
ILW    =          = REPACK
ILX1   / LINE   / ABORTR DRWAPR DRWINT ECHO   READLI WRITAL
ILX2   / LINE   / DRWAPR DRWINT READLI WRITAL
ILY1   / LINE   / DRWAPR DRWINT READLI WRITAL
ILY2   / LINE   / DRWAPR DRWINT READLI WRITAL
IL1    =          = ADDAA  ADDAL  ADDLA  CLTOLC DRWBOX
IL2    =          = ADDAA  ADDAL  ADDLA  CLTOLC DRWBOX
IMAXAZ =          = APPLAR
IMAXL  =          = FNDSDR
IMINAZ =          = APPLAR
IN     =          = EXTRAC FIND   REPACK SMEP   STORE
INA    =          = ADDCON
```

```
INDEX  =         = FNDSDR
INL    =         = ADDCON
INP    =         = ADDCON
IOA    / PATH    / ADDPTH FNDCON WRITPA
IOL    / PATH    / ADDPTH WRITPA
IOPT   / PATH    / ADDPTH WRITPA
IPAPER / GEOVAL  / DRWAPR DRWINT FNDXYP INIPLT READOP
IPATH  / GEOVAL  / FNDPTH READOP
IPEN   =         = DRWARC
IPINT  =         = CHKPTH
IPLOT  / GEOVAL  / ABORTR DRWAPR DRWARC DRWINT DRWLIN EXEC   FNDPTH FNDXYP
                   INIPLT READOP
IPN    =         = SRTCON
IPTH   =         = NDXCON SRTCON
IPTURN / PATH    / ADDPTH CHKPTH FNDCON WRITPA
IR     =         = FIND   REPACK SMEP   STORE
IRA    / PATH    / ADDAA  ADDLA  ADDPTH CATOAC CLTOAC DRWPTH FNDCON WRITPA
IRARC  =         = DRWARC
IRECAD =         = ABORTR
IRET   =         = EXEC
IRT    =         = READAP
IRX    =         = XROTAI XROTI
IRY    =         = XROTAI XROTI
IS     =         = EXTRAC FIND   GEOPRO REPACK STORE
ISAME  / GEOVAL  / FNDPTH READOP
ISCALE =         = DRWAPR DRWINT
ISDR   =         = WRITAP
ISDRA  / APPRO   / WRITAP
ISDRC  =         = DRWAPR DRWINT FNDSDR
ISDRCN =         = DRWAPR DRWINT FNDSDR
ISDRN  / APPRO   / WRITAP
ISDRS  =         = WRITLA
ISEE   =         = FNDSDR
ISLIM  / APPRO   / READAP WRITAP
ISNA   / LANE    / READAP WRITLA
ISNAME =         = EXTRAC FIND   REPACK SMEP   STORE
IST    =         = READAP
ISTART =         = FNDSDR
ISTOP  =         = FNDSDR
ISWARC =         = DRWARC
IT     =         = REPACK STORE
ITEMP  =         = SRTCON
ITEST  =         = CHKPTH CLTOLC FNDSDR READAP
ITEST1 =         = ADDAA  ADDAL  ADDLA
ITEST2 =         = ADDAA  ADDAL  ADDLA
ITITLE / TITLE   / DRWAPR DRWINT HEADER READIN WRITAL
ITURN  =         = CALPTH
ITURNC =         = CHKPTH
IUSED  =         = READAI READAP READLI READSI
IUT    =         = READAP
IV     =         = FIND   REPACK SMEP   STORE
IWD    =         = EXTRAC FIND   REPACK STORE
IX     =         = FNDXYP IROTAX IROTX  REPACK STORE
IXA    / PATH    / ADDAA  ADDLA  ADDPTH CATOAC CLTOAC DRWPTH ECHO   FNDCON
                   WRITPA
IXAPP  / GEOVAL  / CALPTH CHKPTH FNDXYP
IXARC  =         = DRWARC
IXCLAP =         = FNDSDR
IXL    / PATH    / ADDAL  ADDPTH CATOLC CLTOLC FNDCON WRITPA
IXSDRC / SDRC    / ABORTR DRWAPR DRWINT FNDSDR READSI
IX1    =         = DRWAPR DRWBOX DRWINT ICHKL
IX2    =         = DRWAPR DRWBOX DRWINT ICHKL
IY     =         = EXTRAC FIND   FNDXYP IROTAX IROTX  REPACK SMEP   STORE
IYA    / PATH    / ADDAA  ADDLA  ADDPTH CATOAC CLTOAC DRWPTH ECHO   FNDCON
                   WRITPA
IYAPP  / GEOVAL  / CALPTH CHKPTH FNDXYP
IYARC  =         = DRWARC
IYCLAP =         = FNDSDR
IYES   =         = READAP
IYL    / PATH    / ADDAL  ADDPTH CATOLC CLTOLC FNDCON WRITPA
IYSDRC / SDRC    / DRWAPR DRWINT FNDSDR READSI
```

```
IY1    =         = ICHKL
IY2    =         = ICHKL
IZ     =         = FNDCON FNDPTH READAI READAP READLI READSI
I12    =         = NDXCON
J      =         = ABORTR ECHO   READAI READLI READSI SRTCON WRITPA
JA     / INDEX   / ADDPTH APPLAR CALPTH FNDPTH FNDSDR
JAAZIM =         = APPLAR
JAN    / INDEX   / APPLAR FNDPTH FNDSDR READIO
JANGLE / DATA    / CALPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                   RTLTLT STRLFT STRRGH UTURNL UTURNR
JAZIM  / DATA    / ADDPTH CALPTH DRWPTH FNDPTH
JAZIML =         = AJAZIM
JBLN   =         = READAP
JB2    / DATA    / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                   RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
JB2OR3 =         = AJAZIM
JB3    / DATA    / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
JCLOSE =         = FNDCON
JCON   =         = NDXCON SRTCON
JD2    / DATA    / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                   RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
JD2OR3 =         = AJAZIM
JD3    / DATA    / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
JFS    =         = ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC
JGEOCP =         = FNDCON
JL     / INDEX   / ADDPTH CALPTH FNDPTH FNDSDR
JLCH   / DATA    / ADDPTH CALPTH
JLN    / INDEX   / ADDPTH CALPTH FNDPTH FNDSDR
JMAXL  =         = FNDSDR
JNA    =         = ADDCON
JNL    =         = ADDCON
JNP    =         = ADDCON
JOPT   / DATA    / ADDPTH CALPTH FNDPTH
JPATH  =         = READOP
JPINT  =         = CHKPTH
JPLOT  =         = READOP
JRECAD =         = ABORTR
JSAME  =         = READOP
JSCALE =         = DRWAPR DRWINT
JSLIM  =         = ADDPTH
JSPEED / DATA    / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                   RTLTLT STRLFT STRRGH STRSTR UTURNL UTURNR
JTEST  =         = CHKPTH
JTEST1 =         = ADDAA  ADDAL  ADDLA
JTEST2 =         = ADDAA  ADDAL  ADDLA
JTITLE =         = DRWAPR DRWINT
JTURN  =         = CALPTH
JXL    / PATH    / ADDAL  ADDPTH CATOLC CLTOLC FNDCON WRITPA
JYL    / PATH    / ADDAL  ADDPTH CATOLC CLTOLC FNDCON WRITPA
K      =         = ECHO
KA     =         = DRWINT
KAAZIM =         = APPLAR
KAN    =         = DRWINT
KANGLE =         = CALPTH LTGELT LTLTLT RTGELT RTLTLT
KANGL2 =         = LTLTLT RTLTLT
KANGL3 =         = LTLTLT RTLTLT
KAZIM  / DATA    / CALPTH FNDPTH
KL     =         = DRWINT
KLN    =         = DRWINT
KOUNT  =         = DRWINT INIPLT
KP     =         = ADDCON
KSLIM  =         = ADDPTH
KTURN  / DATA    / ADDPTH CALPTH
LAAZIM =         = APPLAR
LARCS  / GEOPRO  / DRWAPR DRWINT ECHO   READAI WRITAL
LAZIM  =         = CALPTH
LA1    / PATH    / ADDAA  ADDAL  ADDLA  ADDPTH CLTOLC DRWPTH FNDCON WRITPA
LA2    / PATH    / ADDAL  ADDPTH CLTOLC DRWPTH FNDCON WRITPA
LBA    =         = APPLAR
LENP   / PATH    / ADDPTH WRITPA
LFALSE =         = BLKDAT
```

```
LGEOM  / LANE   / DRWAPR DRWINT FNDSDR READAP WRITLA
LGEOM1 =        = FNDXYP READAP
LGEOM2 =        = FNDXYP
LGEOM3 =        = FNDXYP
LGEOM4 =        = FNDXYP
LIBA   / GEOPRO / CHKPTH DRWAPR DRWINT ECHO   FNDPTH FNDSDR FNDXYP READAP
                  READIN READIO WRITAP
LIBL   / PATH   / ADDPTH FNDCON WRITPA
LIMP   / PATH   / ADDPTH WRITPA
LINES  / OUTPUT / BLKDAT FNDSDR FNDXYP READAI READAP READIO READLI READOP
                  READSI WRITCO WRITPA
LINTP  / LANE   / CHKPTH WRITLA
LLANES / APPRO  / CHKPTH DRWAPR DRWINT FNDSDR FNDXYP READAP WRITAP
LLINES / GEOPRO / DRWAPR DRWINT ECHO   READLI WRITAL
LLTYPE =        = READAP
LL1    / PATH   / ADDAA  ADDAL  ADDLA  ADDPTH CLTOLC DRWPTH FNDCON WRITPA
LL2    / PATH   / ADDPTH DRWPTH FNDCON WRITPA
LN     =        = CALPTH
LNI    =        = CALPTH
LNJ    =        = CALPTH
LNN    =        = CALPTH
LOBA   / GEOPRO / DRWAPR DRWINT ECHO   FNDPTH FNDXYP READAP READIN READIO
                  WRITAP
LOBL   / PATH   / ADDPTH FNDCON WRITPA
LP     =        = ADDCON
LSDRC  / SDRC   / DRWAPR DRWINT FNDSDR READSI
LTDIRX / PLOTTR / DRWARR DRWUTA FNDXYP
LTDIRY / PLOTTR / DRWARR DRWUTA FNDXYP
LTEST  =        = DRWLIN READAI READAP READLI READSI WRITPA
LTRUE  =        = BLKDAT
LTURN  / LANE   / CHKPTH DRWINT READAP WRITLA
LTYPE  / LANE   / READAP WRITLA
LWID   / LANE   / ABORTR DRWAPR DRWINT ECHO   FNDSDR FNDXYP READAP WRITLA
L1     / DATA   / ADDPTH LTGEGE LTLTGE RTGEGE RTLTGE STRSTR UTURNL UTURNR
                  ZEROP1
L2     / DATA   / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                  RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
L2OR3  =        = AJAZIM
L3     / DATA   / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
L4     / DATA   / ADDPTH LTGELT LTLTLT RTGELT RTLTLT UTURNL UTURNR ZEROP4
MAL    / GEOCP  / ADDAA  ADDAL  ADDLA  BAND   CLTOLC FNDCON
MAXSEE =        = FNDSDR
MAXXA  / PLOTTR / BLKDAT DRWAPR FNDXYP
MAXXI  / PLOTTR / BLKDAT DRWINT FNDXYP
MAXYA  / PLOTTR / BLKDAT DRWAPR FNDXYP
MAXYI  / PLOTTR / BLKDAT DRWINT FNDXYP
MAZIM  =        = CALPTH
MBA    / GEOCP  / ADDAA  ADDAL  BAND   FNDCON
MDA    / GEOCP  / ADDAA  ADDAL  BAND   FNDCON
MGEOCP =        = ADDCON
MIA    / GEOCP  / ADDAA  ADDAL  ADDLA  CLTOLC FNDCON
MIBL   =        = FNDCON
MINXA  / PLOTTR / BLKDAT DRWAPR FNDXYP
MINXI  / PLOTTR / BLKDAT DRWINT FNDXYP
MINYA  / PLOTTR / BLKDAT DRWAPR FNDXYP
MINYI  / PLOTTR / BLKDAT DRWINT FNDXYP
MLCH   =        = FNDCON
MLL    / GEOCP  / ADDAA  ADDAL  ADDLA  BAND   CLTOLC FNDCON
MOA    =        = FNDCON
MOBL   =        = FNDCON
MODELT / OUTPUT / BLKDAT EXEC   WRITAL WRITAP WRITCO WRITLA WRITPA
MPTH   / GEOCP  / ADDAA  ADDAL  ADDLA  CLTOLC FNDCON
MPTHP1 =        = FNDCON
MPTURN =        = FNDCON
MRA    / GEOCP  / ADDAA  ADDAL  BAND   CATOAC CATOLC FNDCON
MSG    =        = ABORTR EXEC
MSGERR =        = EXEC
MSGPP  =        = ABORTR
MSG901 =        = FNDXYP
MSG902 =        = FNDXYP
MSG903 =        = FNDSDR
MSG904 =        = FNDSDR
MSG905 =        = FNDSDR
MSG906 =        = FNDPTH
MSG907 =        = CALPTH
MSG908 =        = CALPTH
MSG909 =        = ADDPTH
MSG910 =        = ADDPTH
MSG911 =        = CHKPTH
MSG912 =        = CHKPTH
MSG913 =        = FNDCON
MSG914 =        = ADDCON
MSG915 =        = ADDCON
MSG916 =        = ICHKA
MSG917 =        = CATOAC
MSG918 =        = NDXCON
MTURN  =        = CALPTH
MXA    / GEOCP  / ADDAA  ADDAL  BAND   CATOAC CATOLC FNDCON
MXL    / GEOCP  / ADDLA  BAND   CLTOAC CLTOLC FNDCON
MYA    / GEOCP  / ADDAA  ADDAL  BAND   CATOAC CATOLC FNDCON
MYL    / GEOCP  / ADDLA  BAND   CLTOAC CLTOLC FNDCON
NAP    / GEOPRO / ECHO   READAP READIO WRITAP
NARCS  / GEOPRO / DRWAPR DRWINT ECHO   READAI WRITAL
NBA    =        = APPLAR ICHKA
NBITS  =        = EXTRAC FIND
NBLANK =        = READAP READOP
NC     =        = ADDAA  ADDAL  ADDCON ADDLA  CATOAC CATOLC CLTOAC CLTOLC
                  FNDCON
NCHS   =        = ABORTR
NCOM01 =        = ABORTR
NCOM02 =        = ABORTR
NCOM03 =        = ABORTR
NCOM04 =        = ABORTR
NCOM05 =        = ABORTR
NCOM06 =        = ABORTR
NCOM07 =        = ABORTR
NCOM08 =        = ABORTR
NCOM09 =        = ABORTR
NCOM10 =        = ABORTR
NCOM11 =        = ABORTR
NCOM12 =        = ABORTR
NCOM13 =        = ABORTR
NCOM14 =        = ABORTR
NCONFS / GEOPRO / ADDCON BLKDAT ECHO   FNDCON NDXCON WRITCO
NDA    =        = ICHKA
NDEGST / APPRO  / CALPTH READAP
NDEGUT / APPRO  / CALPTH READAP
NEXTL  =        = READAP
NGEOCP / PATH   / NDXCON SRTCON WRITPA
NIBA   / GEOPRO / ABORTR CHKPTH DRWAPR DRWINT ECHO   FNDPTH FNDSDR FNDXYP
                  READAP READIN READIO WRITAP
NIBL   / GEOPRO / BLKDAT ECHO   READAP WRITLA
NL     =        = READAP
NLANEI / INDEX  / CALPTH FNDPTH
NLANEJ / INDEX  / CALPTH FNDPTH
NLANES / APPRO  / CHKPTH DRWAPR DRWINT FNDSDR FNDXYP READAP WRITAP
NLEFTD =        = DRWAPR DRWINT
NLINE  / OUTPUT / FNDSDR FNDXYP HEADER READAI READAP READIO READLI READOP
                  READSI WRITCO WRITPA
NLINES / GEOPRO / DRWAPR DRWINT ECHO   READLI WRITAL
NLL    / LANE   / READAP WRITLA
NLR    / LANE   / READAP WRITLA
NNOPLT =        = READOP
NOATTB / NOATTB / ABORTR BLKDAT ECHO   FNDPTH READAI READAP READLI
NOBA   / GEOPRO / DRWAPR DRWINT ECHO   FNDPTH FNDXYP READAP READIN READIO
                  WRITAP
NOBL   / GEOPRO / BLKDAT ECHO   READAP WRITLA
NOPT1  =        = READOP
NPAGE  / OUTPUT / ABORTR BLKDAT HEADER
NPATHS / GEOPRO / ADDPTH BLKDAT ECHO   FNDCON FNDPTH SRTCON WRITPA
NPINT  / LANE   / ADDPTH CHKPTH WRITLA
NPLT   =        = READOP
```

```
NPLTI  =          = READOP
NPM1   =          = FNDCON
NPRIM  =          = READOP
NPTH   / GEOCP  / ADDAA  ADDAL  ADDLA  CLTOLC FNDCON
NR     =          = READAP
NRECAD =          = EXEC
NS     =          = READAP
NSAME  =          = READOP
NSCALE =          = FNDXYP
NSDR   / APPRO  / WRITAP
NSDRAP =          = FNDSDR
NSDRC  / SDRC   / DRWAPR DRWINT FNDSDR READSI
NSDRS  / GEOPRO / BLKDAT ECHO   FNDSDR WRITLA
NSEPAR =          = READOP
NTABL  / OUTPUT / BLKDAT FNDSDR READAI READAP READIO READLI READOP READSI
                  WRITCO WRITPA
NTEST  =          = READIO
NU     =          = READAP
NUM    =          = ABORTR ADDAA  ADDAL  ADDLA  ECHO   FNDPTH READAI READAP
                    READLI
NUMLAN =          = ECHO   WRITLA
NUMPTS =          = ADDAA  ADDAL  ADDLA
NWDS   =          = ABORTR
NWE    =          = EXTRAC FIND   REPACK STORE
NXL    / GEOCP  / ADDLA  BAND   CLTOAC CLTOLC FNDCON
NYES   =          = READAP
NYL    / GEOCP  / ADDLA  BAND   CLTOAC CLTOLC FNDCON
PI     / RADIAN / ABORTR ATAN36 READIN
PHID   =          = FNDXYP
R      =          = MAXVEL READOP
RA     =          = XVAL
RADIAN / RADIAN / ABORTR ADDAA  ADDAL  ADDLA  AZIM36 CALPTH DRWARC LTGEGE
                  LTGELT LTLTGE LTLTLT READIN RTGEGE RTGELT RTLTGE RTLTLT
                  STRLFT STRRGH UTURNL UTURNR XROTX
RADICL =          = CATOAC CATOLC CLTOAC LTLTLT RTLTLT XVAL
RADIUS / GEOVAL / ABORTR CALPTH READOP
RA2    / DATA   / ADDPTH CALPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT
                  RTLTGE RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
RA3    / DATA   / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
RB     =          = XVAL
RC     / DATA   / CALPTH LTGEGE LTLTGE RTGEGE RTLTGE
RX     =          = IROTAX IROTX  XROTAX XROTI  XROTX
RY     =          = IROTAX IROTX  XROTAX XROTI  XROTX
R1     =          = CATOAC XVAL
R1SQ   =          = CATOAC
R2     =          = CATOAC XVAL
R2SQ   =          = CATOAC
SA     =          = FNDXYP READOP
SCALE  / PLOTTR / ABORTR DRWAPR DRWARC DRWINT DRWLIN
SCALEA / GEOVAL / ABORTR DRWAPR FNDXYP READOP
SCALEF =          = FNDXYP
SCALEI / GEOVAL / ABORTR DRWINT FNDXYP READOP
SI     =          = FNDXYP READOP
SINA   =          = XROTX
SINJA  =          = LTLTLT RTLTLT
UX1    =          = DRWUTA
UX2    =          = DRWUTA
UX3    =          = DRWUTA
UX4    =          = DRWUTA
UX5    =          = DRWUTA
UX6    =          = DRWUTA
UY1    =          = DRWUTA
UY2    =          = DRWUTA
UY3    =          = DRWUTA
UY4    =          = DRWUTA
UY5    =          = DRWUTA
UY6    =          = DRWUTA
VAL    =          = DTAN
VELMPH =          = MAXVEL
X      =          = ADDAA  ADDAL  ADDLA  ATAN36 AZIM36 CATOLC CLTOAC DRWAPR
                    DRWARC DRWINT IROTX  XROTAI XROTAX XROTI  XROTX
XB     =          = CATOLC CLTOAC
XBA    =          = LDOWN  LTOL
XBB    =          = LDOWN  LTOL
XBEAR  =          = ADDAL  ADDLA
XBEAR1 =          = ADDAA
XBEAR2 =          = ADDAA
XBIG   =          = FNDSDR
XBOT   =          = DRWARR
XBRDR  =          = DRWAPR DRWINT
XC2    / DATA   / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                  RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
XC3    / DATA   / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
XDMIN  =          = DRWLIN
XFROM  =          = FNDSDR
XI     / DATA   / CALPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                  RTLTLT STRLFT STRRGH STRSTR UTURNL UTURNR
XINT   =          = DRWLIN FNDSDR ICHKL  LDOWN
XINT1  / GEOCP  / ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC
XINT2  / GEOCP  / ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC
XI1    =          = LTOL
XI2    =          = LTOL
XLEFT  =          = DRWARR
XM     =          = CATOLC CLTOAC
XMA    =          = LDOWN  LTOL
XMAX   / PLOTTR / ABORTR DRWAPR DRWARC DRWINT DRWLIN
XMB    =          = LDOWN  LTOL
XMIN   / PLOTTR / ABORTR DRWAPR DRWARC DRWINT DRWLIN
XO     / DATA   / CALPTH LTGELT LTLTLT RTGELT RTLTLT STRLFT STRRGH STRSTR
                  UTURNL UTURNR
XPAGE  =          = DRWAPR DRWARC DRWINT DRWLIN
XRGHT  =          = DRWARR
XROUND / RADIAN / ABORTR ADDAA  ADDAL  ADDCON ADDLA  ADDPTH BAND   BLKDAT
                  CLTOLC DRWARC FNDXYP ICHKA  LDOWN  LTGEGE LTGELT LTLTGE
                  LTLTLT MAXVEL READAP RTGEGE RTGELT RTLTGE RTLTLT STRLFT
                  STRRGH STRSTR UTURNL UTURNR XROTI
XSDR   =          = FNDSDR LDOWN
XSIZEA / PLOTTR / ABORTR DRWAPR FNDXYP
XSIZEI / PLOTTR / ABORTR DRWINT FNDXYP
XTOP   =          = DRWARR
XX1    =          = DRWLIN
XX2    =          = DRWLIN
X0     / PLOTTR / ABORTR DRWAPR DRWINT
X1     =          = CATOAC CLTOLC DRWAPR DRWBOX DRWINT DRWLIN DRWPTH FNDSDR
                    LDOWN  LTOL   XVAL
X11    / DATA   / ADDPTH DRWPTH LTGEGE LTLTGE RTGEGE RTLTGE STRSTR UTURNL
                  UTURNR ZEROP1
X12    / DATA   / ADDPTH DRWPTH LTGEGE LTLTGE RTGEGE RTLTGE STRSTR UTURNL
                  UTURNR ZEROP1
X2     =          = CATOAC CLTOLC DRWAPR DRWBOX DRWINT DRWLIN DRWPTH FNDSDR
                    LDOWN  LTOL   XVAL
X2X1SQ =          = CATOAC
X3     =          = CLTOLC DRWBOX FNDSDR LDOWN  LTOL
X4     =          = CLTOLC DRWBOX FNDSDR LTOL
X41    / DATA   / ADDPTH DRWPTH LTGELT LTLTLT RTGELT RTLTLT UTURNL UTURNR
                  ZEROP4
X42    / DATA   / ADDPTH DRWPTH LTGELT LTLTLT RTGELT RTLTLT UTURNL UTURNR
                  ZEROP4
Y      =          = ATAN36 AZIM36 DRWAPR DRWARC DRWINT IROTX  XROTAI XROTAX
                    XROTI  XROTX
YBEAR  =          = ADDAL  ADDLA
YBEAR1 =          = ADDAA
YBEAR2 =          = ADDAA
YBOT   =          = DRWARR
YBRDR  =          = DRWAPR DRWINT
YC     / DATA   / CALPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                  RTLTLT
YC2    / DATA   / ADDPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                  RTLTLT STRLFT STRRGH UTURNL UTURNR ZEROP2
YC3    / DATA   / ADDPTH LTLTLT RTLTLT STRLFT STRRGH ZEROP3
YDMIN  =          = DRWLIN
YFROM  =          = FNDSDR
```

```
YI       / DATA    / CALPTH LTGEGE LTGELT LTLTGE LTLTLT RTGEGE RTGELT RTLTGE
                     RTLTLT STRLFT STRRGH STRSTR UTURNL UTURNR
YINT     =         = DRWLIN FNDSDR ICHKL  LDOWN
YINT1    / GEOCP   / ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC
YINT2    / GEOCP   / ADDAA  ADDAL  ADDLA  CATOAC CATOLC CLTOAC CLTOLC
YI1      =         = LTOL
YI2      =         = LTOL
YLEFT    =         = DRWARR
YMAX     / PLOTTR  / ABORTR DRWAPR DRWARC DRWINT DRWLIN
YMIN     / PLOTTR  / ABORTR DRWAPR DRWARC DRWINT DRWLIN
YO       / DATA    / CALPTH LTGELT LTLTLT RTGELT RTLTLT STRLFT STRRGH STRSTR
                     UTURNL UTURNR
YPAGE    =         = DRWAPR DRWARC DRWINT DRWLIN
YRGHT    =         = DRWARR
YSDR     =         = FNDSDR LDOWN
YSIZEA   / PLOTTR  / ABORTR DRWAPR FNDXYP
YSIZEI   / PLOTTR  / ABORTR DRWINT FNDXYP
YTOP     =         = DRWARR
YVAL     =         = XVAL
YY1      =         = DRWLIN
YY2      =         = DRWLIN
Y0       / PLOTTR  / ABORTR DRWAPR DRWINT
Y1       =         = CATOAC CLTOLC DRWAPR DRWBOX DRWINT DRWLIN DRWPTH FNDSDR
                     LDOWN  LTOL   XVAL
Y1SQ     =         = CATOAC
Y11      / DATA    / ADDPTH DRWPTH LTGEGE LTLTGE RTGEGE RTLTGE STRSTR UTURNL
                     UTURNR ZEROP1
Y12      / DATA    / ADDPTH DRWPTH LTGEGE LTLTGE RTGEGE RTLTGE STRSTR UTURNL
                     UTURNR ZEROP1
Y2       =         = CATOAC CLTOLC DRWAPR DRWBOX DRWINT DRWLIN DRWPTH FNDSDR
                     LDOWN  LTOL   XVAL
Y2SQ     =         = CATOAC
Y2Y1SQ   =         = CATOAC
Y3       =         = CLTOLC DRWBOX FNDSDR LDOWN  LTOL
Y4       =         = CLTOLC DRWBOX FNDSDR LTOL
Y41      / DATA    / ADDPTH DRWPTH LTGELT LTLTLT RTGELT RTLTLT UTURNL UTURNR
                     ZEROP4
Y42      / DATA    / ADDPTH DRWPTH LTGELT LTLTLT RTGELT RTLTLT UTURNL UTURNR
                     ZEROP4
ZERO     / RADIAN  / ABORTR BLKDAT CALPTH CATOAC CATOLC CLTOAC ICHKL  LDOWN
                     LTOL   XVAL
```

9. GENERALIZED CALLING SEQUENCE DIAGRAM

```
GEDPRO
  ↓
EXEC
  ↓
ISLCPF
READIN
 #HEADER
 #READIU
   >HEADER
 #READAP
   >HEADER
 #APPLAR
 #READAI
   >HEADER
 #READLI
   >HEADER
 #READSI
   >HEADER
 #READOP
   >HEADER
WRITAL
FNDXYP
 #HEADER
FNDSDR
 #LTOL
 #LDOWN
 #HEADER
WRITAP
INIPLT
 #DRWAPR
   >DRWBOX
      DRWLIN
       LTOL
   >DRWARC
   >DRWLIN
      LTOL
 #DRWINT
   >DRWBOX
      DRWLIN
       LTOL
   >DRWUTA
      DRWLIN
       LTOL
      DRWARC
   >DRWARR
      DRWLIN
       LTOL
   >DRWARC
   >DRWLIN
      LTOL
FNDPTH
 #DRWINT
   >DRWBOX
      DRWLIN
       LTOL
   >DRWUTA
      DRWLIN
       LTOL
      DRWARC
   >DRWARR
      DRWLIN
       LTOL
   >DRWARC
   >DRWLIN
      LTOL
 #CALPTH
   >STRLFT
      ZEROP1
      ZEROP4
      MAXVEL
   >STRSTR
      ZEROP2
      ZEROP3
      ZEROP4
   >STRRGH
      ZEROP1
      ZEROP4
      MAXVEL
   >UTURNL
      ZEROP3
      MAXVEL
      ZEROP1
      ZEROP4
   >UTURNR
      ZEROP3
      MAXVEL
      ZEROP1
      ZEROP4
   >LTLTGE
      ZEROP3
      ZEROP4
      MAXVEL
      ZEROP1
   >LTLTLT
      ZEROP3
      ZEROP1
      MAXVEL
      ZEROP4
   >LTGEGE
      ZEROP3
      ZEROP4
      MAXVEL
      ZEROP1
   >LTGELT
      ZEROP3
      ZEROP1
      MAXVEL
      ZEROP4
   >RTLTGE
      ZEROP3
      ZEROP4
      MAXVEL
      ZEROP1
   >RTLTLT
      ZEROP3
      ZEROP1
      MAXVEL
      ZEROP4
   >RTGEGE
      ZEROP3
      ZEROP4
      MAXVEL
      ZEROP1
   >RTGELT
      ZEROP3
      ZEROP1
      MAXVEL
      ZEROP4
 #ADDPTH
 #DRWPTH
   >DRWLIN
      LTOL
   >DRWARC
CHKPTH
WRITLA
FNDCON
 #BAND
 #CLTOLC
   >LTOL
   >ADDCON
 #CLTOAC
   >ADDLA
      ICHKL
      ICHKA
      ADDCON
 #CATOLC
   >ADDAL
      ICHKL
      ICHKA
      ADDCON
 #CATOAC
   >XVAL
   >ADDAA
      ICHKA
      ADDCON
SRTCON
WRITPA
 #HEADER
NDXCON
WRITCO
 #HEADER
```

APPENDIX C

ADDITIONAL INFORMATION FOR THE DRIVER-VEHICLE PROCESSOR

This page intentionally left blank to facilitate printing on 2 sides.

```
C¬     PROGRAM    DVPRO  ( INPUT=513,OUTPUT=513,TAPE9=513,TAPE5=INPUT )
C
C------DRIVER-VEHICLE PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKAGE
C
C------C¬ = CDC ONLY CODE
C------C; = IBM ONLY CODE
C
C¬     DIMENSION          MSG(6)
C¬     DATA       MSG    / 4H FAT,4HAL E,4HXECU,4HTION,4H ERR,2HOR /
       CALL  READIN
C¬     ASSIGN 101 TO NRECAD
C¬     CALL  XMIT     ( NRECAD )
       CALL  WRITDV
       CALL  BIASLT
       CALL  GENHED
       CALL  GENDV
       CALL  PNOTES
       CALL  PSUMDV
       CALL  PSTATS
       CALL  EXIT
C¬101 CONTINUE
C¬     CALL  ABORTR   ( MSG,22,71 )
       STOP
C¬102 GO TO NRECAD                                                       *DEBUG*
       END                                                               DVPRO
```

```
      BLOCK DATA
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / STATS  / SPERD(5,15),SPERL(6,6),SPERT(6,6),SPERV(15,6)
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / ZTEMPD(71)
      DATA      IDISTN / 4HCONS,3HTAN,4HERLA,3HNG ,4HGAMM,3HA   ,
     *                  4HLOGN,3HRML,4HNEGE,3HXP ,4HSNEG,3HEXP,
     *                  4HUNIF,3HORM /
      DATA      IEOF   / .FALSE. /
      DATA      IFORM  / 4H0APP,4HROAC,4HH NU,4HMBER /
      DATA      IYESD  / 3HNO  /
      DATA      IYESP  / 3HNO  /
      DATA      IYESV  / 3HNO  /
      DATA      IYESDL / 05*3HNO  /
      DATA      IYESVL / 15*3HNO  /
      DATA      LINES  / 62 /
      DATA      MAXV   / 161 /
      DATA      MODELT / 9 /
      DATA      NBLANK / 4H     /
      DATA      NLINE  / 0 /
      DATA      NNO    / 3HNO  /
      DATA      NOTE   / 14*0 /
      DATA      NPAGE  / 1 /
      DATA      NTABL  / 1 /
      DATA      NVA    / 6*0 /
      DATA      NYES   / 3HYES /
      DATA      QTLAST / 36*-5.0 /
      DATA      SPERD  / 75*0.0 /
      DATA      SPERL  / 36*0.0 /
      DATA      SPERT  / 36*0.0 /
      DATA      SPERV  / 90*0.0 /
      DATA      ZERO   / 0.000001 /
C------PERCENT OF VEHICLE CLASS IN TRAFFIC STREAM - XPERV(IVEHCL,IAN)
      DATA      XPERV  / 20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0,
     *                  20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0,
     *                  20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0,
     *                  20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0,
     *                  20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0,
     *                  20.,32.,30.,15.,0.5,0.2,0.1,0.2,0.5,1.5, 5*0.0/
C------PERCENT OF DRIVER CLASS IN VEHICLE CLASS - XPERD(IDRICL,IVEHCL)
      DATA      XPERD  / 30.,40.,30., 0., 0.,    35.,35.,30., 0., 0.,
     *                  20.,40.,40., 0., 0.,    25.,50.,25., 0., 0.,
     *                  40.,30.,30., 0., 0.,    50.,40.,10., 0., 0.,
     *                  50.,40.,10., 0., 0.,    20.,30.,50., 0., 0.,
     *                  25.,50.,25., 0., 0.,    50.,40.,10., 27*0.0/
C------VEHICLE CHARACTERISTICS
      DATA      IAMAX  /   8,  9, 11,  8,  8,  7,  6,  6,  5, 14, 5*0 /
      DATA      IDMAX  /   8, 11, 11,  8, 11, 11, 11,  8, 11, 12, 5*0 /
      DATA      IRMIN  /  20, 22, 24, 28, 42, 40, 45, 28, 28, 20, 5*0 /
      DATA      IVCHAR / 100,110,110,100, 85, 80, 75, 90, 85,115, 5*0 /
      DATA      IVMAX  / 150,192,200,150,160,160,150,150,125,205, 5*0 /
      DATA      LENV   /  15, 17, 19, 25, 30, 50, 55, 25, 35, 14, 5*0 /
C------DRIVER CHARACTERISTICS
      DATA      IDCHAR /     110,     100,      85,       0,       0/
      DATA      PIJR   /    0.50,    1.00,    1.50,    0.00,    0.00/
      END                                                              BLOCK D
```

```
      SUBROUTINE READIN
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      LOGICAL           IEOF
      COMMON / TITLE  / ITITLE(20)
  501 FORMAT(20A4)
C
C-----SUBROUTINE READIN READS INPUT DATA AND CHECKS FOR ERRORS
C
      FPSMPH = 88.0/60.0
      SQRT3 = SQRT ( 3.0 )
C-----READ 80 CHARACTER TITLE FOR DVPRO
      READ 501 , ITITLE
      CALL  HEADER
C-----READ THE NUMBER AND LIST OF INBOUND AND OUTBOUND APPRAOCHES AND
C-----CHECK FOR ERRORS
      CALL  READIO
C-----READ THE NUMBER OF APPROACHES AND DRIVER-VEHICLE PROCESSOR OPTIONS
C-----AND CHECK FOR ERRORS
      CALL  READOP
C-----READ THE APPROACH INFORMATION AND CHECK FOR ERRORS
      CALL  READAP
C-----DUMMY READ GEOMETRY PROCESSOR DATA
      CALL  READGP
C-----READ YES OPTIONS
      CALL  READYO
      RETURN
      END
```

READIN

```
      SUBROUTINE HEADER
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / TITLE  / ITITLE(20)
  601 FORMAT(1H1,7X,48HDRIVER-VEHICLE PROCESSOR FOR THE TEXAS TRAFFIC S,
     *       22HIMULATION PACKAGE PAGE,I3,/)
  602 FORMAT(8X,20A4,//)
C
C-----SUBROUTINE HEADER SKIPS TO THE TOP OF A NEW PAGE, PRINTS THE
C-----HEADER MESSAGE, AND PRINTS THE TITLE FOR DVPRO
C
      PRINT 601 , NPAGE
      NLINE = 2
      NPAGE = NPAGE + 1
      PRINT 602 , ITITLE
      NLINE = NLINE + 3
      RETURN
      END
```

```
      SUBROUTINE READIO
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / XPERL(6,6),IAN,IANP1,JAN,ZTEMPD(32)
  501 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,33H  -  LISTING OF INBOUND APPROACH ,
     *       7HNUMBERS,//)
  602 FORMAT(16X,I6)
  603 FORMAT(//,12X,37HTOTAL NUMBER OF INBOUND APPROACHES = ,I2,///)
  604 FORMAT(8X,5HTABLE,I3,34H  -  LISTING OF OUTBOUND APPROACH ,
     *       7HNUMBERS,//)
  605 FORMAT(16X,I6)
  606 FORMAT(//,12X,38HTOTAL NUMBER OF OUTBOUND APPROACHES = ,I2)
  801 FORMAT(32H0NUMBER OF INBOUND APPROACHES = ,I3,16H IS LE 0 OR GT 6)
  802 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,17H IS LE 0 OR GT 12)
  803 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO INBOUND ,
     *       8HAPPROACH,I3,3H = ,I3)
  804 FORMAT(32H0NUMBER OF OUTBOUND APPROACHES =,I3,16H IS LE 0 OR GT 6)
  805 FORMAT(18H0OUTBOUND APPROACH,I3,3H = ,I3,17H IS LE 0 OR GT 12)
  806 FORMAT(18H0OUTBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO OUTBOUND,
     *       9H APPROACH,I3,3H = ,I3)
  807 FORMAT(17H0INBOUND APPROACH,I3,3H = ,I3,21H IS EQUAL TO OUTBOUND,
     *       9H APPROACH,I3,3H = ,I3)
C
C-----SUBROUTINE READIO READS THE NUMBER AND LIST OF INBOUND AND
C-----OUTBOUND APPROACHES AND CHECKS FOR ERRORS
C
C-----READ NUMBER OF INBOUND APPROACHES
      READ 501 , NIBA
                    IF ( NIBA . LE .  0 )         GO TO 8010
                    IF ( NIBA . GT .  6 )         GO TO 8010
            IF ( NLINE+NIBA+9 . GT . LINES )      CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
C-----READ LIST OF INBOUND APPROACHES
      READ 501 , (LIBA(IAN),IAN=1,NIBA)
      PRINT 602 , (LIBA(IAN),IAN=1,NIBA)
      NLINE = NLINE + NIBA
      DO 1020  IAN = 1 , NIBA
                    IF ( LIBA(IAN) . LE .  0 )    GO TO 8020
                    IF ( LIBA(IAN) . GT . 12 )    GO TO 8020
                    IF ( NIBA . EQ . 1 )          GO TO 1020
                    IF ( IAN . EQ . NIBA )        GO TO 1020
C-----CHECK IF APPROACH IS DUPLICATED ON LIST OF INBOUND APPROACHES
      IANP1 = IAN + 1
      DO 1010  JAN = IANP1 , NIBA
                    IF ( LIBA(IAN).EQ.LIBA(JAN) )GO TO 8030
 1010 CONTINUE
 1020 CONTINUE
      PRINT 603 , NIBA
      NLINE = NLINE + 6
C-----READ NUMBER OF OUTBOUND APPROACHES
      READ 501 , NOBA
                    IF ( NOBA . LE .  0 )         GO TO 8040
                    IF ( NOBA . GT .  6 )         GO TO 8040
            IF ( NLINE+NOBA+13 . GT . LINES )     CALL  HEADER
      PRINT 604 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
C-----READ LIST OF OUTBOUND APPROACHES
      READ 501 , (LOBA(IAN),IAN=1,NOBA)
      PRINT 605 , (LOBA(IAN),IAN=1,NOBA)
      NLINE = NLINE + NOBA
      DO 1040  IAN = 1 , NOBA
                    IF ( LOBA(IAN) . LE .  0 )    GO TO 8050
                    IF ( LOBA(IAN) . GT . 12 )    GO TO 8050
                    IF ( NOBA . EQ . 1 )          GO TO 1040
                    IF ( IAN . EQ . NOBA )        GO TO 1040
C-----CHECK IF APPROACH IS DUPLICATED ON LIST OF OUTBOUND APPROACHES
      IANP1 = IAN + 1
      DO 1030  JAN = IANP1 , NOBA
                    IF ( LOBA(IAN).EQ.LOBA(JAN) )GO TO 8060
 1030 CONTINUE
 1040 CONTINUE
      PRINT 606 , NOBA
      NLINE = NLINE + 3
C-----CHECK IF APPROACH NUMBER IS ON LIST OF INBOUND APPROACHES AND
C-----ALSO ON LIST OF OUTBOUND APPROACHES
      DO 1060  IAN = 1 , NIBA
      DO 1050  JAN = 1 , NOBA
                    IF ( LIBA(IAN).EQ.LOBA(JAN) )GO TO 8070
 1050 CONTINUE
 1060 CONTINUE
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8010 CONTINUE
      PRINT 801 , NIBA
      STOP  801
 8020 CONTINUE
      PRINT 802 , IAN,LIBA(IAN)
      STOP  802
 8030 CONTINUE
      PRINT 803 , IAN,LIBA(IAN),JAN,LIBA(JAN)
      STOP  803
 8040 CONTINUE
      PRINT 804 , NOBA
      STOP  804
 8050 CONTINUE
      PRINT 805 , IAN,LOBA(IAN)
      STOP  805
 8060 CONTINUE
      PRINT 806 , IAN,LOBA(IAN),JAN,LOBA(JAN)
      STOP  806
 8070 CONTINUE
      PRINT 807 , IAN,LIBA(IAN),JAN,LOBA(JAN)
      STOP  807
      END                                                               READIO
```

```
      SUBROUTINE READOP
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / XPERL(6,6),ITSIM,NTEST,ZTEMPD(33)
  501 FORMAT(2I4,F4.1,2I4,2F4.0)
  601 FORMAT(///,12X,47HTOTAL NUMBER OF INBOUND AND OUTBOUND APPROACHES,
     *       3H = ,I2,///)
  602 FORMAT(8X,5HTABLE,I3,37H  -  DRIVER-VEHICLE PROCESSOR OPTIONS,///,
     *       12X,39HTIME FOR GENERATING VEHICLES (MIN) ----,I5,/,
     *       12X,39HMINIMUM HEADWAY FOR VEHICLES (SEC) ----,F7.1,/,
     *       12X,39HNUMBER OF VEHICLE CLASSES -------------,I5,/,
     *       12X,39HNUMBER OF DRIVER CLASSES --------------,I5,/,
     *       12X,39HPERCENT OF LEFT TURNS IN MEDIAN LANE --,F6.0,/,
     *       12X,39HPERCENT OF RIGHT TURNS IN CURB LANE ---,F6.0,///)
  808 FORMAT(24H0NUMBER OF APPROACHES = ,I3,17H IS LT 2 OR GT 12)
  809 FORMAT(53H0NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND,
     *       14H APPROACHES = ,I3,30H IS NE NUMBER OF APPROACHES = ,I3)
  810 FORMAT(31H0TIME FOR GENERATING VEHICLES =,I3,
     *       18H IS LT 12 OR GT 65)
  811 FORMAT(35H0MINIMUN HEADWAY BETWEEN VEHICLES =,F4.1,10H IS GT 5.0)
  812 FORMAT(28H0NUMBER OF VEHICLE CLASSES =,I3,17H IS LT 0 OR GT 15)
  813 FORMAT(27H0NUMBER OF DRIVER CLASSES =,I3,16H IS LT 0 OR GT 5)
  814 FORMAT(39H0PERCENT OF LEFT TURNS IN MEDIAN LANE =,F7.1,8H IS LT 5,
     *       15H0.0 OR GT 100.0)
  815 FORMAT(38H0PERCENT OF RIGHT TURNS IN CURB LANE =,F7.1,9H IS LT 50,
     *       14H.0 OR GT 100.0)
C
C-----SUBROUTINE READOP READS THE NUMBER OF APPROACHES AND THE DRIVER-
C-----VEHICLE PROCESSOR OPTIONS AND CHECKS FOR ERRORS
C
C-----READ NUMBER OF APPROACHES AND DRIVER-VEHICLE PROCESSOR OPTIONS
      READ 501 , NAP,ITSIM,HMIN,NVEHCL,NDRICL,FPERL,FPERR
                      IF ( ITSIM . EQ . 0 )         ITSIM = 12
                      IF ( HMIN . LE . 1.0 )        HMIN = 1.0
                      IF ( NVEHCL . EQ . 0 )        NVEHCL = 10
                      IF ( NDRICL . EQ . 0 )        NDRICL = 3
                      IF ( FPERL . LE . 0.0 )       FPERL = 80.0
                      IF ( FPERR . LE . 0.0 )       FPERR = 80.0
C-----ECHO-PRINT VALUES
      PRINT 601 , NAP
      NLINE = NLINE + 7
                      IF ( NLINE+12 . GT . LINES ) CALL  HEADER
      PRINT 602 , NTABL,ITSIM,HMIN,NVEHCL,NDRICL,FPERL,FPERR
      NLINE = NLINE + 12
      NTABL = NTABL + 1
C-----CHECK FOR ERRORS
                      IF ( NAP . LT .  2 )          GO TO 8080
                      IF ( NAP . GT . 12 )          GO TO 8080
      NTEST = NIBA + NOBA
                      IF ( NTEST . NE . NAP )       GO TO 8090
                      IF ( ITSIM . LT . 12 )        GO TO 8100
                      IF ( ITSIM . GT . 65 )        GO TO 8100
                      IF ( HMIN . GT . 5.0 )        GO TO 8110
                      IF ( NVEHCL . LE .  0 )       GO TO 8120
                      IF ( NVEHCL . GT . 15 )       GO TO 8120
                      IF ( NDRICL . LE . 0 )        GO TO 8130
                      IF ( NDRICL . GT . 5 )        GO TO 8130
                      IF ( FPERL . LT .  50.0 )     GO TO 8140
                      IF ( FPERL . GT . 100.0 )     GO TO 8140
                      IF ( FPERR . LT .  50.0 )     GO TO 8150
                      IF ( FPERR . GT . 100.0 )     GO TO 8150
      SIMTIM = ITSIM*60
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8080 CONTINUE
      PRINT 808 , NAP
      STOP  808
 8090 CONTINUE
      PRINT 809 , NTEST,NAP
      STOP  809
 8100 CONTINUE
      PRINT 810 , ITSIM
      STOP  810
 8110 CONTINUE
      PRINT 811 , HMIN
      STOP  811
 8120 CONTINUE
      PRINT 812 , NVEHCL
      STOP  812
 8130 CONTINUE
      PRINT 813 , NDRICL
      STOP  813
 8140 CONTINUE
      PRINT 814 , FPERL
      STOP  814
 8150 CONTINUE
      PRINT 815 , FPERR
      STOP  815
      END
```

```
      SUBROUTINE READAP
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / XPERL(6,6),I,IA,IAN,IUSED(12),IYES,J,JAAZIM,
     *                  JDIST,JVOL,KDIST,LTEST,MDEGST,MLANES,N,PDIST,
     *                  SUM,XMEANS,X85PER,YPERT(6)
      DIMENSION         LGEOM1(6),LGEOM2(6)
      EQUIVALENCE       (LGEOM1,VMAX),(LGEOM2,VMIN),(TMEAN,SUM),
     *                  (JAN,IITURN(1,1)),(KGEOM,IITURN(1,2))
  501 FORMAT(2I4,12X,I4,I3,4X,A4,A3,I5,F6.2,2F5.1,6F3.0,A3)
  502 FORMAT(15F5.1)
  503 FORMAT(4X,2I4,13X,F9.0,10X,2I4,13X,F9.0)
  504 FORMAT(20A4)
  601 FORMAT(8X,5HTABLE,I3,26H  -  LISTING OF APPROACHES,//)
  602 FORMAT(12X,39HAPPROACH NUMBER ------------------------,I5,/,
     *       12X,39HAPPROACH AZIMUTH -----------------------,I5,/,
     *       12X,39HNUMBER OF LANES ------------------------,I5)
  603 FORMAT(12X,39HNUMBER OF DEGREES FOR STRAIGHT ---------,I5,/,
     *       12X,39HHEADWAY DISTRIBUTION NAME --------------,1X,A4,A3)
  604 FORMAT(1H+,61X,11HPARAMETER =,F8.2)
  605 FORMAT(12X,39HEQUIVALENT HOURLY VOLUME (VPH) ---------,I5,/,
     *       12X,39HAPPROACH MEAN SPEED (MPH) --------------,F7.1,/,
     *       12X,39HAPPROACH 85 PERCENTILE SPEED (MPH) -----,F7.1)
  606 FORMAT(16X,35HOUTBOUND APPROACH NUMBER -----------,6I5)
  607 FORMAT(12X,39HPERCENT GOING TO OUTBOUND APPROACHES --,1X,6F5.0)
  608 FORMAT(12X,39HUSER SUPPLIED PERCENT OF VEHICLES -----,3X,A3)
  609 FORMAT(16X,35HVEHICLE CLASS NUMBER ---------------,15I5)
  610 FORMAT(12X,39HUSER SUPPLIED PERCENT OF VEHICLES -----,1X,15F5.1)
  611 FORMAT(12X,39HPROGRAM SUPPLIED PERCENT OF VEHICLES --,1X,15F5.1)
  612 FORMAT(12X,35HPERCENT OF TRAFFIC ENTERING ON LANE,I2,2H -,F6.0)
  613 FORMAT(1H+,57X,13H(MEDIAN LANE))
  614 FORMAT(1H+,74X,45HWARNING  -  THIS LANE WILL NOT HAVE VEHICLES ,
     *       11HENTERING IT)
  615 FORMAT(1H+,57X,11H(CURB LANE))
  616 FORMAT(12X,29HTOTAL NUMBER OF APPROACHES = ,I2,///)
  816 FORMAT(4A4,I3,17H IS LE 0 OR GT 12)
  817 FORMAT(4A4,I3,23H IS USED MORE THAN ONCE)
  818 FORMAT(4A4,I3,10H AZIMUTH =,I4,18H IS LT 0 OR GT 360)
  819 FORMAT(4A4,I3,18H NUMBER OF LANES =,I2,16H IS LE 0 OR GT 6)
  820 FORMAT(4A4,I3,36H IS NOT ON INBOUND OR OUTBOUND LISTS)
  821 FORMAT(4A4,I3,34H NUMBER OF DEGREES FOR STRAIGHT = ,I3,
     *       17H IS LT 0 OR GT 45)
  822 FORMAT(4A4,I3,28H HEADWAY DISTRIBUTION NAME (,A4,A3,
     *       49H) IS NOT (CONSTAN)OR(ERLANG )OR(GAMMA  )OR(LOGNRM,
     *       4HL)OR,/,63X,31H(NEGEXP )OR(SNEGEXP)OR(UNIFORM))
  823 FORMAT(4A4,I3,47H HAS ZERO VOLUME WITH A VALID DISTRIBUTION NAME)
  824 FORMAT(4A4,I3,29H PARAMETER FOR DISTRIBUTION =,F7.2,10H IS LE 0.0)
  825 FORMAT(4A4,I3,36H PARAMETER FOR ERLANG DISTRIBUTION =,F7.2,
     *       24H IS NOT AN INTEGER VALUE)
  826 FORMAT(4A4,I3,35H PARAMETER FOR GAMMA DISTRIBUTION =,F7.2,
     *       10H IS LT 1.0)
  827 FORMAT(4A4,I3,43H PARAMETER FOR SHIFTED NEGATIVE EXPONENTIAL ,
     *       14HDISTRIBUTION =,F7.2,21H IS GE MEAN HEADWAY =,F7.2)
  828 FORMAT(4A4,I3,27H EQUIVALENT HOURLY VOLUME =,I5,14H IS LT 0 OR GT,
     *       5H 4000)
  829 FORMAT(4A4,I3,22H APPROACH MEAN SPEED =,F6.1,17H IS LE 10.0 OR GT,
     *       5H 80.0)
  830 FORMAT(4A4,I3,31H APPROACH 85 PERCENTILE SPEED =,F6.1,
     *       28H IS LT APPROACH MEAN SPEED =,F6.1,11H OR GT 90.0)
  831 FORMAT(4A4,I3,35H APPROACH TURNING PERCENTAGES SUM =,F6.1,
     *       13H IS NOT 100.0)
  832 FORMAT(4A4,I3,32H USER SUPPLIED PERCENT OF VEHICL,
     *       13HES OPTION = (,A3,21H) IS NOT (YES)OR(NO ))
  833 FORMAT(4A4,I3,28H NUMBER OF VEHICLE CLASSES =,I3,
     *       54H IS NOT 10 WHEN ASKING FOR PROGRAM SUPPLIED PERCENT OF,
     *       27H VEHICLES IN TRAFFIC STREAM)
  834 FORMAT(4A4,I3,45H USER SUPPLIED PERCENT OF VEHICLES MAKING UP ,
     *       24HTHE TRAFFIC STREAM SUM =,F6.1,13H IS NOT 100.0)
  835 FORMAT(4A4,I3,5H LANE,I2,37H DOES NOT START AT THE SAME LGEOM(1) ,
     *       19HAS THE FIRST LANE (,I4,1H))
  836 FORMAT(4A4,I3,37H HAS VEHICLES ENTERING ON LANE NUMBER,I2,
     *       53H THAT DOES NOT EXIST AT THE BEGINNING OF THE APPROACH)
  837 FORMAT(4A4,I3,39H PERCENT OF VEHICLES IN EACH LANE SUM =,F6.1,
     *       13H IS NOT 100.0)
  838 FORMAT(4A4,I3,19H HAS A MEAN SPEED =,F7.1,20H AND A 85 PERCENTILE,
     *       8H SPEED =,F7.1,/,37H WHICH GIVES ONE STANDARD DEVIATION =,
     *       F7.1,31H WHICH IS GREATER THAN THE MEAN)
  839 FORMAT(4A4,I3,48H ON OUTBOUND LIST YET HAS INBOUND DATA SPECIFIED)
  840 FORMAT(4A4,I3,32H IS ON OUTBOUND LIST YET HAS PER,
     *       52HCENT OF EACH VEHICLE CLASS MAKING THE TRAFFIC STREAM)
  841 FORMAT(4A4,I3,29H HAS NO INFORMATION SPECIFIED)
C
C-----SUBROUTINE READAP READS THE APPROACH INFORMATION AND CHECKS FOR
C-----ERRORS
C
                    IF ( NLINE+24 . GT . LINES ) CALL  HEADER
      PRINT 601 , NTABL
      NLINE = NLINE + 3
      NTABL = NTABL + 1
      DO 1010  I = 1 , 12
      IUSED(I) = 0
 1010 CONTINUE
C-----READ INFORMATION FOR EACH APPROACH
      DO 4040  I = 1 , NAP
C-----READ APPROACH INFORMATION
      READ 501 , IA,JAAZIM,MLANES,MDEGST,JDIST,KDIST,JVOL,PDIST,XMEANS,
     *           X85PER,YPERT,IYES
                    IF ( MDEGST . EQ . 0 )        MDEGST = 20
                    IF ( IYES . EQ . NBLANK )     IYES = NNO
      LTEST = NLINE + 5
      DO 1020  IAN = 1 , NIBA
                    IF ( IA . EQ . LIBA(IAN) )    LTEST = LTEST+MLANES+10
 1020 CONTINUE
                    IF ( I . EQ . NAP )           LTEST = LTEST + 4
                    IF ( LTEST . GT . LINES )     CALL HEADER
C-----ECHO-PRINT DATA
      PRINT 602 , IA,JAAZIM,MLANES
      NLINE = NLINE + 3
                    IF ( IA . LE .  0 )           GO TO 8160
                    IF ( IA . GT . 12 )           GO TO 8160
                    IF ( IUSED(IA) . NE . 0 )     GO TO 8170
                    IF ( JAAZIM . LT .   0 )      GO TO 8180
                    IF ( JAAZIM . GT . 360 )      GO TO 8180
                    IF ( MLANES . LE . 0 )        GO TO 8190
                    IF ( MLANES . GT . 6 )        GO TO 8190
C-----CHECK IF APPROACH IS ON LIST OF INBOUND APPROACHES
      DO 1030  IAN = 1 , NIBA
                    IF ( IA . EQ . LIBA(IAN) )    GO TO 2010
 1030 CONTINUE
C-----CHECK IF APPROACH IS ON LIST OF OUTBOUND APPROACHES
      DO 1040  IAN = 1 , NOBA
                    IF ( IA . EQ . LOBA(IAN) )    GO TO 4010
 1040 CONTINUE
      GO TO 8200
 2010 CONTINUE
C-----APPROACH IS INBOUND
      PRINT 603 , MDEGST,JDIST,KDIST
      NLINE = NLINE + 2
                    IF ( MDEGST . LT .  0 )       GO TO 8210
                    IF ( MDEGST . GT . 45 )       GO TO 8210
      DO 2020  J = 1 , 7
```

```
      IF ( JDIST.EQ.IDISTN(1,J).AND.KDIST.EQ.IDISTN(2,J) )
     *                                          GO TO 2030
 2020 CONTINUE
      IF ( JDIST.EQ.NBLANK.AND.KDIST.EQ.NBLANK
     *                 .AND.JVOL .EQ.0        ) GO TO 3010
      GO TO 8220
 2030 CONTINUE
                    IF ( JVOL . EQ . 0 )          GO TO 8230
      JDIST = J
                    IF ( JDIST . EQ . 1 )         GO TO 3010
                    IF ( JDIST . EQ . 5 )         GO TO 3010
      PRINT 604 , PDIST
                    IF ( PDIST . LE . 0.0 )       GO TO 8240
      IF (  JDIST . EQ . 2 . AND .
     *       ABS(PDIST-INT(PDIST)) . GT . ZERO )  GO TO 8250
             IF ( JDIST.EQ.3.AND.PDIST.LT.1.0 )   GO TO 8260
      TMEAN = 3600.0/JVOL
             IF ( JDIST.EQ.6.AND.PDIST.GE.TMEAN)  GO TO 8270
 3010 CONTINUE
      PRINT 605 , JVOL,XMEANS,X85PER
      PRINT 606 , (LOBA(J),J=1,NOBA)
      PRINT 607 , (YPERT(J),J=1,NOBA)
      NLINE = NLINE + 5
                    IF ( JVOL . LT .     0 )      GO TO 8280
                    IF ( JVOL . GT . 4000 )       GO TO 8280
                    IF ( XMEANS . LE . 10.0 )     GO TO 8290
                    IF ( XMEANS . GT . 80.0 )     GO TO 8290
                    IF ( X85PER . LT . XMEANS )   GO TO 8300
                    IF ( X85PER . GT . 90.0 )     GO TO 8300
      SUM = 0.0
      DO 3020 JAN = 1 , NOBA
      SUM = SUM + YPERT(JAN)
 3020 CONTINUE
                    IF ( ABS(SUM-100.0).GT.ZERO )GO TO 8310
      PRINT 608 , IYES
      NLINE = NLINE + 1
      IF ( IYES.NE.NYES . AND . IYES.NE.NNO )     GO TO 8320
      PRINT 609 , (J,J=1,NVEHCL)
      NLINE = NLINE + 1
                    IF ( IYES . EQ . NNO )        GO TO 3030
      READ 502 , (XPERV(J,IAN),J=1,NVEHCL)
      PRINT 610 , (XPERV(J,IAN),J=1,NVEHCL)
      GO TO 3040
 3030 CONTINUE
                    IF ( NVEHCL . NE . 10 )       GO TO 8330
      PRINT 611 , (XPERV(J,IAN),J=1,NVEHCL)
 3040 CONTINUE
      NLINE = NLINE + 1
      SUM = 0.0
      DO 3050  J = 1 , NVEHCL
      SUM = SUM + XPERV(J,IAN)
 3050 CONTINUE
                    IF ( ABS(SUM-100.0).GT.ZERO )GO TO 8340
C-----READ PERCENT OF TRAFFIC STREAM FOR EACH LANE
      READ 503 , (LGEOM1(J),LGEOM2(J),XPERL(J,IAN),J=1,MLANES)
      KGEOM = LGEOM1(1)
      DO 3070  J = 1 , MLANES
                    IF ( LGEOM1(J) . NE . KGEOM )GO TO 8350
      PRINT 612 , J,XPERL(J,IAN)
                    IF ( J . EQ . 1 )             PRINT 613
                    IF ( LGEOM1(J).GE.LGEOM2(J) )GO TO 3060
      MAYENT(IAN,J) = .TRUE.
                    IF ( XPERL(J,IAN).LE.0.0 )    PRINT 614
      GO TO 3070
 3060 CONTINUE
      MAYENT(IAN,J) = .FALSE.
                    IF ( XPERL(J,IAN).GT.0.0 )    GO TO 8360
 3070 CONTINUE
                    IF ( MLANES . NE . 1 )        PRINT 615
      NLINE = NLINE + MLANES
      SUM = 0.0
      DO 3080  J = 1 , MLANES
      SUM = SUM + XPERL(J,IAN)
 3080 CONTINUE
                    IF ( ABS(SUM-100.0).GT.ZERO )GO TO 8370
      NDEGST(IAN) = MDEGST
      NLANES(IAN) = MLANES
      IVOL(IAN) = JVOL
      IDIST(IAN) = JDIST
      PARAM(IAN) = PDIST
      VMEAN(IAN) = FPSMPH*XMEANS
      VSIGMA(IAN) = FPSMPH*(X85PER-XMEANS)/1.0364334
                    IF ( VSIGMA(IAN).LT.ZERO )    VSIGMA(IAN) = 0.0
            IF ( VSIGMA(IAN) . GE . VMEAN(IAN) ) GO TO 8380
      DO 3090  JAN = 1 , NOBA
      XPERT(JAN,IAN) = YPERT(JAN)
 3090 CONTINUE
      GO TO 4030
 4010 CONTINUE
C-----APPROACH IS OUTBOUND
                    IF ( JDIST . NE . NBLANK )    GO TO 8390
      IF ( IYES.NE.NYES . AND . IYES.NE.NNO )     GO TO 8320
                    IF ( IYES . EQ . NYES )       GO TO 8400
C-----DUMMY READ LANE INFORMATION
      N = (MLANES+1)/2
      DO 4020  J = 1 , N
      READ 504
 4020 CONTINUE
 4030 CONTINUE
C-----INFORMATIUN FOR ALL APPROACHES
      IUSED(IA) = 1
      IAAZIM(IA) = JAAZIM
      PRINT 501
      PRINT 501
      NLINE = NLINE + 2
C-----END OF APPROACH LOOP
 4040 CONTINUE
C-----CHECK IF INFORMATION FOR EACH INBOUND APPROACH WAS SPECIFIED
      DO 5010  IAN = 1 , NIBA
      IA = LIBA(IAN)
                    IF ( IUSED(IA) . EQ . 0 )     GO TO 8410
 5010 CONTINUE
C-----CHECK IF INFORMATION FOR EACH OUTBOUND APPROACH WAS SPECIFIED
      DO 5020  IAN = 1 , NOBA
      IA = LOBA(IAN)
                    IF ( IUSED(IA) . EQ . 0 )     GO TO 8410
 5020 CONTINUE
      PRINT 616 , NAP
      NLINE = NLINE + 4
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8160 CONTINUE
      PRINT 816 , IFORM,IA
      STOP  816
 8170 CONTINUE
      PRINT 817 , IFORM,IA
      STOP  817
 8180 CONTINUE
      PRINT 818 , IFORM,IA,JAAZIM
      STOP  818
 8190 CONTINUE
      PRINT 819 , IFORM,IA,MLANES
      STOP  819
 8200 CONTINUE
      PRINT 820 , IFORM,IA
      STOP  820
 8210 CONTINUE
      PRINT 821 , IFORM,IA,MDEGST
      STOP  821
 8220 CONTINUE
      PRINT 822 , IFORM,IA,JDIST,KDIST
      STOP  822
```

```
8230 CONTINUE
     PRINT 823 , IFORM,IA
     STOP  823
8240 CONTINUE
     PRINT 824 , IFORM,IA,PDIST
     STOP  824
8250 CONTINUE
     PRINT 825 , IFORM,IA,PDIST
     STOP  825
8260 CONTINUE
     PRINT 826 , IFORM,IA,PDIST
     STOP  826
8270 CONTINUE
     PRINT 827 , IFORM,IA,PDIST,TMEAN
     STOP  827
8280 CONTINUE
     PRINT 828 , IFORM,IA,JVOL
     STOP  828
8290 CONTINUE
     PRINT 829 , IFORM,IA,XMEANS
     STOP  829
8300 CONTINUE
     PRINT 830 , IFORM,IA,X85PER,XMEANS
     STOP  830
8310 CONTINUE
     PRINT 831 , IFORM,IA,SUM
     STOP  831
8320 CONTINUE
     PRINT 832 , IFORM,IA,IYES
     STOP  832
8330 CONTINUE
     PRINT 833 , IFORM,IA,NVEHCL
     STOP  833
8340 CONTINUE
     PRINT 834 , IFORM,IA,SUM
     STOP  834
8350 CONTINUE
     PRINT 835 , IFORM,IA,J,KGEOM
     STOP  835
8360 CONTINUE
     PRINT 836 , IFORM,IA,J
     STOP  836
8370 CONTINUE
     PRINT 837 , IFORM,IA,SUM
     STOP  837
8380 CONTINUE
     PRINT 838 , IFORM,IA,XMEANS,X85PER,VSIGMA(IAN)
     STOP  838
8390 CONTINUE
     PRINT 839 , IFORM,IA
     STOP  839
8400 CONTINUE
     PRINT 840 , IFORM,IA
     STOP  840
8410 CONTINUE
     PRINT 841 , IFORM,IA
     STOP  841
     END                                                         READAP
```

```
      SUBROUTINE READGP
      COMMON / ZTEMPD / XPERL(6,6),I,NARCS,NLINES,NSDRC,ZTEMPD(31)
  501 FORMAT(20I4)
  502 FORMAT(20A4)
  842 FORMAT(17H0NUMBER OF ARCS =,I3,17H IS LT 0 OR GT 20)
  843 FORMAT(18H0NUMBER OF LINES =,I3,18H IS LT 0 OR GT 100)
  844 FORMAT(51H0NUMBER OF SIGHT DISTANCE RESTRICTION COORDINATES =,I3,
     *       17H IS LT 0 OR GT 20)
C
C-----SUBROUTINE READGP DUMMY READS THE GEOMETRY PROCESSOR DATA
C
C-----READ NUMBER OF ARCS
      READ 501 , NARCS
                     IF ( NARCS . LT .  0 )        GO TO 8420
                     IF ( NARCS . EQ .  0 )        GO TO 5010
                     IF ( NARCS . GT . 20 )        GO TO 8420
C-----DUMMY READ INFORMATION FOR EACH ARC
      DO 5020  I = 1 , NARCS
      READ 502
 5020 CONTINUE
 5010 CONTINUE
C-----READ NUMBER OF LINES
      READ 501 , NLINES
                     IF ( NLINES . LT .   0 )      GO TO 8430
                     IF ( NLINES . EQ .   0 )      GO TO 5040
                     IF ( NLINES . GT . 100 )      GO TO 8430
C-----DUMMY READ INFORMATION FOR EACH LINE
      DO 5030  I = 1 , NLINES
      READ 502
 5030 CONTINUE
 5040 CONTINUE
C-----READ NUMBER OF SIGHT DISTANCE RESTRICTION COORDINATES
      READ 501 , NSDRC
                     IF ( NSDRC . LT .  0 )        GO TO 8440
                     IF ( NSDRC . EQ .  0 )        GO TO 5060
                     IF ( NSDRC . GT . 20 )        GO TO 8440
C-----DUMMY READ INFORMATION FOR SIGHT DISTANCE RESTRICTION COORDINATES
      DO 5050  I = 1 , NSDRC
      READ 502
 5050 CONTINUE
 5060 CONTINUE
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8420 CONTINUE
      PRINT 842 , NARCS
      STOP  842
 8430 CONTINUE
      PRINT 843 , NLINES
      STOP  843
 8440 CONTINUE
      PRINT 844 , NSDRC
      STOP  844
      END                                                         READGP
```

```
      SUBROUTINE READYO
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / XPERL(6,6),I,J,SUM,ZTEMPD(32)
  501 FORMAT(26A3)
  502 FORMAT(15F5.1)
  503 FORMAT(20I4)
  601 FORMAT(8X,5HTABLE,I3,
     *       45H  -  DRIVER AND VEHICLE CLASS CHARACTERISTICS,///,
     *       12X,39HUSER SUPPLIED DRIVER CLASS SPLIT ------,3X,A3,/,
     *       12X,39HUSER SUPPLIED VEHICLE CHARACTERISTICS -,3X,A3,/,
     *       12X,39HUSER SUPPLIED DRIVER CHARACTERISTICS --,3X,A3)
  602 FORMAT(16X,35HVEHICLE CLASS NUMBER --------------,15I5)
  603 FORMAT(12X,39HVEHICLE LOGOUT SUMMARY REQUESTED ------,
     *       1X,15(2X,A3))
  604 FORMAT(16X,35HDRIVER CLASS NUMBER ---------------,05I5)
  605 FORMAT(12X,39HDRIVER LOGOUT SUMMARY REQUESTED -------,1X,5(2X,A3))
  606 FORMAT(12X,18HDRIVER CLASS SPLIT)
  607 FORMAT(1H+,40X,22H(USER SUPPLIED VALUES),/)
  608 FORMAT(1H+,40X,25H(PROGRAM SUPPLIED VALUES),/)
  609 FORMAT(16X,20HVEHICLE CLASS NUMBER,I3,1X,11(1H-),1X,5F5.1)
  610 FORMAT(12X,23HVEHICLE CHARACTERISTICS)
  611 FORMAT(/,16X,35HLENGTH OF VEHICLES (FT) -----------,15I5)
  612 FORMAT(16X,35HVEHICLE OPERATIONAL FACTOR --------,15I5)
  613 FORMAT(16X,35HMAXIMUM DECELERATION (FT/SEC/SEC) -,15I5)
  614 FORMAT(16X,35HMAXIMUM ACCELERATION (FT/SEC/SEC) -,15I5)
  615 FORMAT(16X,35HMAXIMUM VELOCITY (FT/SEC) ---------,15I5)
  616 FORMAT(16X,35HMINIMUM TURNING RADIUS (FT) -------,15I5)
  617 FORMAT(12X,22HDRIVER CHARACTERISTICS)
  618 FORMAT(/,16X,35HDRIVER OPERATIONAL FACTOR ---------,05I5)
  619 FORMAT(16X,35HDRIVER REACTION TIME (SEC) --------,05F5.1)
  845 FORMAT(35H0USER SUPPLY DRIVER CLASS SPLIT = (,A3,8H) IS NOT,
     *       13H (YES)OR(NO ))
  846 FORMAT(40H0USER SUPPLY VEHICLE CHARACTERISTICS = (,A3,8H) IS NOT,
     *       13H (YES)OR(NO ))
  847 FORMAT(39H0USER SUPPLY DRIVER CHARACTERISTICS = (,A3,8H) IS NOT,
     *       13H (YES)OR(NO ))
  848 FORMAT(14H0VEHICLE CLASS,I2,29H LOGOUT SUMMARY REQUESTED = (,A3,
     *       21H) IS NOT (YES)OR(NO ))
  849 FORMAT(13H0DRIVER CLASS,I2,29H LOGOUT SUMMARY REQUESTED = (,A3,
     *       21H) IS NOT (YES)OR(NO ))
  850 FORMAT(28H0NUMBER OF VEHICLE CLASSES =,I3,11H IS NOT 10 ,
     *       46HWHEN DEFAULT DRIVER CLASS SPLITS ARE REQUESTED)
  851 FORMAT(27H0NUMBER OF DRIVER CLASSES =,I3,10H IS NOT 3 ,
     *       46HWHEN DEFAULT DRIVER CLASS SPLITS ARE REQUESTED)
  852 FORMAT(38H0DRIVER CLASS SPLITS FOR VEHICLE CLASS,I3,6H SUM =,
     *       F6.1,13H IS NOT 100.0)
  853 FORMAT(28H0NUMBER OF VEHICLE CLASSES =,I3,11H IS NOT 10 ,
     *       50HWHEN DEFAULT VEHICLE CHARACTERISTICS ARE REQUESTED)
  854 FORMAT(24H0LENGTH OF VEHICLE CLASS,I3,2H =,I4,14H IS LT 5 OR GT,
     *       4H 100)
  855 FORMAT(31H0DRIVER FACTOR OF VEHICLE CLASS,I3,2H =,I4,6H IS LT,
     *       13H 50 OR GT 150)
  856 FORMAT(38H0DECELERATION MAXIMUM OF VEHICLE CLASS,I3,2H =,I4,
     *       17H IS LT 4 OR GT 12)
  857 FORMAT(38H0ACCELERATION MAXIMUM OF VEHICLE CLASS,I3,2H =,I4,
     *       17H IS LT 3 OR GT 18)
  858 FORMAT(34H0VELOCITY MAXIMUM OF VEHICLE CLASS,I3,2H =,I4,
     *       19H IS LT 10 OR GT 235)
  859 FORMAT(40H0MINIMUM TURNING RADIUS OF VEHICLE CLASS,I3,2H =,I4,
     *       18H IS LT 4 OR GT 300)
  860 FORMAT(27H0NUMBER OF DRIVER CLASSES =,I3,10H IS NOT 3 ,
     *       49HWHEN DEFAULT DRIVER CHARACTERISTICS ARE REQUESTED)
  861 FORMAT(30H0DRIVER FACTOR OF DRIVER CLASS,I3,2H =,I4,6H IS LT,
     *       13H 50 OR GT 150)
  862 FORMAT(26H0PIJR TIME OF DRIVER CLASS,I3,2H =,F6.2,6H IS LT,
     *       16H 0.25 OR GT 5.00)
C
C-----SUBROUTINE READYO READS THE YES OPTIONS AND CHECKS FOR ERRORS
C
C-----DUMMY READ GEOMETRY PROCESSOR OPTIONS
      READ (5,501,END=1010)
C-----READ THE YES OPTIONS
      READ (5,501,END=1010)  IYESP,IYESV,IYESD,IYESVL,IYESDL
      GO TO 1020
 1010 CONTINUE
      IEOF = .TRUE.
 1020 CONTINUE
C-----SET DEFAULT VALUES FOR YES OPTIONS
                   IF ( IYESP . EQ . NBLANK )    IYESP = NNO
                   IF ( IYESV . EQ . NBLANK )    IYESV = NNO
                   IF ( IYESD . EQ . NBLANK )    IYESD = NNO
      DO 1030  I = 1 , NVEHCL
           IF ( IYESVL(I) . EQ . NBLANK )        IYESVL(I) = NNO
 1030 CONTINUE
      DO 1040  I = 1 , NDRICL
           IF ( IYESDL(I) . EQ . NBLANK )        IYESDL(I) = NNO
 1040 CONTINUE
                   IF ( NLINE+12 . GT . LINES ) CALL  HEADER
C-----ECHO PRINT YES OPTIONS
      PRINT 601 , NTABL,IYESP,IYESV,IYESD
      NLINE = NLINE + 6
      NTABL = NTABL + 1
      PRINT 602 , (I,I=1,NVEHCL)
      PRINT 603 , (IYESVL(I),I=1,NVEHCL)
      PRINT 604 , (I,I=1,NDRICL)
      PRINT 605 , (IYESDL(I),I=1,NDRICL)
      PRINT 501
      PRINT 501
      NLINE = NLINE + 6
      IF ( IYESP.NE.NYES . AND . IYESP.NE.NNO )  GO TO 8450
      IF ( IYESV.NE.NYES . AND . IYESV.NE.NNO )  GO TO 8460
      IF ( IYESD.NE.NYES . AND . IYESD.NE.NNO )  GO TO 8470
      DO 2010  I = 1 , NVEHCL
      IF ( IYESVL(I).NE.NYES . AND . IYESVL(I).NE.NNO )
     *                                           GO TO 8480
 2010 CONTINUE
      DO 2020  I = 1 , NDRICL
      IF ( IYESDL(I).NE.NYES . AND . IYESDL(I).NE. NNO )
     *                                           GO TO 8490
 2020 CONTINUE
                IF ( NLINE+NVEHCL+6 . GT . LINES )  CALL  HEADER
      PRINT 606
                   IF ( IYESP . EQ . NNO )      GO TO 3020
      PRINT 607
      DO 3010  I = 1 , NVEHCL
C-----READ PERCENT OF DRIVER CLASS IN VEHICLE CLASS (XPERD)
      READ 502 , (XPERD(J,I),J=1,NDRICL)
 3010 CONTINUE
      GO TO 3030
 3020 CONTINUE
                   IF ( NVEHCL . NE . 10 )      GO TO 8500
                   IF ( NDRICL . NE .  3 )      GO TO 8510
      PRINT 608
 3030 CONTINUE
C-----PRINT DEFAULT OR USER SUPPLIED VALUES OF XPERD
      PRINT 604 , (I,I=1,NDRICL)
      PRINT 501
      NLINE = NLINE + 6
      DO 3050  I = 1 , NVEHCL
      PRINT 609 , I,(XPERD(J,I),J=1,NDRICL)
      SUM = 0.0
      DO 3040  J = 1 , NDRICL
      SUM = SUM + XPERD(J,I)
 3040 CONTINUE
```

```
                  IF ( ABS(SUM-100.0).GT.ZERO )GO TO 8520
 3050 CONTINUE
      PRINT 501
      PRINT 501
      NLINE = NLINE + NVEHCL
                  IF ( NLINE+12 . GT . LINES ) CALL  HEADER
      PRINT 610
                  IF ( IYESV . EQ . NNO )       GO TO 4010
      PRINT 607
C-----READ IN VEHICLE CHARACTERISTICS
      READ 503 , (LENV  (I),I=1,NVEHCL)
      READ 503 , (IVCHAR(I),I=1,NVEHCL)
      READ 503 , (IDMAX (I),I=1,NVEHCL)
      READ 503 , (IAMAX (I),I=1,NVEHCL)
      READ 503 , (IVMAX (I),I=1,NVEHCL)
      READ 503 , (IRMIN (I),I=1,NVEHCL)
      GO TO 4020
 4010 CONTINUE
                  IF ( NVEHCL . NE . 10 )       GO TO 8530
      PRINT 608
 4020 CONTINUE
C-----PRINT VEHICLE CHARACTERISTICS
      PRINT 602 , (I        ,I=1,NVEHCL)
      PRINT 611 , (LENV  (I),I=1,NVEHCL)
      PRINT 612 , (IVCHAR(I),I=1,NVEHCL)
      PRINT 613 , (IDMAX (I),I=1,NVEHCL)
      PRINT 614 , (IAMAX (I),I=1,NVEHCL)
      PRINT 615 , (IVMAX (I),I=1,NVEHCL)
      PRINT 616 , (IRMIN (I),I=1,NVEHCL)
      PRINT 501
      PRINT 501
      NLINE = NLINE + 12
      DO 4030  I = 1 , NVEHCL
                  IF ( LENV  (I) . LT .   5 )  GO TO 8540
                  IF ( LENV  (I) . GT . 100 )  GO TO 8540
                  IF ( IVCHAR(I) . LT .  50 )  GO TO 8550
                  IF ( IVCHAR(I) . GT . 150 )  GO TO 8550
                  IF ( IDMAX (I) . LT .   4 )  GO TO 8560
                  IF ( IDMAX (I) . GT .  12 )  GO TO 8560
                  IF ( IAMAX (I) . LT .   3 )  GO TO 8570
                  IF ( IAMAX (I) . GT .  18 )  GO TO 8570
                  IF ( IVMAX (I) . LT .  10 )  GO TO 8580
                  IF ( IVMAX (I) . GT . 235 )  GO TO 8580
                  IF ( IRMIN (I) . LT .   4 )  GO TO 8590
                  IF ( IRMIN (I) . GT . 300 )  GO TO 8590
 4030 CONTINUE
                  IF ( NLINE+9 . GT . LINES )  CALL  HEADER
      PRINT 617
                  IF ( IYESD . EQ . NNO )       GO TO 5010
      PRINT 607
C-----READ IN DRIVER CHARACTERISTICS
      READ 503 , (IDCHAR(I),I=1,NDRICL)
      READ 502 , (PIJR  (I),I=1,NDRICL)
      GO TO 5020
 5010 CONTINUE
                  IF ( NDRICL . NE . 3 )        GO TO 8600
      PRINT 608
 5020 CONTINUE
C-----PRINT DRIVER CHARACTERISTICS
      PRINT 604 , (I        ,I=1,NDRICL)
      PRINT 618 , (IDCHAR(I),I=1,NDRICL)
      PRINT 619 , (PIJR  (I),I=1,NDRICL)
      PRINT 501
      PRINT 501
      PRINT 501
      NLINE = NLINE + 9
      DO 5030  I = 1 , NDRICL
                  IF ( IDCHAR(I) . LT .  50 )  GO TO 8610
                  IF ( IDCHAR(I) . GT . 150 )  GO TO 8610
                  IF ( PIJR  (I) . LT . 0.25 ) GO TO 8620
                  IF ( PIJR  (I) . GT . 5.00 ) GO TO 8620
 5030 CONTINUE
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8450 CONTINUE
      PRINT 845 , IYESP
      STOP  845
 8460 CONTINUE
      PRINT 846 , IYESV
      STOP  846
 8470 CONTINUE
      PRINT 847 , IYESD
      STOP  847
 8480 CONTINUE
      PRINT 848 , I,IYESVL(I)
      STOP  848
 8490 CONTINUE
      PRINT 849 , I,IYESDL(I)
      STOP  849
 8500 CONTINUE
      PRINT 850 , NVEHCL
      STOP  850
 8510 CONTINUE
      PRINT 851 , NDRICL
      STOP  851
 8520 CONTINUE
      PRINT 852 , I,SUM
      STOP  852
 8530 CONTINUE
      PRINT 853 , NVEHCL
      STOP  853
 8540 CONTINUE
      PRINT 854 , I,LENV  (I)
      STOP  854
 8550 CONTINUE
      PRINT 855 , I,IVCHAR(I)
      STOP  855
 8560 CONTINUE
      PRINT 856 , I,IDMAX (I)
      STOP  856
 8570 CONTINUE
      PRINT 857 , I,IAMAX (I)
      STOP  857
 8580 CONTINUE
      PRINT 858 , I,IVMAX (I)
      STOP  858
 8590 CONTINUE
      PRINT 859 , I,IRMIN (I)
      STOP  859
 8600 CONTINUE
      PRINT 860 , NDRICL
      STOP  860
 8610 CONTINUE
      PRINT 861 , I,IDCHAR(I)
      STOP  861
 8620 CONTINUE
      PRINT 862 , I,PIJR  (I)
      STOP  862
      END
```

READYO

```
      SUBROUTINE WRITDV
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / TITLE  / ITITLE(20)
      COMMON / ZTEMPD / XPERL(6,6),APIJR,DVCHAR,IAN,ID,IV,PERV,SUMP,TTV,
     *                  VCHAR,VOLIAN,VMMS,VMPS,VSIG,ZTEMPD(22)
  501 FORMAT(20A4)
  502 FORMAT(20I4)
  503 FORMAT(15F5.1)
C
C-----SUBROUTINE WRITDV CALCULATES MINIMUM AND MAXIMUM SPEEDS FOR EACH
C-----DRIVER AND VEHICLE CLASS BASED ON ONE STANDARD DEVIATION AWAY FROM
C-----THE MEAN SPEED FOR EACH APPROACH.  WRITDV ALSO WRITES THE VEHICLE
C-----AND DRIVER CHARACTERISTICS ONTO TAPE FOR SIMPRO.
C
      SUMP = 0.0
      TTV = 0.0
C-----CALCULATE THE MINIMUM AND MAXIMUM SPEEDS ALLOWABLE FOR EACH DRIVER
C-----AND VEHICLE CLASS BASED ON ONE STANDARD DEVIATION AWAY FROM THE
C-----MEAN SPEED FOR EACH APPROACH.  THIS CODE ALSO CALCULATES THE
C-----AVERAGE PIJR TIME FOR ALL DRIVER-VEHICLE UNITS
      DO 1030  IAN = 1 , NIBA
      VOLIAN = IVOL(IAN)
      TTV = TTV + VOLIAN
      VSIG = VSIGMA(IAN)
      VMMS = VMEAN(IAN) - VSIG
      VMPS = VMEAN(IAN) + VSIG
      DO 1020  IV = 1 , NVEHCL
      PERV = XPERV(IV,IAN)/10000.0
      VCHAR = IVCHAR(IV)/10000.0
      DO 1010  ID = 1 , NDRICL
      SUMP = SUMP + PIJR(ID)*PERV*XPERD(ID,IV)*VOLIAN
      DVCHAR = IDCHAR(ID)*VCHAR
                      IF ( VSIG . LE . 0.0 )        DVCHAR = 1.0
      VMIN(IAN,ID,IV) = DVCHAR*VMMS
      VMAX(IAN,ID,IV) = DVCHAR*VMPS
 1010 CONTINUE
 1020 CONTINUE
 1030 CONTINUE
      APIJR = SUMP/TTV
C-----WRITE ONTO TAPE FOR SIMPRO THE VEHICLE AND DRIVER CHARACTERISTICS
      WRITE (MODELT,501)  ITITLE
      WRITE (MODELT,502)  NVEHCL,NDRICL
      WRITE (MODELT,502)  (LENV   (IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IVCHAR(IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IDMAX  (IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IAMAX  (IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IVMAX  (IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IRMIN  (IV),IV=1,NVEHCL)
      WRITE (MODELT,502)  (IDCHAR(ID),ID=1,NDRICL)
      WRITE (MODELT,503)  (PIJR   (ID),ID=1,NDRICL),APIJR
      RETURN
      END                                                         WRITDV
```

```
      SUBROUTINE BIASLT
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / ZTEMPD / XPERL(6,6),FPER,IA,IAN,IANGLE,IAZIM,ILN,ITURN,
     *                  JA,JAN,JAZIM,JLN,MDEGST,NL,SUM,XPERTS(3,6),
     *                  ZTEMPD(3)
C-----PROCESS EACH INBOUND APPROACH
      DO 1050  IAN = 1 , NIBA
      IA = LIBA(IAN)
      XPERTS(1,IAN) = 0.0
      XPERTS(2,IAN) = 0.0
      XPERTS(3,IAN) = 0.0
      MDEGST = NDEGST(IAN)
      IAZIM = IAAZIM(IA) + 180
                      IF ( IAZIM . GT . 360 )        IAZIM = IAZIM - 360
C-----PROCESS EACH OUTBOUND APPROACH
      DO 1040  JAN = 1 , NOBA
      JA = LOBA(JAN)
C-----FIND THE ANGLE FROM THE INBOUND APPROACH TO THE OUTBOUND APPROACH
      JAZIM = IAAZIM(JA)
                      IF ( JAZIM . LT . IAZIM )      JAZIM = JAZIM + 360
      IANGLE = JAZIM - IAZIM
C-----IF THE ANGLE IS BETWEEN 0 AND 180-NDEGST THEN GO TO 1010
                      IF ( IANGLE.LT.180-MDEGST )   GO TO 1010
C-----IF THE ANGLE IS BETWEEN 180-NDEGST AND 180+NDEGST THEN GO TO 1020
                      IF ( IANGLE.LE.180+MDEGST )   GO TO 1020
C-----APPROACH JAN IS A RIGHT TURN FOR APPROACH IAN
      ITURN = 3
      GO TO 1030
 1010 CONTINUE
C-----APPROACH JAN IS A U-TURN OR A LEFT TURN FOR APPROACH IAN
      ITURN = 1
      GO TO 1030
 1020 CONTINUE
C-----APPROACH JAN IS A STRAIGHT THROUGH MOVEMENT FOR APPROACH IAN
      ITURN = 2
 1030 CONTINUE
      IITURN(JAN,IAN) = ITURN
C-----SUM THE TURNING PERCENTAGES BY TURN CODE
      XPERTS(ITURN,IAN) = XPERTS(ITURN,IAN) + XPERT(JAN,IAN)
 1040 CONTINUE
 1050 CONTINUE
C-----PROCESS EACH INBOUND APPROACH
      DO 2040  IAN = 1 , NIBA
      NL = NLANES(IAN)
      FPER = FPERL/100.0
                      IF ( NL . EQ . 1 )             FPER = 1.00
      IF ( NL.GT.1.AND.XPERL(2,IAN).LE.0.0 )         FPER = 1.00
      SUM = 0.0
C-----PROCESS EACH LANE OF INBOUND APPROACH FROM MEDIAN TO CURB
      DO 2010  ILN = 1 , NL
C-----MAXIMIZE MEDIAN LANE OCCUPANCY FOR U-TURNS AND LEFT TURNS
      XPERLO(ILN,1,IAN) = AMIN1( XPERL(ILN,IAN),FPER*XPERTS(1,IAN)-SUM )
      FPER = 1.00
      SUM = SUM + XPERLO(ILN,1,IAN)
 2010 CONTINUE
      FPER = FPERR/100.0
                      IF ( NL . EQ . 1 )             FPER = 1.00
      IF ( NL.GT.1.AND.XPERL(NL-1,IAN).LE.0.0 )      FPER = 1.00
      SUM = 0.0
C-----PROCESS EACH LANE OF INBOUND APPROACH FROM CURB TO MEDIAN
      DO 2020  ILN = 1 , NL
      JLN = NL - ILN + 1
C-----MAXIMIZE CURB LANE OCCUPANCY FOR RIGHT TURNS
      XPERLO(JLN,3,IAN) = AMIN1( XPERL(JLN,IAN)-XPERLO(JLN,1,IAN),
     *                           FPER*XPERTS(3,IAN)-SUM )
```

```
      FPER = 1.0
      SUM = SUM + XPERLO(JLN,3,IAN)
 2020 CONTINUE
C-----PROCESS EACH LANE OF INBOUND APPROACH
      DO 2030  ILN = 1 , NL
C-----DISTRIBUTE STRAIGHT THROUGH MOVEMENTS TO SATISFY LANE OCCUPANCY
      XPERLO(ILN,2,IAN) = XPERL(ILN,IAN) - XPERLO(ILN,1,IAN)
     *                                   - XPERLO(ILN,3,IAN)
C-----FACTOR XPERLO SO THAT IT RANGES FROM 0.00 TO 100.0
      XPERLO(ILN,1,IAN) = 100.0*XPERLO(ILN,1,IAN)/XPERTS(1,IAN)
      XPERLO(ILN,2,IAN) = 100.0*XPERLO(ILN,2,IAN)/XPERTS(2,IAN)
      XPERLO(ILN,3,IAN) = 100.0*XPERLO(ILN,3,IAN)/XPERTS(3,IAN)
 2030 CONTINUE
 2040 CONTINUE
      RETURN
      END                                                           BIASLT
```

```
      SUBROUTINE GENHED
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPO / NVAIAN,PARIAN,TMEAN,I,IAN,IDNUM,ISUMIV,ISUMNG,
     *                  ISUMVG,IVOLGN,IVOLIA,PERDIF,ZTEMPD(59)
      DIMENSION         MSG901(18)
      DATA       MSG901 / 4H MOR,4HE TH,4HAN 1,4H000 ,4HVEHI,4HCLES,
     *                  4H ON ,4HAN A,4HPPRO,4HACH ,4H= IN,4HCREA,
     *                  4HSE D,4HIMEN,4HSION,4H OF ,4HQTIM,4HE    /
  601 FORMAT(8X,5HTABLE,I3,36H  -  GENERATION OF APPROACH HEADWAYS,///,
     *       12X,47HAPPROACH  DISTRIBUTION    NUMBER     VOLUME    ,
     *           17HINPUT    PERCENT ,/,
     *       12X,47H NUMBER       NAME      GENERATED  GENERATED  V,
     *           17HOLUME  DIFFERENCE,/)
  602 FORMAT(15X,I2,8X,A4,A3,2(7X,I4),6X,I4,5X,F7.2)
  603 FORMAT(/,13X,5HTOTAL,21X,I4,7X,I4,6X,I4,5X,F7.2,///)
  901 FORMAT(4A4,I3,28H HAS MORE THAN 1000 VEHICLES)
C
C-----SUBROUTINE GENHED GENERATES APPROACH HEADWAYS UNDER SPECIFIED
C-----DISTRIBUTIONS USING THE ASSOCIATED LOCATION AND DISPERSION
C-----PARAMETERS
C
      ISUMNG = 0
      ISUMVG = 0
      ISUMIV = 0
C-----BEGIN INBOUND APPROACH LOOP FOR HEADWAY GENERATION
      DO 1100  IAN = 1 , NIBA
      IVOLIA = IVOL(IAN)
                    IF ( IVOLIA . EQ . 0 )          GO TO 1100
      IDNUM = IDIST(IAN)
      PARIAN = PARAM(IAN)
      TMEAN = 3600.0/IVOLIA
      QTIME(1,IAN) = 2.0*RANF(0)*TMEAN
      GO TO ( 1010,1020,1030,1040,1050,1060,1070 ) , IDNUM
 1010 CONTINUE
      QTIME(1,IAN) = 0.0
      CALL  CONST    ( QTIME(1,IAN) )
      GO TO 1080
 1020 CONTINUE
      CALL  ERLANG   ( QTIME(1,IAN) )
      GO TO 1080
 1030 CONTINUE
      CALL  GAMMA    ( QTIME(1,IAN) )
      GO TO 1080
 1040 CONTINUE
      CALL  LGNRML   ( QTIME(1,IAN) )
      GO TO 1080
 1050 CONTINUE
      CALL  NEGEXP   ( QTIME(1,IAN) )
      GO TO 1080
 1060 CONTINUE
      CALL  SNEGEX   ( QTIME(1,IAN) )
      GO TO 1080
 1070 CONTINUE
      CALL  UNIFRM   ( QTIME(1,IAN) )
 1080 CONTINUE
                    IF ( NVAIAN . LT . 0 )          GO TO 9010
                    IF ( ISUMNG . NE . 0 )          GO TO 1090
               IF ( NLINE+NIBA+11 . GT . LINES )    CALL  HEADER
      PRINT 601 , NTABL
      NTABL = NTABL + 1
      NLINE = NLINE + 6
 1090 CONTINUE
C-----PRINT GENERATED VOLUME INFORMATION
```

```
      IVOLGN = NVAIAN*3600.0/SIMTIM + 0.5
      PERDIF = 100.0*FLOAT(IVOLGN-IVOLIA)/FLOAT(IVOLIA)
      PRINT 602 , LIBA(IAN),(IDISTN(I,IDNUM),I=1,2),NVAIAN,IVOLGN,
     *           IVOLIA,PERDIF
      NVA(IAN) = NVAIAN
      ISUMNG = ISUMNG + NVAIAN
      ISUMVG = ISUMVG + IVOLGN
      ISUMIV = ISUMIV + IVOLIA
 1100 CONTINUE
                     IF ( ISUMNG . EQ . 0 )         RETURN
      PERDIF = 100.0*FLOAT(ISUMVG-ISUMIV)/FLOAT(ISUMIV)
      PRINT 603 , ISUMNG,ISUMVG,ISUMIV,PERDIF
      NLINE = NLINE + NIBA + 5
      RETURN
C-----PROCESS EXECUTION ERROR AND STOP
 9010 CONTINUE
      PRINT 901 , IFORM,LIBA(IAN)
      CALL  ABORTR  ( MSG901,71,12+10 )
      STOP  901
      END                                                          GENHED
```

```
      SUBROUTINE CONST  ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPO / NVAIAN,PARIAN,TMEAN,GENHED(9),I,ZTEMPD(58)
      DIMENSION         QTIMS(1)
      DO 1010  I = 2 , 1000
      QTIMS(I) = QTIMS(I-1) + TMEAN
                     IF ( QTIMS(I) . GT . SIMTIM )GO TO 1020
 1010 CONTINUE
      NVAIAN = -1
      RETURN
 1020 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                          CONST
```

```
      SUBROUTINE ERLANG ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),ALPHA,I,J,K,
     *                  THEAD,TR,ZTEMPD(53)
      DIMENSION         QTIMS(1)
      K = PARIAN + 0.5
      ALPHA = K/TMEAN
      DO 1020  I = 2 , 1000
      TR = 1.0
      DO 1010  J = 1 , K
      TR = TR*RANF(0)
 1010 CONTINUE
      THEAD = -ALOG(TR)/ALPHA
      QTIMS(I) = QTIMS(I-1) + THEAD
                   IF ( QTIMS(I) . GT . SIMTIM )GO TO 1030
 1020 CONTINUE
      NVAIAN = -1
      RETURN
 1030 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                               ERLANG
```

```
      SUBROUTINE GAMMA  ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),A,ALPHA,I,J,K,K1,
     *                  K2,Q,THEAD,TR,ZTEMPD(49)
      DIMENSION         QTIMS(1)
      A = PARIAN
      ALPHA = A/TMEAN
      K1 = A
      K2 = A + 1.0
      Q = A - K1
      DO 1020  I = 2 , 1000
      TR = 1.0
      K = K2
                   IF ( RANF(0) . GT . Q )       K = K1
      DO 1010  J = 1 , K
      TR = TR*RANF(0)
 1010 CONTINUE
      THEAD = -ALOG(TR)/ALPHA
      QTIMS(I) = QTIMS(I-1) + THEAD
                   IF ( QTIMS(I) . GT . SIMTIM )GO TO 1030
 1020 CONTINUE
      NVAIAN = -1
      RETURN
 1030 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                               GAMMA
```

```
      SUBROUTINE LGNRML ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),EX,EY,I,J,STDX,
     *                  STDY,SUM,THEAD,VARY,ZTEMPD(50)
      DIMENSION         QTIMS(1)
      EX = TMEAN
      STDX = PARIAN
      VARY = ALOG((STDX**2/(EX**2))+1.0)
      STDY = SQRT(VARY)
      EY = ALOG(EX) - 0.5*VARY
      DO 1020  I = 2 , 1000
      SUM = 0.0
      DO 1010  J = 1 , 12
      SUM = SUM + RANF(0)
 1010 CONTINUE
      THEAD = EXP(EY+STDY*(SUM-6.0))
      QTIMS(I) = QTIMS(I-1) + THEAD
                IF ( QTIMS(I) . GT . SIMTIM )GO TO 1030
 1020 CONTINUE
      NVAIAN = -1
      RETURN
 1030 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                               LGNRML
```

```
      SUBROUTINE NEGEXP ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),I,THEAD,ZTEMPD(57)
      DIMENSION         QTIMS(1)
      DO 1010  I = 2 , 1000
      THEAD = -ALOG(RANF(0))*TMEAN
      QTIMS(I) = QTIMS(I-1) + THEAD
                IF ( QTIMS(I) . GT . SIMTIM )GO TO 1020
 1010 CONTINUE
      NVAIAN = -1
      RETURN
 1020 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                               NEGEXP
```

```
      SUBROUTINE SNEGEX ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),I,TAU,TBAR,THEAD,
     *                  ZTEMPD(55)
      DIMENSION         QTIMS(1)
      TAU = PARIAN
      TBAR = TMEAN - TAU
      DO 1010  I = 2 , 1000
      THEAD = -ALOG(RANF(0))*TBAR + TAU
      QTIMS(I) = QTIMS(I-1) + THEAD
                 IF ( QTIMS(I) . GT . SIMTIM )GO TO 1020
 1010 CONTINUE
      NVAIAN = -1
      RETURN
 1020 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                              SNEGEXP
```

```
      SUBROUTINE UNIFRM ( QTIMS )
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / ZTEMPD / NVAIAN,PARIAN,TMEAN,GENHED(9),I,THEAD,B,
     *                  BMA,A,ZTEMPD(54)
      DIMENSION         QTIMS(1)
      A = TMEAN - SQRT3*PARIAN
      B = TMEAN + SQRT3*PARIAN
      BMA = B - A
      DO 1010  I = 2 , 1000
      THEAD = A + BMA*RANF(0)
      QTIMS(I) = QTIMS(I-1) + THEAD
                 IF ( QTIMS(I) . GT . SIMTIM )GO TO 1020
 1010 CONTINUE
      NVAIAN = -1
      RETURN
 1020 CONTINUE
      NVAIAN = I - 1
      RETURN
      END                                                              UNIFRM
```

```
C;     FUNCTION   RANF ( A )
C;     DATA       ISEED   / 1717171 /
C;     DATA       I31     / 2147483647 /
C;     DATA       TM31    / .4656612873077392578125E-9 /
C;     DATA       I16P3   / 65539 /
C;     DATA       I1       / 1 /
C;                   IF ( A )                         101 , 102 , 103
C;101  CONTINUE
C;     RANF = ISEED
C;     RETURN
C;102  CONTINUE
C;     ISEED = ISEED*I16P3
C;                   IF ( ISEED . LT . 0 )            ISEED=ISEED+I31+I1
C;     RANF = ISEED*TM31
C;     RETURN
C;103  CONTINUE
C;     ISEED = A/TM31 + 0.5
C;     GO TO 102
C;     END                                                                RANF
```

```
      SUBROUTINE GENDV
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / LITCON / FPSMPH,IDISTN(2,7),SQRT3,NBLANK,NNO,NYES
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / STATS  / SPERD(5,15),SPERL(6,6),SPERT(6,6),SPERV(15,6)
      COMMON / ZTEMPD / NGWRIT(6),NSWRIT(6),NSREAD,LVTOT,
     *                  HEAD,I,IA,IAN,IAP,ID,IDC,IDV,IL,ILN,INEXTV(6),
     *                  IPLOGO,IPRTLO,ITABL,JAP,ITURN,IV,IVC,IVEL,J,JA,
     *                  JAN,KAN,QBIG,QMIN,QTIM,VEL,ISPLHD,ZTEMPD(24)
      LOGICAL           ISPLHD,ITABL
  501 FORMAT(3X,F7.2,7I5)
  601 FORMAT(7X,F10.2,2I7,I8,2I9,1X,3I7)
  602 FORMAT(7X,F10.2,31X,I9,21X,1H1)
C
C-----SUBROUTINE GENDV GENERATES EACH INDIVIDUAL DRIVER-VEHICLE UNIT FOR
C-----SIMPRO, READS IN SPECIAL VEHICLES (IF ANY), CHECKS THE SPECIAL
C-----VEHICLE≠S LOGIN ATTRIBUTES, WRITES ALL CORRECT DRIVER-VEHICLE
C-----UNITS ONTO A TAPE FOR SIMPRO, AND COLLECT STATISTICAL DATA
C
C-----INITIALIZE VARIABLES AND ARRAYS
      ISPLHD = .TRUE.
      ITABL = .FALSE.
      LVTOT = 0
      NSREAD = 0
      QBIG = 1.00E75
      QTIM = QBIG
      DO 1010  IAN = 1 , NIBA
      INEXTV(IAN) = 1
      NGWRIT(IAN) = 0
      NSWRIT(IAN) = 0
 1010 CONTINUE
C-----CHECK TO SEE IF THERE ARE ANY SPECIAL VEHICLES AND READ ONE
                    IF ( IEOF )                     GO TO 2010
      READ (5,501,END=1020)  QTIM,IVC,IDC,IDV,JAP,IAP,ILN,IPRTLO
      J = 2
      NSREAD = NSREAD + 1
      GO TO 2010
 1020 CONTINUE
      IEOF = .TRUE.
      QTIM = QBIG
C-----START OF SORTING LOOP TO WRITE VEHICLES OUT INCREASING IN TIME
 2010 CONTINUE
      QMIN = QBIG
C-----FIND INBOUND APPROACH ASSOCIATED WITH LOWEST QUEUE-IN TIME FOR
C-----THE GENERATED VEHICLES
      DO 2020  IAN = 1 , NIBA
      IV = INEXTV(IAN)
                    IF ( IV . GT . NVA(IAN) )       GO TO 2020
                IF ( QTIME(IV,IAN) . GE . QMIN )    GO TO 2020
      QMIN = QTIME(IV,IAN)
C-----KAN IS THE STACK NUMBER OF APPROACH WITH SMALLEST QUEUE-IN TIME
      KAN = IAN
 2020 CONTINUE
C-----IF NO MORE SPECIAL VEHICLES GO TO 5010 AND GENERATE LOGIN
C-----ATTRIBUTES FOR GENERATED VEHICLE WITH SMALLEST QUEUE-IN TIME
                    IF ( IEOF )                     GO TO 5010
 3010 CONTINUE
C-----START CHECKING SPECIAL VEHICLE≠S LOGIN PARAMETERS
                    IF ( QTIM . GT . SIMTIM )       GO TO 3020
C-----IF GENERATED QUEUE-IN TIME IS LESS THAN THE NEXT SPECIAL VEHICLE≠S
C-----QUEUE-IN TIME THEN GO TO 5010 AND INSERT GENERATED VEHICLE
```

```
                  IF ( QMIN . LT . QTIM )       GO TO 5010
 3020 CONTINUE
                  IF ( ISPLHD )                 GO TO 3030
      PRINT 601
      NLINE = NLINE + 1
 3030 CONTINUE
C-----PRINT SPECIAL VEHICLE AS READ IN
      CALL  GENDVH  ( ITABL,4 )
      PRINT 601 , QTIM,IVC,IDC,IDV,JAP,IAP,ILN,IPRTLO,J
      NLINE = NLINE + 1
      NOTE(J) = 1
C-----IF THIS SPECIAL VEHICLE#S LOGIN PARAMETERS HAVE ALREADY BEEN
C-----CHECKED AND ONLY THE HEADWAY WAS CHANGED TO MEET A MINIMUM OF
C-----HMIN SECONDS THEN GO TO 3080 AND CHECK HEADWAY AGAIN
                  IF ( J . EQ . 14 )            GO TO 3080
C-----SET ERROR CODE AND GO TO 4010 AND PRINT SPECIAL VEHICLE
      J = 3
                  IF ( QTIM . LT . 0.0 )        GO TO 4010
      J = 4
                  IF ( IVC . LE . 0 )           GO TO 4010
                  IF ( IVC . GT . NVEHCL )      GO TO 4010
      J = 5
                  IF ( IDC . LE . 0 )           GO TO 4010
                  IF ( IDC . GT . NDRICL )      GO TO 4010
      J = 6
                  IF ( IDV . LE . 0 )           GO TO 4010
                  IF ( IDV . GT . MAXV )        GO TO 4010
      DO 3040  JAN = 1 , NOBA
                  IF ( JAP . EQ . LOBA(JAN) )  GO TO 3050
 3040 CONTINUE
      J = 7
      GO TO 4010
 3050 CONTINUE
      DO 3060  IAN = 1 , NIBA
                  IF ( IAP . EQ . LIBA(IAN) )  GO TO 3070
 3060 CONTINUE
      J = 8
      GO TO 4010
 3070 CONTINUE
      J = 9
                  IF ( XPERT(JAN,IAN).LE.0.0 ) GO TO 4010
      J = 10
                  IF ( ILN . LE . 0 )           GO TO 4010
                  IF ( ILN . GT . NLANES(IAN) )GO TO 4010
      J = 11
                  IF ( .NOT. MAYENT(IAN,ILN) ) GO TO 4010
      J = 12
                  IF ( QTIM . GT . SIMTIM )     GO TO 4010
                  IF ( IPRTLO . NE . 0 )        IPRTLO = 1
 3080 CONTINUE
C-----SPECIAL VEHICLE#S LOGIN PARAMETERS ARE CORRECT NOW CHECK THE
C-----HEADWAY TO SEE IF VEHICLE MAY BE WRITTEN ONTO TAPE FOR SIMPRO
      HEAD = QTIM - QTLAST(IAN,ILN)
                  IF ( HEAD . LT . HMIN )       GO TO 4030
      QTLAST(IAN,ILN) = QTIM
C-----WRITE SPECIAL VEHICLE ONTO TAPE FOR SIMPRO
      WRITE (MODELT,501)  QTIM,IVC,IDC,IDV,JAP,IAP,ILN,IPRTLO
      LVTOT = LVTOT + LENV(IVC) + 4
      NSWRIT(IAN) = NSWRIT(IAN) + 1
      J = 13
 4010 CONTINUE
C-----PRINT SPECIAL VEHICLE AND ITS NOTE (POSSIBLY ERROR CODE) AND READ
C-----NEXT SPECIAL VEHICLE AND IF NO MORE GO TO 4020 AND SET IEOF FLAG
      NOTE(J) = 1
      CALL  GENDVH  ( ITABL,3 )
      PRINT 601 , QTIM,IVC,IDC,IDV,JAP,IAP,ILN,IPRTLO,J
      PRINT 601
      ISPLHD = .TRUE.
      NLINE = NLINE + 2
      READ (5,501,END=4020)  QTIM,IVC,IDC,IDV,JAP,IAP,ILN,IPRTLO
      J = 2
      NSREAD = NSREAD + 1
      GO TO 3010
C-----SET IEOF FLAG AND GO TO 5010 AND CHECK ON GENERATED VEHICLES TO BE
C-----WRITTEN ONTO TAPE FOR SIMPRO
 4020 CONTINUE
      IEOF = .TRUE.
      GO TO 5010
C-----RESET SPECIAL VEHICLE#S QUEUE-IN TIME TO HAVE HMIN SEC HEADWAY
 4030 CONTINUE
      QTIM = QTLAST(IAN,ILN) + HMIN
      J = 14
C-----GO TO 3010 AND CHECK FOR NEXT VEHICLE TO BE QUEUED IN
      GO TO 3010
C-----START OF GENERATION OF GENERATED VEHICLES LOGIN ATTRIBUTES
 5010 CONTINUE
C-----IF MINIMUM QUEUE-IN TIME IS VERY LARGE GO TO 6010 AND ENDFILE
C-----TAPE FOR SIMPRO
                  IF ( QMIN . GE . QBIG )   GO TO 6010
      IAN = KAN
      IA = LIBA(IAN)
C-----ATTRIBUTES ARE GENERATED UNDER DISCRETE MULTINOMIAL DISTRIBUTION
      I = 0
      CALL  DISCRT  ( XPERT(1,IAN),NOBA,JAN )
      JA = LOBA(JAN)
      ITURN = IITURN(JAN,IAN)
 5020 CONTINUE
      CALL  DISCRT  ( XPERLO(1,ITURN,IAN),NLANES(IAN),IL )
C-----CHECK HEADWAYS BETWEEN VEHICLES ON THE SAME APPROACH AND LANE SO
C-----THAT THEY ARE ARE MINIMUM OF HMIN SECONDS APART.  IF HMIN IS
C-----VIOLATED THEN TRY TO GENERATE AN ALTERNATE LANE (25 CHANCES)
      HEAD = QMIN - QTLAST(IAN,IL)
                  IF ( HEAD . GE . HMIN )      GO TO 5030
      I = I + 1
                  IF ( I . LT . 25 )           GO TO 5020
C-----GENERATED VEHICLE IS IGNORED (HEADWAY LESS THAN HMIN)
      NOTE(1) = 1
      CALL  GENDVH  ( ITABL,2 )
                  IF ( J . NE . 14 )           GO TO 5025
                  IF ( . NOT . ISPLHD )        GO TO 5025
      PRINT 601
      NLINE = NLINE + 1
 5025 CONTINUE
      PRINT 602 , QMIN,IA
      ISPLHD = .FALSE.
      NLINE = NLINE + 1
      INEXTV(IAN) = INEXTV(IAN) + 1
C-----GO TO 2010 AND CHECK TO FIND APPROACH WITH MINIMUM QUEUE-IN TIME
      GO TO 2010
 5030 CONTINUE
      SPERT(JAN,IAN) = SPERT(JAN,IAN) + 1.0
      SPERL(IL,IAN) = SPERL(IL,IAN) + 1.0
      CALL  DISCRT  ( XPERV(1,IAN),NVEHCL,IV )
      SPERV(IV,IAN) = SPERV(IV,IAN) + 1.0
      CALL  DISCRT  ( XPERD(1,IV),NDRICL,ID )
      SPERD(ID,IV) = SPERD(ID,IV) + 1.0
      IPLOGO = 0
                  IF ( IYESVL(IV) . EQ . NYES )IPLOGO = 1
                  IF ( IYESDL(ID) . EQ . NYES )IPLOGO = 1
C-----ARRIVING SPEED IS GENERATED UNDER NORMAL DISTRIBUTION AND MUST BE
C-----WITHIN ONE STANDARD DEVIATION OF APPROACH#S MEAN SPEED WITH A
C-----SLIGHT VARIATION TO ACCOUNT FOR DIFFERENT DRIVERS AND VEHICLES
 5040 CONTINUE
      CALL  NORMAL  ( VMEAN(IAN),VSIGMA(IAN),VEL )
              IF ( VEL . LT . VMIN(IAN,ID,IV) )   GO TO 5040
              IF ( VEL . GT . VMAX(IAN,ID,IV) )   GO TO 5040
      IVEL = VEL + 0.5
C-----WRITE GENERATED DRIVER-VEHICLE UNIT ONTO TAPE FOR SIMPRO
      WRITE (MODELT,501)  QMIN,IV,ID,IVEL,JA,IA,IL,IPLOGO
      LVTOT = LVTOT + LENV(IV) + 4
      QTLAST(IAN,IL) = QMIN
      INEXTV(IAN) = INEXTV(IAN) + 1
```

```
      NGWRIT(IAN) = NGWRIT(IAN) + 1
      GO TO 2010
C-----WRITE AN END OF FILE ONTO TAPE FOR SIMPRO
 6010 CONTINUE
      ENDFILE MODELT
      PRINT 601
      NLINE = NLINE + 1
      RETURN
      END                                                          GENDV
```

```
      SUBROUTINE GENDVH ( ITABL,I )
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      LOGICAL           ITABL
  601 FORMAT(8X,5HTABLE,I3,33H  -  EXPLANATION OF SPECIAL CASES,///,
     *       12X,48HQTIME  VEHICLE DRIVER VELOCITY OUTBOUND  INBOUND,
     *       19H  LANE  LOGOUT NOTE,/,
     *       20X,        40HCLASS  CLASS    (FPS)  APPROACH  APPROAC,
     *       13HH  NO.  PRINT,/)
                      IF ( ITABL )                 GO TO 1010
                      IF ( NLINE+9 . GT . LINES )  CALL  HEADER
C
C-----SUBROUTINE GENDVH PRINTS THE TABLE AND TABLE HEADINGS THE FIRST
C-----TIME IT IS CALLED AND FROM THEN ON ONLY CHECKS TO SEE IF A NEW
C-----PAGE HEADING IS NEEDED BEFORE PRINTING OUT A VEHICLE AND ITS NOTE
C
      ITABL = .TRUE.
      PRINT 601 , NTABL
      NLINE = NLINE + 6
      NTABL = NTABL + 1
      RETURN
 1010 CONTINUE
                      IF ( NLINE+I . GT . LINES )  CALL  HEADER
      RETURN
      END                                                          GENDVH
```

```
      SUBROUTINE DISCRT ( XPER,NUM,I )
      COMMON / ZTEMPD / GENDV(69),RANNUM,SUM
C
C-----SUBROUTINE DISCRT GENERATES A DISCRETE MULTINOMIAL RANDOM DEVIATE
C-----FOR A GIVEN PERCENTAGE ( 0.00 TO 100.0)
C
      DIMENSION          XPER(1)
      RANNUM = RANF(0)*100.0
      SUM = 0.0
      DO 1010  I = 1 , NUM
      SUM = SUM + XPER(I)
                      IF ( SUM . GE . RANNUM )     RETURN
 1010 CONTINUE
      I = NUM
      RETURN
      END                                                           DISCRT
```

```
      SUBROUTINE NORMAL ( VMEAN,VSIGMA,VEL )
      COMMON / ZTEMPD / GENDV(69),I,SUM
C
C-----SUBROUTINE NORMAL GENERATES NORMALLY DISTRIBUTED RANDOM DEVIATES
C
      SUM = 0.0
      DO 1010  I = 1 , 12
      SUM = SUM + RANF(0)
 1010 CONTINUE
      VEL = VMEAN + VSIGMA*(SUM-6.0)
      RETURN
      END                                                           NORMAL
```

```
      SUBROUTINE PNOTES
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      LOGICAL           IEOF,MAYENT
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / ZTEMPD(68),I,LTEST,MTEST
  601 FORMAT(12X,32HNOTE  EXPLANATION OF THE NOTE(S),/)
  602 FORMAT(14X,21H1   HEADWAY LESS THAN,F4.1,21H SECONDS FROM PREVIOU,
     *       55HS VEHICLE FOR THIS APPROACH AND ITS LANE(S) - GENERATED,
     *       16H VEHICLE IGNORED)
  603 FORMAT(14X,30H2   SPECIAL VEHICLE AS READ IN)
  604 FORMAT(14X,51H3   QUEUE-IN TIME LESS THAN ZERO - SPECIAL VEHICLE ,
     *       7HIGNORED)
  605 FORMAT(14X,51H4   VEHICLE CLASS INCORRECT - SPECIAL VEHICLE IGNOR,
     *       2HED)
  606 FORMAT(14X,51H5   DRIVER CLASS INCORRECT - SPECIAL VEHICLE IGNORE,
     *       1HD)
  607 FORMAT(14X,51H6   QUESTIONABLE DESIRED SPEED - SPECIAL VEHICLE IG,
     *       5HNORED)
  608 FORMAT(14X,51H7   LINKING OUTBOUND APPROACH NUMBER INCORRECT - SP,
     *       21HECIAL VEHICLE IGNORED)
  609 FORMAT(14X,51H8   INBOUND APPROACH NUMBER INCORRECT - SPECIAL VEH,
     *       12HICLE IGNORED)
  610 FORMAT(14X,51H9   QUESTIONABLE OUTBOUND APPROACH - SPECIAL VEHICL,
     *       9HE IGNORED)
  611 FORMAT(13X,51H10   LANE NUMBER INCORRECT - SPECIAL VEHICLE IGNORE,
     *       1HD)
  612 FORMAT(13X,49H11   LANE DOES NOT EXIST AT THE BEGINNING OF THE ,
     *       34HAPPROACH - SPECIAL VEHICLE IGNORED)
  613 FORMAT(13X,51H12   SPECIAL VEHICLE QUEUE-IN TIME IS GREATER THAN ,
     *       15HSIMULATION TIME)
  614 FORMAT(13X,32H13   SPECIAL VEHICLE AS INSERTED)
  615 FORMAT(13X,22H14   HEADWAY LESS THAN,F4.1,20H SECONDS FROM PREVIO,
     *       36HUS VEHICLE ON SAME APPROACH AND LANE,/,18X,9HSPECIAL V,
     *       27HEHICLE HEADWAY INCREASED TO,F4.1,8H SECONDS)
  616 FORMAT(/,1X,131(1H*),/,21X,30HNOTES 3 THRU 12 EACH INDICATE ,
     *       52HSPECIAL VEHICLE(S) IGNORED - CORRECT INPUT AND RERUN,
     *       11H IF DESIRED,/,1X,131(1H*))
  617 FORMAT(//)
C
C-----SUBROUTINE PNOTES PRINTS THE EXPLANATION OF THE NOTES ASSOCIATED
C-----WITH THE WRITING AND CHECKING OF DRIVER-VEHICLE UNITS ONTO A TAPE
C-----FOR SIMPRO
C
C-----COUNT UP NUMBER OF NOTE LINES TO BE PRINTED
      LTEST = 0
      DO 1010  I = 1 , 14
                  IF ( NOTE(I) . NE . 0 )      LTEST = LTEST + 1
 1010 CONTINUE
                  IF ( NOTE(14) . NE . 0 )     LTEST = LTEST + 1
C-----IF NOT ANY NOTES TO BE PRINTED GO TO 2010 AND RETURN
                  IF ( LTEST . EQ . 0 )        GO TO 2010
      MTEST = 0
C-----CHECK TO SEE IF ANY INCORRECT PARAMETERS WERE READ IN FOR SPECIAL
C-----VEHICLES AND PRINT 4 LINE WARNING
      DO 1020  I = 3 , 12
                  IF ( NOTE(I) . NE . 0 )      MTEST = 4
 1020 CONTINUE
      LTEST = LTEST + MTEST + 2
C-----PRINT ANY PERTINET NOTES
              IF ( NLINE+LTEST . GT . LINES )  CALL  HEADER
      PRINT 601
                  IF ( NOTE( 1) . NE . 0 )     PRINT 602 , HMIN
                  IF ( NOTE( 2) . NE . 0 )     PRINT 603
                  IF ( NOTE( 3) . NE . 0 )     PRINT 604
                  IF ( NOTE( 4) . NE . 0 )     PRINT 605
                  IF ( NOTE( 5) . NE . 0 )     PRINT 606
                  IF ( NOTE( 6) . NE . 0 )     PRINT 607
                  IF ( NOTE( 7) . NE . 0 )     PRINT 608
                  IF ( NOTE( 8) . NE . 0 )     PRINT 609
                  IF ( NOTE( 9) . NE . 0 )     PRINT 610
                  IF ( NOTE(10) . NE . 0 )     PRINT 611
                  IF ( NOTE(11) . NE . 0 )     PRINT 612
                  IF ( NOTE(12) . NE . 0 )     PRINT 613
                  IF ( NOTE(13) . NE . 0 )     PRINT 614
                  IF ( NOTE(14) . NE . 0 )     PRINT 615 , HMIN,HMIN
                  IF ( MTEST . NE . 0 )        PRINT 616
      NLINE = NLINE + LTEST
                  IF ( NLINE+3 . GT . LINES )  GO TO 2010
      PRINT 617
      NLINE = NLINE + 3
 2010 CONTINUE
      RETURN
      END                                                         PNOTES
```

```
      SUBROUTINE PSUMDV
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / DVDATA / FPERL,FPERR,HMIN,IEOF,MAYENT(6,6),QTIME(1000,6),
     *                  QTLAST(6,6),SIMTIM,XPERT(6,6),XPERV(15,6),ZERO
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / ZTEMPD / NGWRIT(6),NSWRIT(6),NSREAD,LVTOT,NGTOT,NGVOL,
     *                  NSTOT,NSVOL,NTOTAL,NTVOL,IVOLT,IDENSE,IAN,
     *                  NSELIM,ZTEMPD(47)
      DIMENSION         IREAD(2),IELIM(3),MSG902(8)
      DATA      IREAD  / 4HREAD,3H IN /
      DATA      IELIM  / 4HELIM,4HINAT,2HED /
      DATA      MSG902 / 4H TOT,4HAL V,4HOLUM,4HE FO,4HR AP,4HPROA,
     *                  4HCH L,3HE 0 /
  601 FORMAT(8X,5HTABLE,I3,27H  -  FINAL APPROACH VOLUMES,///,
     *         25X,17HSPECIAL  VEHICLES,7X,19HGENERATED  VEHICLES,8X,
     *         15HTOTAL  VEHICLES,/,23X,3(22(1H-),3X),/,12X,
     *         11HAPPROACH   ,3(25HNUMBER FOR  VOLUME FOR    ),6H INPUT,/,
     *    12X,11H NUMBER    ,3(25HSIMULATION  SIMULATION    ),6HVOLUME,/)
  602 FORMAT(15X,I2,3(8X,I4,9X,I4),7X,I4)
  603 FORMAT(/,13X,5HTOTAL,2X,3(4X,I5,8X,I5,3X),3X,I5,/)
  604 FORMAT(12X,I4,23H SPECIAL VEHICLES WERE ,3A4)
  605 FORMAT(12X,37HTHE INTERSECTION HAS A JAM DENSITY OF,I4,
     *         18H VEHICLES PER MILE)
  606 FORMAT(//)
  902 FORMAT(4A4,I3,16H HAS NO VEHICLES)
C
C-----SUBROUTINE PSUMDV PRINT THE SUMMARY STATISTICS OF THE VEHICLES
C-----ACTUALLY WRITTEN ONTO A TAPE FOR SIMPRO
C
                        IF ( NLINE+NIBA+10 . GT . LINES )CALL  HEADER
      PRINT 601 , NTABL
      NTABL = NTABL + 1
      NLINE = NLINE + 8
      IVOLT = 0
      NGTOT = 0
      NSTOT = 0
C-----START INBOUND APPROACH LOOP
      DO 1010  IAN = 1 , NIBA
      NSVOL = NSWRIT(IAN)*3600.0/SIMTIM + 0.5
      NGVOL = NGWRIT(IAN)*3600.0/SIMTIM + 0.5
      NTOTAL = NSWRIT(IAN) + NGWRIT(IAN)
      NTVOL = NTOTAL*3600.0/SIMTIM + 0.5
C-----PRINT STATISTICS FOR INBOUND APPROACH IAN
      PRINT 602 , LIBA(IAN),NSWRIT(IAN),NSVOL,NGWRIT(IAN),NGVOL,NTOTAL,
     *            NTVOL,IVOL(IAN)
C-----IF THERE WERE NOT ANY VEHICLES WRITTEN ONTO THE TAPE FOR SIMPRO
C-----FOR INBOUND APPROACH IAN THEN GO TO 9020 AND PRINT ERROR MESSAGE
      IF ( NGWRIT(IAN)+NSWRIT(IAN) . LE . 0 )    GO TO 9020
      NGTOT = NGTOT + NGWRIT(IAN)
      NSTOT = NSTOT + NSWRIT(IAN)
      IVOLT = IVOLT + IVOL(IAN)
 1010 CONTINUE
C-----CALCULATE TOTALS FOR THE INTERSECTION AND PRINT THE TOTALS
      NSVOL = NSTOT*3600.0/SIMTIM + 0.5
      NGVOL = NGTOT*3600.0/SIMTIM + 0.5
      NTOTAL = NGTOT + NSTOT
      NTVOL = NTOTAL*3600.0/SIMTIM + 0.5
      PRINT 603 , NSTOT,NSVOL,NGTOT,NGVOL,NTOTAL,NTVOL,IVOLT
      NLINE = NLINE + NIBA + 2
C-----IF NO SPECIAL VEHICLES THEN GO TO 1020 AND PRINT JAM DENSITY
      LTEST = 5
                        IF ( NSREAD . LE . 0 )       LTEST = 2
                        IF ( NLINE+LTEST.GT.LINES )  CALL  HEADER
                        IF ( NSREAD . LE . 0 )       GO TO 1020
      NSELIM = NSREAD - NSTOT
      PRINT 604 , NSREAD,IREAD
      PRINT 604 , NSELIM,IELIM
      PRINT 602
 1020 CONTINUE
C-----CALCULATE AND PRINT THE JAM DENSITY FOR THE INTERSECTION
      IDENSE = NTOTAL*5280.0/LVTOT + 0.5
      PRINT 605 , IDENSE
      NLINE = NLINE + LTEST
                        IF ( NLINE+3 . GT . LINES )  RETURN
      PRINT 606
      NLINE = NLINE + 3
      RETURN
C-----PROCESS EXECUTION ERROR AND STOP
 9020 CONTINUE
      PRINT 902 , IFORM,LIBA(IAN)
      CALL  ABORTR  ( MSG902,31,24 )
      STOP  902
      END                                                               PSUMDV
```

```
      SUBROUTINE PSTATS
      COMMON / APPRO  / IAAZIM(12),IDIST(6),IITURN(6,6),IVOL(6),
     *                  NDEGST(6),NLANES(6),NVA(6),PARAM(6),VMEAN(6),
     *                  VSIGMA(6),XPERLO(6,3,6)
      COMMON / CLASS  / IAMAX(15),IDCHAR(5),IDMAX(15),IRMIN(15),
     *                  IVCHAR(15),IVMAX(15),IYESD,IYESDL(5),IYESP,
     *                  IYESV,IYESVL(15),LENV(15),MAXV,NDRICL,NVEHCL,
     *                  PIJR(5),VMAX(6,5,15),VMIN(6,5,15),XPERD(5,15)
      COMMON / INTER  / LIBA(6),LOBA(6),NAP,NIBA,NOBA
      COMMON / OUTPUT / IFORM(4),LINES,MODELT,NLINE,NOTE(14),NPAGE,NTABL
      COMMON / STATS  / SPERD(5,15),SPERL(6,6),SPERT(6,6),SPERV(15,6)
      COMMON / ZTEMPD / NGWRIT(6),IA,IAN,ID,ILN,IV,JAN,MLANES,NUMV,SUM,
     *                  ZTEMPD(56)
  601 FORMAT(8X,5HTABLE,I3,29H  -  STATISTICS OF GENERATION,///,
     *       12X,19HAPPROACH STATISTICS,/,12X,19(1H-),/)
  602 FORMAT(12X,39HAPPROACH NUMBER ------------------------,I5)
  603 FORMAT(16X,35HOUTBOUND APPROACH NUMBER -----------,6I5)
  604 FORMAT(12X,39HPERCENT GOING TO OUTBOUND APPROACHES --,1X,6F5.1)
  605 FORMAT(16X,35HVEHICLE CLASS NUMBER --------------,15I5)
  606 FORMAT(12X,39HGENERATION PERCENT OF VEHICLES ---------,1X,15F5.1)
  607 FORMAT(12X,35HPERCENT OF TRAFFIC ENTERING ON LANE,I2,2H -,F6.1)
  608 FORMAT(1H+,57X,13H(MEDIAN LANE))
  609 FORMAT(1H+,57X,11H(CURB LANE))
  610 FORMAT(12X,29HDRIVER CLASS SPLIT STATISTICS,/,12X,29(1H-),//,
     *       12X,39HDRIVER CLASS NUMBER -------------------,05I6)
  611 FORMAT(12X,20HVEHICLE CLASS NUMBER,I3,2H (,I4,11H VEH) -----,
     *       5F6.1)
  612 FORMAT()
C
C-----SUBROUTINE PSTATS CALCULATES AND PRINTS BY EACH INBOUND APPROACH
C-----THE GENERATED PERCENTAGES FOR THE LOGIN ATTRIBUTES PREVIOUSLY
C-----SPECIFIED BY THE USER (OR DEFAULT VALUES)
C
C-----CHECK TO SEE IF THERE ARE ANY GENERATED VEHICLES TO COMPUTE
C-----STATISTICS OF GENERATION
      NUMV = 0
      DO 1000  IAN = 1 , NIBA
      NUMV = NUMV + NGWRIT(IAN)
 1000 CONTINUE
                      IF ( NUMV . LE . 0 )          RETURN
      IF ( NLINE+NLANES(1)+13 . GT . LINES )      CALL  HEADER
      PRINT 601 , NTABL
      NTABL = NTABL + 1
      NLINE = NLINE + 6
C-----PRINT APPROACH STATISTICS BY EACH INBOUND APPROACH
      DO 4010  IAN = 1 , NIBA
C-----IF NO GENERATED VEHICLES FOR THIS APPROACH GO TO 4010 AND PROCESS
C-----OTHER APPROACHES
                      IF ( NGWRIT(IAN) . LE . 0 )  GO TO 4010
      MLANES = NLANES(IAN)
            IF ( NLINE+MLANES+7 . GT . LINES )  CALL  HEADER
      PRINT 602 , LIBA(IAN)
C-----SUM UP NUMBER OF VEHICLES GOING TO EACH OUTBOUND APPROACH
      SUM = 0.0
      DO 1010  JAN = 1 , NOBA
      SUM = SUM + SPERT(JAN,IAN)
 1010 CONTINUE
C-----CALCULATE THE PERCENTAGE GOING TO EACH OUTBOUND APPROACH
      DO 1020  JAN = 1 , NOBA
      SPERT(JAN,IAN) = 100.0*SPERT(JAN,IAN)/SUM
 1020 CONTINUE
C-----PRINT THE PERCENTAGES GOING TO EACH OUTBOUND APPROACH
      PRINT 603 , (LOBA(JAN),JAN=1,NOBA)
      PRINT 604 , (SPERT(JAN,IAN),JAN=1,NOBA)
C-----SUM THE NUMBER OF VEHICLES OF EACH VEHICLE CLASS GENERATED
      SUM = 0.0
      DO 2010  IV = 1 , NVEHCL
      SUM = SUM + SPERV(IV,IAN)
 2010 CONTINUE
C-----CALCULATE THE PERCENTAGE
      DO 2020  IV = 1 , NVEHCL
      SPERV(IV,IAN) = 100.0*SPERV(IV,IAN)/SUM
 2020 CONTINUE
C-----PRINT THE PERCENTAGE OF EACH VEHICLE CLASS GENERATED
      PRINT 605 , (IV,IV=1,NVEHCL)
      PRINT 606 , (SPERV(IV,IAN),IV=1,NVEHCL)
C-----SUM THE NUMBER OF VEHICLES ENTERING ON EACH LANE
      SUM = 0.0
      DO 3010  ILN = 1 , MLANES
      SUM = SUM + SPERL(ILN,IAN)
 3010 CONTINUE
C-----CALCULATE AND PRINT THE PERCENTAGE OF VEHICLES ENTERING EACH LANE
      DO 3020  ILN = 1 , MLANES
      SPERL(ILN,IAN) = 100.0*SPERL(ILN,IAN)/SUM
      PRINT 607 , ILN,SPERL(ILN,IAN)
                      IF ( ILN . EQ . 1 )           PRINT 608
 3020 CONTINUE
                      IF ( MLANES . NE . 1 )        PRINT 609
      PRINT 612
      PRINT 612
      NLINE = NLINE + MLANES + 7
C-----END OF INBOUND APPROACH LOOP
 4010 CONTINUE
C-----PRINT DRIVER CLASS SPLIT STATISTICS (XPERD)
            IF ( NLINE+NVEHCL+5 . GT . LINES )   CALL  HEADER
      PRINT 610 , (ID,ID=1,NDRICL)
      PRINT 612
C-----SUM THE NUMBER OF VEHICLES GENERATED UNDER EACH VEHICLE AND DRIVER
C-----CLASS
      DO 6010  IV = 1 , NVEHCL
      SUM = 0.0
      DO 5010  ID = 1 , NDRICL
      SUM = SUM + SPERD(ID,IV)
 5010 CONTINUE
                      IF ( SUM . LE . 0.0 )         GO TO 5030
C-----CALCULATE THE PERCENTAGE OF DRIVER TYPES IN EACH VEHICLE CLASS
      DO 5020  ID = 1 , NDRICL
      SPERD(ID,IV) = 100.0*SPERD(ID,IV)/SUM
 5020 CONTINUE
 5030 CONTINUE
C-----PRINT PERCENTAGE OF DRIVER TYPES GENERATED FOR EACH VEHICLE CLASS
      NUMV = SUM
      PRINT 611 , IV,NUMV,(SPERD(ID,IV),ID=1,NDRICL)
 6010 CONTINUE
      NLINE = NLINE + NVEHCL + 5
                      IF ( NLINE+3 . GT . LINES )  RETURN
      PRINT 612
      PRINT 612
      PRINT 612
      NLINE = NLINE + 3
      RETURN
      END                                                              PSTATS
```

```
      SUBROUTINE ABORTR ( MSG,NCHS,NZTEMP )
      COMMON / ZTEMPD / ZTEMPD(71)
      DIMENSION          MSG(1),MSGPP(9),ITEMPD(71)
      EQUIVALENCE        (ZTEMPD(1),ITEMPD(1))
  601 FORMAT(20A4)
C¬602 FORMAT(8H ZTEMPD(,I2,11H)  OCTAL = ,O20,1HB,
C¬   *         9H  REAL = ,G20.10,12H  INTEGER = ,I15)
C;602 FORMAT(8H ZTEMPD(,I2,09H)  HEX = ,Z8,1HZ,
C;   *         9H  REAL = ,G20.10,12H  INTEGER = ,I15)
C
C-----SUBROUTINE ABORTR PROCESSES SYSTEM AND USER ERRORS
C
C¬    ASSIGN 2010 TO IRECAD
C¬    ASSIGN 101 TO JRECAD
C¬    CALL  XMIT     ( IRECAD )
      NWDS = (NCHS+3)/4
      PRINT 601
      PRINT 601 , (MSG(I),I=1,NWDS)
                     IF ( NZTEMP . LE . 0 )        GO TO 2010
      PRINT 601
      DO 1010  I = 1 , NZTEMP
      PRINT 602 , I,ZTEMPD(I),ZTEMPD(I),ITEMPD(I)
 1010 CONTINUE
 2010 CONTINUE
C¬    CALL  XMIT     ( JRECAD )
C¬    ICHS = NWDS*4
C¬          ENCODE ( ICHS,601,MSGPP )              (MSG(I),I=1,NWDS)
C¬    I = (NCHS+9)/10 + 1
C¬    MSGPP(I) = 0
C¬    CALL  REMARK  ( MSGPP )
C¬101 CONTINUE
C¬    CALL  XMIT     ( 0 )
      RETURN
C¬102 GO TO IRECAD                                                       *DEBUG*
C¬103 GO TO JRECAD                                                       *DEBUG*
      END                                                                ABORTR
```

PROGRAMMERS DOCUMENTATION

DRIVER-VEHICLE PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKAGE

LATEST UPDATE: 29 NOV 77

THIS DOCUMENTATION IS DIVIDED INTO THE FOLLOWING SECTIONS:

1. DRIVER-VEHICLE PROCESSOR LIMITATIONS
2. EXPLANATION OF THE INPUT ERRORS
3. EXPLANATION OF THE EXECUTION ERRORS
4. DEFINITION OF THE VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED
5. DEFINITION OF THE LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL
6. ALPHABETICAL LISTING OF ALL THE ROUTINES AND THE ROUTINES WHICH CAN CALL THEM
7. GENERALIZED CALLING SEQUENCE DIAGRAM
8. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED

1. DRIVER-VEHICLE PROCESSOR LIMITATIONS

MAXIMUM NUMBER OF INBOUND APPROACHES ----------------	6
MAXIMUM NUMBER OF OUTBOUND APPROACHES ---------------	6
RANGE OF APPROACH NUMBERS ----------------------------	1-12
MAXIMUM NUMBER OF LANES PER APPROACH ----------------	6
MAXIMUM NUMBER OF VEHICLES GENERATED PER APPROACH ---	1000
MAXIMUM NUMBER OF INBOUND LANES ---------------------	25
MAXIMUM NUMBER OF OUTBOUND LANES --------------------	25
MAXIMUM NUMBER OF DRIVER CLASSES --------------------	5
MAXIMUM NUMBER OF VEHICLE CLASSES -------------------	15
MAXIMUM DECELERATION RATE OF VEHICLES (UNIFORM) -----	-12 FT/SEC/SEC
MAXIMUM ACCELERATION RATE OF VEHICLES (UNIFORM) -----	18 FT/SEC/SEC

2. EXPLANATION OF INPUT ERRORS

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READIO:

```
STOP 801 = NUMBER OF INBOUND APPROACHES = <NIBA> IS LE 0 OR GT 6
           (NUMBER OF INBOUND APPROACHES OUT OF RANGE 1-6)
STOP 802 = INBOUND APPROACH <I> = <LIBA(I)> IS LE 0 OR GT 12
           (INBOUND APPROACH NUMBER OUT OF RANGE 1-12)
STOP 803 = INBOUND APPROACH <I> = <LIBA(I)> IS EQUAL TO INBOUND
           APPROACH <K> = <LIBA(K)>
           (APPROACH NUMBER IS ALREADY ON LIST OF INBOUND APPROACHES)
STOP 804 = NUMBER OF OUTBOUND APPROACHES = <NOBA> IS LE 0 OR GT 6
           (NUMBER OF OUTBOUND APPROACHES OUT OF RANGE 1-6)
STOP 805 = OUTBOUND APPROACH <I> = <LOBA(I)> IS LE 0 OR GT 12
           (OUTBOUND APPROACH NUMBER OUT OF RANGE 1-12)
STOP 806 = OUTBOUND APPROACH <I> = <LOBA(I)> IS EQUAL TO OUTBOUND
           APPROACH <K> = <LOBA(K)>
           (APPROACH NUMBER IS ALREADY ON LIST OF OUTBOUND APPROACHES)
STOP 807 = INBOUND APPROACH <I> = <LIBA(I)> IS EQUAL TO OUTBOUND
           APPROACH <J> = <LOBA(J)>
           (APPROACH NUMBER IS ON BOTH INBOUND AND OUTBOUND LISTS)
```

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READOP:

```
STOP 808 = NUMBER OF APPROACHES = <NAP> IS LE 0 OR GT 12
           (NUMBER OF APPROACHES IS OUT OF RANGE 1-12)
STOP 809 = NUMBER OF INBOUND APPROACHES PLUS NUMBER OF OUTBOUND APPROACHES =
           <NTEST> IS NE NUMBER OF APPROACHES <NAP>
STOP 810 = TIME FOR GENERATING VEHICLES = <ITSIM> IS LT 12 OR GT 65
           (TIME FOR GENERATING VEHICLES IS OUT OF RANGE 12-65)
STOP 811 = MINIMUM HEADWAY BETWEEN VEHICLES = <HMIN> IS GT 5.0
           (MINIMUM HEADWAY BETWEEN VEHICLES IS GREATER THAN 5)
STOP 812 = NUMBER OF VEHICLE CLASSES = <NVEHCL> IS LT 0 OR GT 15
           (NUMBER OF VEHICLE CLASSES IS OUT OF RANGE 0-15)
STOP 813 = NUMBER OF DRIVER CLASSES = <NDRICL> IS LT 0 OR GT 5
           (NUMBER OF DRIVER CLASSES IS OUT OF RANGE 0-5)
STOP 814 = PERCENT OF LEFT TURNS IN MEDIAN LANE = <FPERL> IS LT 50.0
           OR GT 100.0
           (PERCENT OF LEFT TURNS IN MEDIAN LANE IS OUT OF RANGE 50-100)
STOP 815 = PERCENT OF RIGHT TURNS IN CURB LANE = <FPERR> IS LT 50.0
           OR GT 100.0
           (PERCENT OF RIGHT TURNS IN CURB LANE OUT OF RANGE 50-100)
```

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READAP:

```
STOP 816 = APPROACH NUMBER <J> IS LE 0 OR GT 12
           (APPROACH NUMBER OUT OF RANGE 1-12)
STOP 817 = APPROACH NUMBER <J> IS USED MORE THAN ONCE
           (APPROACH DATA ALREADY ENTERED FOR THIS APPROACH)
STOP 818 = APPROACH NUMBER <J> AZIMUTH = <IAAZIM> IS LT 0 OR GT 360
           (APPROACH AZIMUTH OUT OF RANGE 0-360)
STOP 819 = APPROACH NUMBER <J> NUMBER OF LANES = <NLANES> IS LE 0 OR GT 6
           (APPROACH NUMBER OF LANES IS OUT OF RANGE 1-6)
STOP 820 = APPROACH NUMBER <J> IS NOT ON INBOUND OR OUTBOUND LISTS
           (APPROACH DATA SPECIFIED FOR AN APPROACH THAT IS NOT ON THE
           INBOUND OR OUTBOUND LISTS)
STOP 821 = APPROACH NUMBER <J> NUMBER OF DEGREES FOR STRAIGHT = <NDEGST>
           IS LT 0 OR GT 45
           (NUMBER OF DEGREES FOR STRAIGHT IS OUT OF RANGE 0-45)
STOP 822 = APPROACH NUMBER <IA> HEADWAY DISTRIBUTION NAME = <JDIST,KDIST>
           IS NOT (CONSTAN)OR(ERLANG )OR(GAMMA   )OR(LOGNRML)OR(NEGEXP )
           OR(SNEGEXP)OR(UNIFORM)
STOP 823 = APPROACH NUMBER <IA> HAS ZERO VOLUME WITH A VALID DISTRIBUTION NAME
STOP 824 = APPROACH NUMBER <IA> PARAMETER FOR DISTRIBUTION = <PDIST> IS LE 0.0
           (APPROACH PARAMETER FOR DISTRIBUTION IS LESS THAN 0)
STOP 825 = APPROACH NUMBER <IA> PARAMETER FOR ERLANG DISTRIBUTION IS NOT AN
           INTEGER VALUE
STOP 826 = APPROACH NUMBER <IA> PARAMETER FOR GAMMA DISTRIBUTION = <PDIST>
           IS LT 1.0
STOP 827 = APPROACH NUMBER <IA> PARAMETER FOR SHIFTED NEGATIVE EXPONENTIAL
           DISTRIBUTION = <PDIST> IS GE MEAN HEADWAY = <THEAN>
STOP 828 = APPROACH NUMBER <IA> EQUIVALENT HOURLY VOLUME = <JVOL> IS
           LT 0 OR GT 4000
           (APPROACH EQUIVALENT HORLY VOLUME IS OUT OF RANGE 0-4000)
STOP 829 = APPROACH NUMBER <IA> APPROACH MEAN SPEED = <XMEANS> IS LE
           10.0 OR GT 80.0
           (APPROACH MEAN SPEED IS OUT OF RANGE 10-80)
STOP 830 = APPROACH NUMBER <IA> APPROACH 85 PERCENTILE SPEED = <X85PER>
           IS LT APPROACH MEAN SPEED = <XMEANS> OR GT 90.0
           (APPROACH 85 PERCENTILE SPEED IS OUT OF RANGE [APPROACH MEAN
            SPEED-90])
STOP 831 = APPROACH NUMBER <IA> APPROACH TURNING PERCENTAGES SUM = <SUM>
           IS NOT 100.0
STOP 832 = APPROACH NUMBER <IA> USER SUPPLIED PERCENT OF VEHICLES OPTION
           = <IYES> IS NOT (YES) OR (NO )
STOP 833 = APPROACH NUMBER <IA> NUMBER OF VEHICLE CLASSES = <NVEHCL>
           IS NOT 10 WHEN ASKING FOR PROGRAM SUPPLIED PERCENT OF
           VEHICLES IN TRAFFIC STREAM
STOP 834 = APPROACH NUMBER <IA> USER SUPPLIED PERCENT OF VEHICLES
           MAKING UP THE TRAFFIC STREAM SUM = <STREAM SUM> IS NOT 100.0
STOP 835 = APPROACH NUMBER <IA> LANE <J> DOES NOT START AT THE SAME LGEOM(1) AS
           THE FIRST LANE (<LGEOM1>)
           (ALL LANES FOR ONE APPROACH MUST START AT THE SAME PLACE SO THAT THE
           HEADWAY DISTRIBUTIONS ARE CORRECT)
STOP 836 = APPROACH NUMBER <IA> HAS VEHICLES ENTERING ON LANE NUMBER
           <J> THAT DOES NOT EXIST AT THE BEGINING OF THE APPROACH
STOP 837 = APPROACH NUMBER <IA> PERCENT OF VEHICLES IN EACH LANE SUM
           = <SUM> IS NOT 100.0
STOP 838 = APPROACH NUMBER <IA> HAS A MEAN SPEED = <XMEANS> AND A 85 PERCENTILE
           SPEED = <X85PER> WHICH GIVES ONE STANDARD DEVIATION = <VSIGMA> WHICH
           IS GREATER THAN THE MEAN
STOP 839 = APPROACH NUMBER <IA> IS ON OUTBOUND LIST YET HAS INBOUND DATA
           (APPROACH IS ON OUTBOUND LIST YET HAS A HEADWAY DISTRIBUTION)
STOP 840 = APPROACH NUMBER <IA> IS ON OUTBOUND LIST YET HAS PERCENT OF
           EACH VEHICLE CLASS MAKING THE TRAFFIC STREAM
STOP 841 = INFORMATION FOR APPROACH <IA> IS NOT SPECIFIED
```

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE READGP:

```
STOP 842 = NUMBER OF ARCS = <NARCS> IS LT 0 OR GT 20
           (NUMBER OF ARCS IS OUT OF RANGE 0-20)
STOP 843 = NUMBER OF LINES = <NLINES> IS LT 0 OR GT 100
           (NUMBER OF LINES IS OUT OF RANGE 0-100)
STOP 844 = NUMBER OF SIGHT DISTANCE RESATICTIONS = <NSDRC> IS LT 0 OR GT 20
           (NUMBER OF SIGHT DISTANCE RESTRICTIONS IS OUT OF RANGE 0-20)
```

THE FOLLOWING ERRORS ARE DETECTED IN SUBROUTINE READYO:

```
STOP 845 = USER SUPPLY DRIVER CLASS SPLIT = <IYESP> IS NOT (YES) OR (NO )
           (USER SUPPLY DRIVER CLASS SPLIT IS NOT (YES) OR (NO ) )
STOP 846 = USER SUPPLY VEHICLE CLASS CHARACTERISTICS = <IYESV> IS NOT
           (YES) OR (NO )
           (USER SUPPLY VEHICLE CLASS CHARACTERISTICS IS NOT (YES) OR (NO ) )
STOP 847 = USER SUPPLY DRIVER CLASS CHARACTERISTICS = <IYESD> IS NOT
           (YES) OR (NO )
           (USER SUPPLY DRIVER CLASS CHARACTERISTICS IS NOT (YES) OR (NO ) )
STOP 848 = VEHICLE LOGOUT SUMMARY REQUESTED = <IYESVL> IS NOT (YES) OR (NO )
           (VEHICLE LOGOUT SUMMARY REQUESTED IS NOT (YES) OR (NO ) )
STOP 849 = DRIVER LOGOUT SUMMARY REQUESTED = <IYESDL> IS NOT (YES) OR (NO )
           (DRIVER LOGOUT SUMMARY REQUESTED IS NOT (YES) OR (NO ) )
STOP 850 = NUMBER OF VEHICLE CLASSES = <NVEHCL> IS NOT 10 WHEN DEFAULT
           DRIVER CLASS SPLITS ARE REQUESTED
STOP 851 = NUMBER OF DRIVER CLASSES = <NDRICL> IS NOT 3 WHEN DEFAULT
           DRIVER CLASS SPLITS ARE REQUESTED
STOP 852 = DRIVER CLASS SPLITS FOR VEHICLE CLASS SUM = <SUM> IS NOT 100.0
           (SUM OF DRIVER CLASS SPLITS FOR VEHICLE CLASS IS NOT 100)
STOP 853 = NUMBER OF VEHICLE CLASSES = <NVEHCL> IS NOT 10 WHEN DEFAULT
           VEHICLE CHARACTERISTICS ARE REQUESTED
STOP 854 = LENGTH OF VEHICLE CLASS = <LENV> IS LT 5 OR GT 100
           (LENGTH OF VEHICLE CLASS OUT OF RANGE 5-100)
STOP 855 = DRIVER FACTOR OF VEHICLE CLASS = <IDCHAR> IS LT 50 OR GT 150
```

```
         (DRIVER FACTOR OF VEHICLE CLASS IS OUT OF RANGE 50#150)
STOP 856 = DECELERATION MAXIMUM OF VEHICLE CLASS = <IDMAX> IS LT 4 OR GT 12
         (DECELERATION MAXIMUM OF VEHICLE CLASS OUT OF RANGE 4#12)
STOP 857 = ACCELERATION MAXIMUM OF VEHICLE CLASS = <IAMAX> IS LT 3 OR GT 18
         (ACCELERATION MAXIMUM OF VEHICLE CLASS OUT OF RANGE 3#18)
STOP 858 = VELOCITY MAXIMUM OF VEHICLE CLASS = <IVMAX> IS LT 10 OR GT 235
         (VELOCITY MAXIMUM OF VEHICLE CLASS OUT OF RANGE 10#235)
STOP 859 = MINIMUM TURNING RADIUS OF VEHICLE CLASS = <IRMIN> IS LT 4 OR GT 300
         (MINIMUM TURNING RADIUS OF VEHICLE CLASS IS OUT OF RANGE 4#300)
STOP 860 = NUMBER OF DRIVER CLASSES = <NDRICL> IS NOT 3 WHEN DEFAULT
         DRIVER CHARACTERISTICS ARE REQUESTED
STOP 861 = DRIVER FACTOR OF DRIVER CLASS = <IDCHAR> IS LT 50 OR GT 150
         (DRIVER FACTOR OF DRIVER CLASS IS OUT OF RANGE 50#150)
STOP 862 = PIJR TIME OF DRIVER CLASS <PIJR> IS LT 0.25 OR GT 5.00
         (PIJR TIME OF DRIVER CLASS IS OUT OF RANGE 0.25#5.00)
```

3. EXPLANATION OF EXECUTION ERRORS

```
STOP 901 IN GENHED = APPROACH <LIBA> HAS MORE THAN 1000 VEHICLES
                     (NO MORE VEHICLES CAN BE GENERATED ON THIS APPROACH)
STOP 902 IN PSUMDV = APPROACH NUMBER <IA> HAS NO VEHICLES
                     (NO VEHICLES WERE GENERATED FOR THIS APPROACH AND NO
                     SPECIAL VEHICLES WERE ENTERED FOR THIS APPROACH)
```

4. DEFINITION OF VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED

```
COMMON / APPRO /  APPROACH INFORMATION
                      BLKDAT READAP WRITDV BIASLT GENHED GENDV  PSTATS

   IAAZIM(12)     AZIMUTH OF APPROACH [0#360]
   IDIST(6)       DISTRIBUTION NUMBER FOR APPROACH [1,7]
   ITTURN(6,6)    TURN CODE BETWEEN EACH INBOUND AND OUTBOUND APPROACH [1-3]
                     1 = U-TURN OR LEFT
                     2 = STRAIGHT
                     3 = RIGHT
   IVOL(6)        VOLUME TO GENERATE FOR EACH APPROACH [0,4000]
   NDEGST(6)      NUMBER OF DEGREES LEFT OR RIGHT OF STRAIGHT FOR PATH TO
                  BE CONSIDERED STRAIGHT (DEFAULT IS 20) [0#45]
   NLANES(6)      NUMBER OF LANES FOR EACH APPROACH [1#6]
   NVA(6)         NUMBER OF VEHICLES GENERATED FOR EACH APPROACH [0-1000]
   PARAM(6)       DISTRIBUTION PARAMETER FOR EACH APPROACH
   VMEAN(6)       MEAN SPEED FOR EACH APPROACH [10-80 MPH]
   VSIGMA(6)      STANDARD DEVIATION OF SPEEDS FOR NORMAL DISTRIBUTION
   XPERLO(6,3,6)  PERCENT OF APPROACH VOLUME ENTERING BY AN INBOUND LANE FOR
                  A CERTAIN TURN CODE (INBOUND LANE,TURN CODE,INBOUND APPROACH)


COMMON / CLASS /  DRIVER AND VEHICLE PERFORMANCE VALUES
                      BLKDAT READDP READAP READYO WRITDV GENDV  PSTATS

   IAMAX(15)      MAXIMUM UNIFORM ACCELERATION RATE FOR EACH VEHICLE CLASS
                  (FT/SEC/SEC)
   IDCHAR(5)      DRIVER CHARACTERISTIC FOR EACH DRIVER CLASS
                  (SLOW <100,AVERAGE =100,AGGRESSIVE >100)
   IDMAX(15)      MAXIMUM UNIFORM DECELERATION RATE FOR EACH VEHICLE CLASS
                  (FT/SEC/SEC)
   IRMIN(15)      MINIMUM TURNING RADIUS FOR EACH VEHICLE CLASS (FEET)
   IVCHAR(15)     VEHICLE CHARACTERISTIC FOR EACH VEHICLE CLASS
                  (SLOW <100,AVERAGE =100,AGGRESSIVE >100)
   IVMAX(15)      MAXIMUM VELOCITY FOR EACH VEHICLE CLASS (FT/SEC)
   IYESD          CHANGE DEFAULT DRIVER ATTRIBUTES [YES/NO]
   IYSEDL(5)      DRIVER CLASS PRINT ON LOGOUT FROM SIMPRO [YES/NO]
   IYESP          CHANGE PERCENTAGE OF DRIVER CLASS FOR VEHICLE CLASS [YES/NO]
                  CLASS [YES/NO]
   IYESV          CHANGE DEFAULT VEHICLE ATTRIBUTES [YES/NO]
   IYESVL(15)     VEHICLE CLASS PRINT ON LOGOUT FROM SIMPRO [YES/NO]
   LENV(15)       LENGTH OF VEHICLE FOR EACH VEHICLE CLASS (FEET)
   MAXV           MAXIMUM VELOCITY VEHICLES CAN ENTER ON AN APPROACH
   NDRICL         NUMBER OF DRIVER CLASSES (DEFAULT=3) [1-5]
   NVEHCL         NUMBER OF VEHICLE CLASSES (DEFAULT=10) [1-15]
   PIJR(5)        PERCEPTION-REACTION TIME FOR EACH DRIVER CLASS (SECONDS)
                  [0.25-5.0]
   VMAX(6,5,15)   MAXIMUM VELOCITY FOR EACH INBOUND APPROACH, DRIVER
                  CLASS, AND VEHICLE CLASS (FT/SEC)
   VMIN(6,5,15)   MINIMUM VELOCITY FOR EACH INBOUND APPROACH, DRIVER
                  CLASS, AND VEHICLE CLASS (FT/SEC)
   XPERD(5,15)    PERCENTAGE OF DRIVER CLASSES IN EACH VEHICLE
                  CLASS (IDRICL,IVEHCL)


COMMON / DVDATA / DRIVER-VEHICLE PROCESSOR DATA
                      BLKDAT READDP READAP READYO WRITDV BIASLT GENHED CONST
                      ERLANG GAMMA  LGNRML NEGEXP SNEGEX UNIFRM GENDV  PNOTES

   FPERL          PERCENTAGE OF LEFT TURNING VEHICLES TO BE IN LEFT MOST LANE
   FPERR          PERCENTAGE OF RIGHT TURNING VEHICLES TO BE IN RIGHT MOST LANE
   HMIN           MINIMUM HEADWAY
   IEOF           STORES WORD FOR END-OF-FILE ON INPUT
                        0 = NO EOF
                        1 = EOF
   MAYENT(6,6)    MAY VEHICLES ENTER LANE AT START (T/F)
   QTIME(1000,6)  QUEUE-IN TIME FOR EACH VEHICLE (VEHICLE, INBOUND APPROACH)
```

```
QTLAST(6,6)     QUEUE-IN TIME FOR LAST VEHICLE WRITTEN TO TAPE (INBOUND
                APPROACH, INBOUND LANE)
SIMTIM          TIME FOR GENERATING VEHICLES
XPERT(6,6)      TURNING PERCENTAGES (OUTBOUND APPROACH, INBOUND APPROACH)
XPERV(15,6)     PERCENT OF EACH VEHICLE CLASS MAKING UP AN APPROACH
                TRAFFIC STREAM (VEHICLE CLASS, INBOUND APPROACH)
ZERO            VALUE OF A SMALL NUMBER ASSUMED TO BE ZERO


COMMON / INTER / DATA ABOUT INTERSECTION
                    BLKDAT READIO READOP READAP WRITDV BIASLT GENHED GENDV
                    PSTATS

LIBA(6)         LIST OF ENTRY NUMBERS FOR INBOUND APPROACHES [1→12]
LOBA(6)         LIST OF ENTRY NUMBERS FOR OUTBOUND APPROACHES [1→12]
NAP             TOTAL NUMBER OF APPROACHES IN THE INTERSECTION [1→12]
NIBA            NUMBER OF INBOUND APPROACHES [1→6]
NOBA            NUMBER OF OUTBOUND APPROACHES [1→6]


COMMON / LITCON / LITERAL AND CONSTANT DATA
                    BLKDAT READIN READAP READYO GENHED UNIFRM GENDV

FPSMPH          VALUE TO CONVERT MPH TO FPS
IDISTN(2,7)     HEADWAY DISTRIBUTION NAME
                     (1→2,1) = CONSTANT
                           2 = ERLANG
                           3 = GAMMA
                           4 = LOGNORMAL
                           5 = NEGATIVE EXPONENTIAL
                           6 = SHIFTED NEGATIVE EXPONENTIAL
                           7 = UNIFORM
NBLANK          (    )
NNO             (NO  )
NYES            (YES )
SQRT3           SQUARE ROOT OF 3.0


COMMON / OUTPUT / OUTPUT DATA
                    BLKDAT HEADER READIO READOP READAP READYO WRITDV GENHED
                    GENDV  GENDVH PNOTES PSTATS

LINES           NUMBER OF LINES PER PAGE
MODELT          TAPE NUMBER FOR SIMULATION MODEL
NLINE           NUMBER OF LINES PRINTED ON CURRENT PAGE
NOTE(14)        STATUS WORDS FOR NOTES TO BE PRINTED
                    0 = NO
                    1 = YES
NPAGE           PAGE NUMBER
NTABL           TABLE NUMBER


COMMON / STATS / STATISTICS OF GENERATION
                BLKDAT GENDV  PSTATS

SPERD(5,15)     PERCENT OF DRIVER CLASS IN EACH VEHICLE CLASS (DRIVER
                CLASS, VEHICLE CLASS)
SPERL(6,6)      PERCENT OF VEHICLES ENTERING BY A LANE FROM AN INBOUND
                APPROACH (INBOUND LANE, INBOUND APPROACH)
SPERT(6,6)      TURNING PERCENTAGES (OUTBOUND APPROACH, INBOUND APPROACH)
SPERV(15,6)     PERCENT OF EACH VEHICLE CLASS MAKING UP AN APPROACH TRAFFIC
                STREAM (VEHICLE CLASS, INBOUND APPROACH)


COMMON / TITLE / TITLE FOR DVPRO RUN
                    BLKDAT READIN HEADER WRITDV

ITITLE(20)      80 CHARACTER TITLE FOR DVPRO RUN


COMMON / ZTEMPD / TEMPORARY DATA
                    BLKDAT READIO READOP READAP READGP READYO WRITDV BIASLT
                    GENHED CONST  ERLANG GAMMA  LGNRML NEGEXP SNEGEX UNIFRM
                    GENDV  DISCRT NORMAL PNOTES PSTATS ABORTR

NVAIAN          NVA(IAN) NUMBER OF VEHICLES GENERATED FOR AN APPROACH
PARIAN          PARAM(IAN) HEADWAY DISTRIBUTION PARAMETER FOR AN APPROACH
TMEAN           MEAN HEADWAY FOR AN APPROACH
XPERL(6,6)      PERCENT OF VEHICLES ENTERING BY A LANE FROM AN INBOUND
                APPROACH (INBOUND LANE, INBOUND APPROACH)
ZTEMPD(71)      TEMPORARY DATA
```

5. DEFINITION OF LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL

VARIABLES THAT ARE LOCAL WITHIN SUBROUTINES ARE LISTED BELOW, EXCEPT FOR MOST DO-LOOP INDICES

SUBROUTINE ABORTR PROCESSES SYSTEM AND USER ERRORS
(CALLED FROM DVPRO GENHED)
(CALLS XMIT)

IRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
ITEMPD	INTEGER EQUIVALENCE TO ZTEMPD
JRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
MSG	ERROR MESSAGE PRINTED
MSGPP	ERROR MESSAGE FOR REMARK (CDC ONLY)
NCHS	NUMBER OF CHARACTERS IN ERROR MESSAGE
NWDS	NUMBER OF WORDS FOR ERROR MESSAGE MSG
NZTEMP	NUMBER OF ZTEMPD VARIABLES TO BE PRINTED
ZTEMPD	TEMPORARY (LOCAL) STORAGE OF VARIABLES FOR SUBROUTINES

SUBROUTINE BIASLT BIASES LANE ENTRY BY TURNING CODE
(CALLED FROM DVPRO)

FPER	PERCENTAGE OF TURNING MOVEMENTS TO BE IN CORRECT LANE
IA	NUMBER OF INBOUND APPROACH BEING PROCESSED [1#12]
IAN	INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1#6]
IANGLE	ANGLE BETWEN INBOUND AND OUTBOUND APPROACH
IAZIM	AZIMUTH OF INBOUND APPROACH [Ø#360]
ILN	INDEX NUMBER FOR NLANES ARRAY OF /APPRO/ OF INBOUND LANE BEING PROCESSED [1#6]
ITURN	TURN CODE BETWEEN INBOUND AND OUTBOUND APPROACHES
JA	NUMBER OF OUTBOUND APPROACH BEING PROCESSED [1#12]
JAN	INDEX NUMBER FOR LOBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1#6]
JAZIM	AZIMUTH OF OUTBOUND APPROACH [Ø#360]
JLN	BACKWARD INDEX NUMBER FOR NLANES ARRAY OF /APPRO/ OF INBOUND LANE BEING PROCESSED [1#6]
MDEGST	NUMBER OF DEGREES LEFT OR RIGHT OF TRUE STRAIGHT TO BE CONSIDERED STRAIGHT-THRU
NL	NUMBER OF LANES FOR APPROACH BEING PROCESSED [1#6]
SUM	SUMMATION OF TURNING PERCENTAGES AND LANE OCCUPANCY
XPERL(6,6)	PERCENT OF VEHICLES GOING FROM INBOUND TO OUTBOUND APPROACHES
XPERT(3,6)	PERCENT OF VEHICLES FOR EACH APPROACH MAKING ONE OF THREE TURNING MOVEMENTS

SUBROUTINE CONST GENERATES CONSTANT HEADWAYS
(CALLED FROM GENHED)

QTIMS(1)	ARRAY FOR QUEUE-IN TIME

SUBROUTINE DISCRT GENERATES A DISCRETE RANDOM VARIATE
(CALLED FROM GENDV)
(CALLS RANF)

I	GENERATED CLASS NUMBER
NUM	NUMBER OF CLASSES
RANNUM	RANDOM NUMBER [Ø#1ØØ]
SUM	CUMULATIVE SUM OF PERCENTAGES
XPER(1)	ARRAY OF PERCENTAGES OF OCCURRENCES FOR CLASS

SUBROUTINE ERLANG GENERATES ERLANG HEADWAYS
(CALLED FROM GENHED)
(CALLS RANF)

ALPHA	K/MEAN
K	K PARAMETER FOR ERLANG DISTRIBUTION
QTIMS(1)	ARRAY FOR QUEUE-IN TIME
THEAD	RANDOM ERLANG DISTRIBUTION
TR	PRODUCT OF K RANDOM NUMBERS

SUBROUTINE GAMMA GENERATES GAMMA HEADWAYS
(CALLED FROM GENHED)
(CALLS RANF)

A	A PARAMETER FOR GAMMA DISTRIBUTION
ALPHA	ALPHA PARAMETER FOR GAMMA DISTRIBUTION
K	NUMBER OF RANDOM NUMBERS TO BE DRAWN FOR A GIVEN VARIATE
K1	INTEGER ROUNDED-DOWN VALUE OF A PARAMETER
K2	INTEGER ROUNDED-UP VALUE OF A PARAMETER
Q	FRACTIONAL PORTION OF A ABOVE K1
QTIMS(1)	ARRAY FOR QUEUE-IN TIME
THEAD	RANDOM GAMMA HEADWAY
TR	PRODUCT OF K RANDOM NUMBERS

SUBROUTINE GENDV GENERATES DRIVER-VEHICLE UNITS
(CALLED FROM DVPRO)
(CALLS GENDVH NORMAL DISCRT)

HEAD	HEADWAY BETWEEN THE LAST VEHICLE ON THIS LANE AND THE NEXT VEHICLE WAITING TO ENTER THIS LANE
IA	NUMBER OF INBOUND APPROACH BEING PROCESSED [1#12]
IAN	INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1#6]
IAP	SPECIAL VEHICLE INBOUND APPROACH NUMBER
ID	GENERATED DRIVER CLASS NUMBER
IDC	SPECIAL VEHICLE DRIVER CLASS NUMBER
IDV	SPECIAL VEHICLE DESIRED VELOCITY
IL	NUMBER OF LANE BEING PROCESSED [1#6]
ILN	INDEX NUMBER FOR NLANES ARRAY OF /APPRO/ OF INBOUND LANE BEING PROCESSED [1#6]
INEXTV(6)	NEXT VEHICLE TO ENTER FOR EACH APPROACH
IPLOGO	PRINT ON LOGOUT FROM SIMPRO FOR GENERATED VEHICLE
IPRTLO	PRINT ON LOGOUT FROM SIMPRO FOR SPECIAL VEHICLE
ISPLHD	TRUE FOR SPECIAL VEHICLE WAS LAST PRINTED
ITABL	TRUE FOR SPECIAL VEHICLE HEADER WAS PRINTED
ITURN	TURN CODE FOR GENERATED VEHICLE
IV	VEHICLE CLASS NUMBER FOR GENERATED VEHICLES
IVC	VEHICLE CLASS NUMBER FOR SPECIAL VEHICLES
IVEL	INTEGER VALUE FOR DESIRED VELOCITY OF GENERATED VEHICLES
J	SPECIAL VEHICLE NOTE NUMBER
JA	NUMBER OF OUTBOUND APPROACH BEING PROCESSED [1#12]
JAN	INDEX NUMBER FOR LOBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1#6]
JAP	OUTBOUND APPROACH NUMBER FOR SPECIAL VEHICLE
KAN	APPPOACH NUMBER WITH LOWEST QUEUE-IN TIME
LVTOT	TOTAL LENGTH OF ALL VEHICLES WRITTEN ONTO TAPE
NGWRIT(6)	NUMBER OF GENERATED VEHICLES WRITTEN PER APPROACH
NSREAD	NUMBER OF SPECIAL VEHICLES READ
NSWRIT(6)	NUMBER OF SPECIAL VEHICLES WRITTEN PER APPROACH
QBIG	LARGE NUMBER (1.ØØE75) FOR CHECKING QUEUE-IN TIME
QMIN	MINIMUM QUEUE-IN TIME OF GENERATED VEHICLES
QTIM	QUEUE-IN TIME FOR SPECIAL VEHICLES
VEL	GENERATED VELOCITY FOR ENTERING

SUBROUTINE GENDVH PRINTS THE TABLE AND TABLE HEADING
(CALLED FROM GENDV)
(CALLS HEADER)

I	NUMBER OF LINES TO BE PRINTED
ITABL	STATUS OF WHETHER TABLE HEADER HAS BEEN PRINTED YET (T/F)

```
SUBROUTINE GENHED GENERATES APPROACH HEADWAYS
                  (CALLED FROM DVPRO)
                  (CALLS CONST  LGNRML SNEGEX RANF   ABORTR HEADER
                         NEGEXP UNIFRM GAMMA  ERLANG)

   IAN            INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING
                  PROCESSED [1-6]
   IDNUM          IDIST(IAN)
   ISUMIV         SUMMATION OF INPUT VOLUMES
   ISUMNG         SUMMATION OF NUMBER OF VEHICLES GENERATED
   ISUMVG         SUMMATION OF VOLUMES GENERATED
   IVOLGN         VOLUME GENERATED FOR EACH APPROACH
   IVOLIA         IVOL(IAN)
   PERDIF         PERCENT DIFFERENCE BETWEEN INPUT AND GENERATED VOLUMES


SUBROUTINE HEADER PRINTS THE HEADER MESSAGE
                  (CALLED FROM GENDVH GENHED PNOTES PSTATS PSUMDV
                               READAP READIN READIO READOP READYO)


SUBROUTINE LGNRML GENERATES LOG NORMAL HEADWAYS
                  (CALLED FROM GENHED)
                  (CALLS RANF)

   EX             THEAN (EXPECTED VALUE OF X VARIATE)
   EY             EXPECTED VALUE OF Y VARIATE
   QTIMS(1)       ARRAY FOR QUEUE=IN TIME
   STDX           STANDARD DEVIATION OF X VARIATE
   STDY           STANDARD DEVIATION OF Y VARIATE
   SUM            SUMMATION OF 12 RANDOM NUMBERS
   THEAD          RANDOM LOG NORMAL HEADWAY
   VARY           VARIANCE OF Y VARIATE


SUBROUTINE NEGEXP GENERATES NEGATIVE EXPONENTIAL HEADWAYS
                  (CALLED FROM GENHED)
                  (CALLS RANF)

   QTIMS(1)       ARRAY FOR QUEUE=IN TIME
   THEAD          RANDOM NEGATIVE EXPONENTIAL HEADWAY


SUBROUTINE NORMAL GENERATES NORMAL DEVIATES FOR DESIRED VELOCITY
                  (CALLED FROM GENDV)
                  (CALLS RANF)

   SUM            SUMMATION OF 12 RANDOM NUMBERS
   VEL            RANDOM GENERATED VELOCITY
   VMEAN          MEAN SPEED
   VSIGMA         STANDARD DEVIATION OF SPEED


SUBROUTINE PNOTES PRINTS THE EXPLANATION OF NOTES
                  (CALLED FROM DVPRO)
                  (CALLS HEADER)

   LTEST          NUMBER OF LINES OF NOTES TO BE PRINTED
   MTEST          NUMBER OF HEADER LINES TO BE PRINTED


SUBROUTINE PSTATS CALCULATES AND PRINTS THE GENERATED PERCENTAGES OF LOGIN
                  ATTRIBUTES
                  (CALLED FROM DVPRO)
                  (CALLS HEADER)

   IA             NUMBER OF INBOUND APPROACH BEING PROCESSED [1-12]
   IAN            INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING
                  PROCESSED [1-6]
   ID             GENERATED DRIVER CLASS NUMBER
   ILN            INDEX NUMBER FOR NLANES ARRAY OF /APPRO/ OF INBOUND LANE
                  BEING PROCESSED [1-6]
   IV             VEHICLE CLASS NUMBER FOR GENERATED VEHICLES
   JAN            INDEX NUMBER FOR LOBA ARRAY OF /INTER/ OF APPROACH BEING
                  PROCESSED [1-6]
   NLANES         NLANES(IAN)
   NGWRIT         NUMBER OF GENERATED VEHICLES WRITTEN BY APPROACH
   NUMV           TOTAL NUMBER OF VEHICLES WRITTEN
   SUM            SUMMATION OF VARIOUS GENERATED STATISTICS TO FIGURE
                  PERCENTAGES


SUBROUTINE PSUMDV PRINTS THE SUMMARY STATISTICS
                  (CALLED FROM DVPRO)
                  (CALLS HEADER ABORTR)

   IAN            INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING
                  PROCESSED [1-6]
   IDENSE         JAM DENSITY OF TRAFFIC GENERATED
   IVOLT          TOTAL INTERSECTION VOLUME
   LVTOT          TOTAL LENGTH OF VEHICLES PLUS 4 FEET FOR EACH VEHICLE
   NGTOT          TOTAL NUMBER OF GENERATED VEHICLES WRITTEN
   NGVOL          VOLUME OF VEHICLES GENERATED
   NGWRIT(6)      NUMBER OF GENERATED VEHICLES WRITTEN FOR EACH APPROACH
   NSELIM         NUMBER OF SPECIAL VEHICLES ELIMINATED
   NSREAD         NUMBER OF SPECIAL VEHICLES READ
   NSTOT          TOTAL NUMBER OF SPECIAL VEHICLES WRITTEN
   NSVOL          VOLUME OF SPECIAL VEHICLES WRITTEN
   NSWRIT(6)      NUMBER OF SPECIAL VEHICLES WRITTEN FOR EACH APPROACH
   NTOTAL         TOTAL NUMBER OF SPECIAL AND GENERATED VEHICLES WRITTEN FOR AN
                  APPROACH
   NTVOL          TOTAL VOLUME OF SPECIAL AND GENERATED VEHICLES WRITTEN FOR AN
                  APPROACH


FUNCTION RANF     GENERATES RANDOM NUMBERS (IBM ONLY)
                  (CALLED FROM DISCPT ERLANG GAMMA  GENHED LGNRML
                               NEGEXP NORMAL SNEGEX UNIFRM)

   A              FUNCTION PARAMETER WHICH CONTROLS OPERATION OF RANF
                        <0 = RETURN RANDOM NUMBER SEED
                        =0 = GENERATE A NEW RANDOM NUMBER
                        >0 = SET RANDOM NUMBER SEED USING A
   ISEED          RANDOM NUMBER SEED
   I1             1
   I16P3          2**16+3
   I31            2**31
   TM31           2**=31


SUBROUTINE READAP READS THE APPROACH INFORMATION
                  (CALLED FROM READIN)
                  (CALLS HEADER)

   IA             NUMBER OF INBOUND APPROACH BEING PROCESSED [1-12]
   IAN            INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING
                  PROCESSED [1-6]
   IUSED(12)      STATUS OF WHETHER DATA HAS BEEN ENTERED FOR ENTRY
                        0 = NOT ENTERED
                        1 = ENTERED
   IYES           YES/NO FOR USER=SUPPLIED PERCENT OF EACH VEHICLE CLASS
                  MAKING UP THE TRAFFIC STREAM
   JAAZIM         AZIMUTH FOR APPROACH
   JDIST          FIRST 4 CHARACTERS OF HEADWAY DISTRIBUTION NAME FOR APPROACH
   JVOL           VOLUME OF TRAFFIC TO BE GENERATED FOR APPROACH
   KDIST          SECOND 4 CHARACTERS OF HEADWAY DISTRIBUTION NAME FOR APPROACH
   LTEST          LINE COUNT TEST VARIABLE FOR HEADER
   MDEGST         NUMBER OF DEGREES LEFT OR RIGHT OF EXACTLY STRAIGHT
                  CONSIDERED STRAIGHT THROUGH MOVEMENT FOR APPROACH
   NLANES         NUMBER OF LANES FOR APPROACH
   N              NUMBER OF CARDS OF LANE DATA FOR OUTBOUND APPROACH
   PDIST          PARAMETER FOR HEADWAY DISTRIBUTION FOR APPROACH
```

SUM	SUM OF PERCENTAGES
XMEANS	MEAN SPEED FOR APPROACH
XPERL(6,6)	PERCENT OF TRAFFIC VOLUME FOR INBOUND APPROACH ENTERING BY A LANE (INBOUND LANE, INBOUND APPROACH)
XPERT(6,6)	PERCENT OF TRAFFIC VOLUME GOING FROM AN INBOUND TO AN OUTBOUND APPROACH (OUTBOUND APPROACH, INBOUND APPROACH)
X85PER	85 PERCENTILE SPEED FOR APPROACH

SUBROUTINE READGP READS THE GEOMETRY PROCESSOR DATA
(CALLED FROM READIN)

NARCS	NUMBER OF ARCS
NLINES	NUMBER OF LINES
NSDRC	NUMBER OF SIGHT DISTANCE RESTRICTION COORDINATES

SUBROUTINE READIN READS INPUT DATA
(CALLED FROM DVPRO)
(CALLS READGP READYO HEADER READIO READAP READOP)

SUBROUTINE READIO READS THE NUMBER AND LIST OF INBOUND AND OUTBOUND APPROACHES
(CALLED FROM READIN)
(CALLS HEADER)

IAN	INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1-6]
IANP1	IAN + 1
JAN	INDEX NUMBER FOR LOBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1-6]

SUBROUTINE READOP READS THE NUMBER OF APPROACHES AND THE DRIVER-VEHICLE PROCESSOR OPTIONS
(CALLED FROM READIN)
(CALLS HEADER)

ITSIM	NUMBER OF MINUTES FOR GENERATING TRAFFIC
NTEST	TEST FOR NIBA + NOBA = NAP

SUBROUTINE READYO READS THE YES OPTIONS
(CALLED FROM READIN)
(CALLS HEADER)

SUM	SUM OF PERCENTAGES

SUBROUTINE SNEGEX GENERATES SHIFTED NEGATIVE EXPONENTIAL HEADWAYS
(CALLED FROM GENHED)
(CALLS RANF)

QTIMS(1)	ARRAY FOR QUEUE-IN TIME
TAU	TAU PARAMETER FOR SHIFTED NEGATIVE EXPONENTIAL DISTRIBUTION (MINIMUM VALUE OF SHIFT)
TBAR	MEAN OF SHIFTED NEGATIVE EXPONENTIAL DISTRIBUTION
THEAD	RANDOM SHIFTED NEGATIVE EXPONENTIAL HEADWAY

SUBROUTINE UNIFRM GENERATES UNIFORM HEADWAYS
(CALLED FROM GENHED)
(CALLS RANF)

A	MINIMUM VALUE FOR UNIFORM DISTRIBUTION
B	MAXIMUM VALUE FOR UNIFORM DISTRIBUTION
BMA	B - A
QTIMS(1)	ARRAY FOR QUEUE-IN TIME
THEAD	RANDOM UNIFORM HEADWAY

SUBROUTINE WRITDV CALCULATES MINIMUM AND MAXIMUM SPEEDS
(CALLED FROM DVPRO)

APIJR	AVERAGE PIJR VALUE WEIGHTED BY VOLUME, PERCENT OF DRIVER CLASSES, AND PERCENT OF VEHICLE CLASSES
DVCHAR	DRIVER-VEHICLE OPERATIONAL FACTOR (IDCHAR*IVCHAR)
IAN	INDEX NUMBER FOR LIBA ARRAY OF /INTER/ OF APPROACH BEING PROCESSED [1-6]
PERV	PERCENT OF VEHICLES MAKING UP THE TRAFFIC STREAM
SUMP	SUM OF WEIGHTED PIJR TIME
TTV	TOTAL VOLUME OF TRAFFIC GENERATED
VCHAR	VEHICLE CHARACTERISTICS
VMMS	MEAN SPEED MINUS ONE STANDARD DEVIATION
VMPS	MEAN SPEED PLUS ONE STANDARD DEVIATION
VOLIAN	IVOL(IAN)
VSIG	STANDARD DEVIATION OF SPEED

6. ALPHABETICAL LISTING OF ALL THE ROUTINES AND THE ROUTINES WHICH CAN CALL THEM

```
BIASLT = DVPRO
CONST  = GENHED
DISCRT = GENDV
ERLANG = GENHED
GAMMA  = GENHED
GENDV  = DVPRO
GENDVH = GENDV
GENHED = DVPRO
HEADER = READIN READIO READOP READAP READYO GENHED GENDVH PNOTES PSUMDV
         PSTATS
LGNRML = GENHED
NEGEXP = GENHED
NORMAL = GENDV
PNOTES = DVPRO
PSTATS = DVPRO
PSUMDV = DVPRO
RANF   = ERLANG GAMMA  LGNRML NEGEXP SNEGEX UNIFRM DISCRT NORMAL GENHED
READAP = READIN
READGP = READIN
READIN = DVPRO
READIO = READIN
READOP = READIN
READYO = READIN
SNEGEX = GENHED
UNIFRM = GENHED
WRITDV = DVPRO
```

7. GENERALIZED CALLING SEQUENCE DIAGRAM

```
                                         DVPRO
                                           ↓
                                           ↓
  -------------------------------------------------------------------------------
  ↓          ↓        ↓        ↓          ↓          ↓          ↓         ↓
  ↓          ↓        ↓        ↓          ↓          ↓          ↓         ↓
READIN    WRITDV   BIASLT   GENHED     GENDV      PNOTES     PSUMDV    PSTATS
#HEADER                     #CONST     #GENDVH    #HEADER    #HEADER   #HEADER
#READIO                     #ERLANG     >HEADER
 >HEADER                     >RANF     #DISCRT
#READOP                     #GAMMA      >RANF
 >HEADER                     >RANF     #NORMAL
#READAP                     #LGNRML     >RANF
 >HEADER                     >RANF
#READGP                     #NEGEXP
#READYO                      >RANF
 >HEADER                    #SNEGEX
                             >RANF
                            #UNIFRM
                             >RANF
                            #HEADER
```

8. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED

```
A        LOCAL    GAMMA  UNIFRM
ALPHA    LOCAL    ERLANG GAMMA
APIJP    LOCAL    WRITDV
B        LOCAL    UNIFRM
BMA      LOCAL    UNIFRM
DVCHAR   LOCAL    WRITDV
EX       LOCAL    LGNRML
EXP      LOCAL    LGNRML
EY       LOCAL    LGNRML
FPER     LOCAL    BIASLT
FPERL   /DVDATA/  BIASLT READOP
FPERR   /DVDATA/  BIASLT READOP
FPSMPH  /LITCON/  READAP READIN
HEAD     LOCAL    GENDV
HMIN    /DVDATA/  GENDV  PNOTES READOP
I        LOCAL    ABORTR CONST  DISCRT ERLANG GAMMA  GENDV  GENDVH GENHED
                  LGNRML NEGEXP NORMAL PNOTES READAP READGP READYO SNEGEX
                  UNIFRM
IA       LOCAL    BIASLT GENDV  READAP
IAAZIM  /APPRO /  BIASLT READAP
IAMAX   /CLASS /  BLKDAT READYO WRITDV
IAN      LOCAL    BIASLT GENDV  GENHED PSTATS PSUMDV READAP READIO WRITDV
IANGLE   LOCAL    BIASLT
IANP1    LOCAL    READIO
IAP      LOCAL    GENDV
IAZIM    LOCAL    BIASLT
ID       LOCAL    GENDV  PSTATS WRITDV
IDC      LOCAL    GENDV
IDCHAR  /CLASS /  BLKDAT READYO WRITDV
IDENSE   LOCAL    PSUMDV
IDIST   /APPRO /  GENHED READAP
IDISTN  /LITCON/  BLKDAT GENHED READAP
IDMAX   /CLASS /  BLKDAT READYO WRITDV
IDNUM    LOCAL    GENHED
IDV      LOCAL    GENDV
IELIM    LOCAL    PSUMDV
IEOF    /DVDATA/  BLKDAT GENDV  READYO
IFORM    LOCAL    BLKDAT GENHED PSUMDV READAP
ITTURN  /APPRO /  BIASLT GENDV  READAP
IL       LOCAL    GENDV
ILN      LOCAL    BIASLT GENDV  PSTATS
INEXTV   LOCAL    GENDV
IPLOGO   LOCAL    GENDV
IPRTLO   LOCAL    GENDV
IREAD    LOCAL    PSUMDV
IRMIN   /CLASS /  BLKDAT READYO WRITDV
ISPLHD   LOCAL    GENDV
ISUMIV   LOCAL    GENHED
ISUMNG   LOCAL    GENHED
ISUMVG   LOCAL    GENHED
ITABL    LOCAL    GENDV  GENDVH
ITEMPD   LOCAL    ABORTR
ITITLE  /TITLE /  HEADER READIN WRITDV
ITSIM    LOCAL    READOP
ITURN    LOCAL    BIASLT GENDV
IUSED    LOCAL    READAP
IV       LOCAL    GENDV  PSTATS WRITDV
IVC      LOCAL    GENDV
IVCHAR  /CLASS /  BLKDAT READYO WRITDV
IVEL     LOCAL    GENDV
IVMAX   /CLASS /  BLKDAT READYO WRITDV
IVOL    /APPRO /  GENHED PSUMDV READAP WRITDV
IVOLGN   LOCAL    GENHED
IVOLIA   LOCAL    GENHED
IVOLT    LOCAL    PSUMDV
IYES     LOCAL    READAP
```

```
IYESD   /CLASS / BLKDAT READYO
IYESDL  /CLASS / BLKDAT GENDV  READYO
IYESP   /CLASS / BLKDAT READYO
IYESV   /CLASS / BLKDAT READYO
IYESVL  /CLASS / BLKDAT GENDV  READYO
J        LOCAL   ERLANG GAMMA  GENDV  LGNRML READAP READYO
JA       LOCAL   BIASLT GENDV
JAAZIM   LOCAL   READAP
JAN      LOCAL   BIASLT GENDV  PSTATS READAP READIO
JAP      LOCAL   GENDV
JAZIM    LOCAL   BIASLT
JDIST    LOCAL   READAP
JLN      LOCAL   BIASLT
JVOL     LOCAL   READAP
K        LOCAL   ERLANG GAMMA
KAN      LOCAL   GENDV
KDIST    LOCAL   READAP
KGEOM    LOCAL   READAP
K1       LOCAL   GAMMA
K2       LOCAL   GAMMA
LENV    /CLASS / BLKDAT GENDV  READYO WRITDV
LGEOM1   LOCAL   READAP
LGEOM2   LOCAL   READAP
LIBA    /INTER / BIASLT GENDV  GENHED PSTATS PSUMDV READAP READIO
LINES   /OUTPUT/ BLKDAT GENDVH GENHED PNOTES PSTATS PSUMDV READAP READIO
                 READOP READYO
LOBA    /INTER / BIASLT GENDV  PSTATS READAP READIO
LTEST    LOCAL   PNOTES PSUMDV READAP
LVTOT    LOCAL   GENDV  PSUMDV
MAXV    /CLASS / BLKDAT GENDV
MAYENT  /DVDATA/ GENDV  READAP
MDEGST   LOCAL   BIASLT READAP
MLANES   LOCAL   PSTATS READAP
MODELT  /OUTPUT/ BLKDAT GENDV  WRITDV
MSG      LOCAL   ABORTR
MSG901   LOCAL   GENHED
MSG902   LOCAL   PSUMDV
MTEST    LOCAL   PNOTES
N        LOCAL   READAP
NAP     /INTER / READAP READOP
NARCS    LOCAL   READGP
NBLANK  /LITCON/ BLKDAT READAP READYO
NCHS     LOCAL   ABORTR
NDEGST  /APPRO / BIASLT READAP
NDRICL  /CLASS / GENDV  PSTATS READOP READYO WRITDV
NGTOT    LOCAL   PSUMDV
NGVOL    LOCAL   PSUMDV
NGWRIT   LOCAL   GENDV  PSTATS PSUMDV
NIBA    /INTER / BIASLT GENDV  GENHED PSTATS PSUMDV READAP READIO READOP
                 WRITDV
NL       LOCAL   BIASLT
NLANES  /APPRO / BIASLT GENDV  PSTATS READAP
NLINE   /OUTPUT/ BLKDAT GENDV  GENDVH GENHED HEADER PNOTES PSTATS PSUMDV
                 READAP READIO READOP READYO
NLINES   LOCAL   READGP
NNO     /LITCON/ BLKDAT READAP READYO
NOBA    /INTER / BIASLT GENDV  PSTATS READAP READIO READOP
NOTE    /OUTPUT/ BLKDAT GENDV  PNOTES
NPAGE   /OUTPUT/ BLKDAT HEADER
NSDRC    LOCAL   READGP
NSELIM   LOCAL   PSUMDV
NSREAD   LOCAL   GENDV  PSUMDV
NSTOT    LOCAL   PSUMDV
NSVOL    LOCAL   PSUMDV
NSWRIT   LOCAL   GENDV  PSUMDV
NTABL   /OUTPUT/ BLKDAT GENDVH GENHED PSTATS PSUMDV READAP READIO READOP
                 READYO
NTEST    LOCAL   READOP
NTOTAL   LOCAL   PSUMDV
NTVOL    LOCAL   PSUMDV
NUM      LOCAL   DISCRT
NUMV     LOCAL   PSTATS
NVA     /APPRO / BLKDAT GENDV  GENHED
NVATAN  /ZTEMPD/ CONST  ERLANG GAMMA  GENHED LGNRML NEGEXP SNEGEX UNIFRM
NVEHCL  /CLASS / GENDV  PSTATS READAP READOP READYO WRITDV
NWDS     LOCAL   ABORTR
NYES    /LITCON/ BLKDAT GENDV  READAP READYO
NZTEMP   LOCAL   ABORTR
PARAM   /APPRO / GENHED READAP
PARIAN  /ZTEMPD/ ERLANG GAMMA  GENHED LGNRML SNEGEX UNIFRM
PDIST    LOCAL   READAP
PERDIF   LOCAL   GENHED
PERV     LOCAL   WRITDV
PIJR    /CLASS / BLKDAT READYO WRITDV
Q        LOCAL   GAMMA
QBIG     LOCAL   GENDV
QMIN     LOCAL   GENDV
QTIM     LOCAL   GENDV
QTIME   /DVDATA/ GENDV  GENHED
QTIMS    LOCAL   CONST  ERLANG GAMMA  LGNRML NEGEXP SNEGEX UNIFRM
QTLAST  /DVDATA/ BLKDAT GENDV
RANNUM   LOCAL   DISCRT
SIMTIM  /DVDATA/ CONST  ERLANG GAMMA  GENDV  GENHED LGNRML NEGEXP PSUMDV
                 READOP SNEGEX UNIFRM
SPERD   /STATS / BLKDAT GENDV  PSTATS
SPERL   /STATS / BLKDAT GENDV  PSTATS
SPERT   /STATS / BLKDAT GENDV  PSTATS
SPERV   /STATS / BLKDAT GENDV  PSTATS
SQRT3   /LITCON/ READIN UNIFRM
STDX     LOCAL   LGNRML
STDY     LOCAL   LGNRML
SUM      LOCAL   BIASLT DISCRT LGNRML NORMAL PSTATS READAP READYO
SUMP     LOCAL   WRITDV
TAU      LOCAL   SNEGEX
TBAR     LOCAL   SNEGEX
THEAD    LOCAL   ERLANG GAMMA  LGNRML NEGEXP SNEGEX UNIFRM
THEAN   /ZTEMPD/ CONST  ERLANG GAMMA  GENHED LGNRML NEGEXP READAP SNEGEX
                 UNIFRM
TR       LOCAL   ERLANG GAMMA
TTV      LOCAL   WRITDV
VARY     LOCAL   LGNRML
VCHAR    LOCAL   WRITDV
VEL      LOCAL   GENDV  NORMAL
VMAX    /CLASS / GENDV  READAP WRITDV
VMEAN   /APPRO / GENDV  NORMAL READAP WRITDV
VMIN    /CLASS / GENDV  READAP WRITDV
VMMS     LOCAL   WRITDV
VMPS     LOCAL   WRITDV
VOLIAN   LOCAL   WRITDV
VSIG     LOCAL   WRITDV
VSIGMA  /APPRO / GENDV  NORMAL READAP WRITDV
XMEANS   LOCAL   READAP
XPER     LOCAL   DISCRT
XPERU   /CLASS / BLKDAT GENDV  READYO WRITDV
XPERL   /ZTEMPD/ BIASLT READAP
XPERLO  /APPRO / BIASLT GENDV
XPERT   /DVDATA/ BIASLT GENDV  READAP
XPERTS   LOCAL   BIASLT
XPERV   /DVDATA/ BLKDAT GENDV  READAP WRITDV
X85PER   LOCAL   READAP
YPERT    LOCAL   READAP
ZERO    /DVDATA/ BLKDAT READAP READYO
ZTEMPD  /ZTEMPD/ ABORTR
```

APPENDIX D

ADDITIONAL INFORMATION FOR THE TRAFFIC SIMULATION PROCESSOR

This page intentionally left blank to facilitate printing on 2 sides.

```
C IDENTIFY,SIMPRO,60,3,SIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PAC
C    FILES,TAPE5=513,TAPE7=65,TAPE8=513,TAPE9=513,OUTPUT=513
C ENTITY
C    NAME,APPRO,12,***** ENTITY FOR APPROACHES *****
C    ORDINARY,NLANES,6,LLANES(6),50,NVIL(6),63,ISLIM,118,IALEFT,12
C    ORDINARY,NSDR,5,ISDRN(5),30,ISDRA(5),12
C    NAME,CONFLT,1000,***** ENTITY FOR INTERSECTION CONFLICTS *****
C    ORDINARY,ICONP(2),125,ICONA(2),12,ICOND(2),250,ICONAN,360
C    ORDINARY,ICONI(2),60,ICONV(2),200,IDUMCO,0
C    NAME,LANE,50,***** ENTITY FOR APPROACH LANES *****
C    ORDINARY,LWID,15,NLL,50,NLR,50,ISNA,12,NPINT,7,LINTP(7),125
C    ORDINARY,IFVL,200,ILVL,200,LCONTR,7,LTURN,15,LGEOM(4),1000
C    ORDINARY,NLDL,5,LLDL(5),20,IBLN,25,IDUMLA,0
C    NAME,PATH,125,***** ENTITY FOR INTERSECTION PATHS *****
C    ORDINARY,LENP,250,IOPT,1,LIBL,50,LOBL,50,IFVP,200,ILVP,200,LIMP,118
C    ORDINARY,IPT,8,NGEOCP,60,NCPSET,60,ICPSET(60),1,LOBAP,12,ILCH,1
C    ORDINARY,IGEOCP(60),1000
C    NAME,SDR,30,***** ENTITY FOR SIGHT DISTANCE RESTRICTION *****
C    ORDINARY,ICANSE(40),1000
C    NAME,VEHD,200,***** ENTITY FOR DYNAMIC VEHICLE ATTRIBUTES *****
C    ORDINARY,ISLP,8000,IACC,16000,IVEL,4034,IPOS,25000,ISET,6
C    ORDINARY,LCHGE,3,ISPDP,1,LEGAL,30,IPRTM,15,ITIMV,2000,IQDS,2000
C    ORDINARY,ISPDS,258134,ISDS,2000,IDVS,2000,ISTCON,61,IVMAXA,320
C    ORDINARY,IVMAXD,320,LATPOS,240,IDTS,56240,LALT,5,NORC,201,LOGFLG,15
C    LOGICI,MSTPF,MLAG,MTCARS,MFINL,MSFLG,MPOBS,MOASF,MSAOR,MPRO,MBLOCK
C    LOGICI,MININT
C    LOGICD,IFVA,IACDS,ICDFS,ISDEC,ISTMO,IACLDS,IRSTOP
C    FUNCTION,MSTPF    ,MPOBS    ,MLAG     ,MLAG     ,IFVA     ,MFINL =1
C    FUNCTION,MFINL =1,MTCARS   ,MOASF =1,MOASF =1,MSFLG    ,IFVA
C    FUNCTION,MTCARS   ,MSFLG    ,MBLOCK   ,MBLOCK   ,MSFLG    ,MPRO  =1
C    FUNCTION,MPRO  =1,IACDS    ,MSFLG    ,MSFLG    ,ICDFS    ,ISDEC
C    FUNCTION,MPOBS    ,ISTMO    ,MFINL =2,MFINL =2,MSAOR =1,MOASF =2
C    FUNCTION,MOASF =2,MSAOR =2,IACLDS   ,MSAOR =1,MPRO  =2,IACDS
C    FUNCTION,MSAOR =2,IRSTOP   ,IACDS    ,MPRO  =2,IACDS    ,IRSTOP
C    NAME,VEHF,200,***** ENTITY FOR FIXED VEHICLE ATTRIBUTES *****
C    ORDINARY,IDRICL,5,IVEHCL,15,ISPD,161,NOF,200,NOR,200,LNEXT,125
C    ORDINARY,LPRES,125,ITURN,3,IBAPS,6,IPRTLO,1,IEXTIM,25,NOBAPD,12
C    NAME,VEHIL,200,***** ENTITY FOR VEHICLE INTERSECTION LOGIC *****
C    LOGICI,MDEDIC,MINFLZ,MLUNC,MIUNC,MLYELD,MLSTOP,MATSTL,MSSRED,MLRTOR
C    LOGICI,MSSGRN,MCHKCF,MDUMIL
C    LOGICD,IDEDIC,INFLZ,ILUNC,ILYELD,ILSTOP,ICONTN,ICHKCF,IERROR
C    FUNCTION,MDEDIC   ,MINFLZ   ,IDEDIC   ,MINFLZ   ,MLUNC    ,INFLZ
C    FUNCTION,MLUNC    ,MIUNC    ,MLYELD   ,MLYELD   ,ILYELD   ,MLSTOP
C    FUNCTION,MLSTOP   ,MATSTL   ,MSSRED   ,MATSTL   ,ILSTOP   ,ICONTN
C    FUNCTION,MSSRED   ,MLRTOR   ,MSSGRN   ,MLRTOR   ,ICHKCF   ,ICONTN
C    FUNCTION,MSSGRN   ,MCHKCF   ,IERROR   ,MCHKCF   ,ICHKCF   ,ICONTN
C    FUNCTION,MIUNC    ,ILUNC    ,MCHKCF
C EXECUTIVE
C    ROUTINE,RGEOPD,APPRO,CONFLT,LANE        ,NOATTB,PATH,SDR
C    ROUTINE,RDVPRD                    ,LOGICV
C    ROUTINE,OBAP  ,APPRO        ,LANE,LOGICV,NOATTB            ,VEHD,VEHF
C    ROUTINE,SSOBAP                                             ,VEHD,VEHF
C    ROUTINE,LOGOUT,APPRO        ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,FLGNOR                    ,LOGICV                       ,VEHF
C    ROUTINE,INTERP       ,CONFLT      ,LOGICV,NOATTB,PATH      ,VEHD,VEHF
C    ROUTINE,LOKIOB                    ,LOGICV       ,PATH      ,VEHD,VEHF
C    ROUTINE,SSINTR                                  ,PATH      ,VEHD,VEHF
C    ROUTINE,CLRCON       ,CONFLT             ,NOATTB,PATH      ,VEHD,VEHF
C    ROUTINE,LOGIOB,APPRO        ,LANE,LOGICV       ,PATH      ,VEHD,VEHF
C    ROUTINE,IBAP  ,APPRO        ,LANE,LOGICV,NOATTB            ,VEHD,VEHF,VEHIL
C    ROUTINE,LOKIBI              ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,CHKDSP,APPRO                                       ,VEHD,VEHF
C    ROUTINE,CHKLDT              ,LANE                               ,VEHF
C    ROUTINE,SSIBAP,APPRO              ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,LOGIBI,APPRO        ,LANE,LOGICV       ,PATH      ,VEHD,VEHF
C    ROUTINE,PREST1              ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,PREST2                    ,LOGICV                  ,VEHD
C    ROUTINE,UNBIAS                                             ,VEHD,VEHF
C    ROUTINE,NEWVEL                    ,LOGICV                  ,VEHD
C    ROUTINE,LCHGEO                                             ,VEHD,VEHF
C    ROUTINE,ENDLCH                                             ,VEHD,VEHF
C    ROUTINE,LCHDES              ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,SVEHU               ,LANE                          ,VEHD
C    ROUTINE,DELAY               ,LANE                          ,VEHD,VEHF
C    ROUTINE,CKLALT              ,LANE                          ,VEHD,VEHF
C    ROUTINE,GAPACC                    ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,CHGMLN,APPRO        ,LANE,LOGICV                   ,VEHD,VEHF,VEHIL
C    ROUTINE,ACDCP               ,LANE,LOGICV                   ,VEHD,VEHF,VEHIL
C    ROUTINE,CARFOL                    ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,ACCEL                     ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,CRIDIS                    ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,ADLVAI,APPRO              ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,INTLOG                    ,LOGICV                  ,VEHD,VEHF,VEHIL
C    ROUTINE,SIGRES                    ,LOGICV                  ,VEHD,VEHF,VEHIL
C    ROUTINE,LSTOP                     ,LOGICV       ,PATH      ,VEHD,VEHF,VEHIL
C    ROUTINE,CHKSDR,APPRO,CONFLT,LANE,LOGICV       ,PATH      ,VEHD,VEHF,VEHIL
C    ROUTINE,CHKCON       ,CONFLT,LANE,LOGICV       ,PATH      ,VEHD,VEHF,VEHIL
C    ROUTINE,SETPTV,APPRO        ,LANE              ,PATH      ,VEHD,VEHF
C    ROUTINE,SETCON       ,CONFLT      ,LOGICV       ,PATH      ,VEHD,VEHF
C    ROUTINE,UNSETC       ,CONFLT                    ,PATH      ,VEHD,VEHF
C    ROUTINE,INFLZN              ,LANE,LOGICV                   ,VEHD,VEHF,VEHIL
C    ROUTINE,PATHF ,APPRO        ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,CHKMLN              ,LANE,LOGICV                   ,VEHD
C    ROUTINE,BANGS               ,LANE,LOGICV                   ,VEHD,VEHF
C    ROUTINE,BIAS                      ,LOGICV                  ,VEHD,VEHF
C    ROUTINE,LOGIN ,APPRO        ,LANE,LOGICV,NOATTB            ,VEHD,VEHF,VEHIL
C    ROUTINE,ACTSIG                    ,LOGICV
C    ROUTINE,ABORTR,APPRO,CONFLT,LANE        ,NOATTB,PATH,SDR,VEHD,VEHF,VEHIL
C    EXECUTE,EXEC
C TASKS
      PROGRAM    SIMPRO ( TAPE5=513,TAPE7=65,TAPE8=513,TAPE9=513,        COLEASE
     *                    OUTPUT=513 )                                  COLEASE
      COMMON / APPRO  / NLANES(   26)                                   COLEASE
      COMMON / CONFLT / ICONP (   12)                                   COLEASE
      COMMON / LANE   / LWID  (   28)                                   COLEASE
      COMMON / PATH   / LENP  (  132)                                   COLEASE
      COMMON / SDR    / ICANSE(   40)                                   COLEASE
      COMMON / VEHD   / ISLP  (   40)                                   COLEASE
      COMMON / VEHF   / IDRICL(   12)                                   COLEASE
      COMMON / VEHIL  / MDEDIC(   20)                                   COLEASE
      COMMON / ATTB   / IAT (3,  310)                                   COLEASE
      COMMON / ENTITY / IEN (9,    8)                                   COLEASE
      COMMON / FUN    / IFU (2,   31)                                   COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / NOATTB / NOATTB(8)                                       COLEASE
      COMMON / STACK  / IS     ( 5821)                                  COLEASE
      DO 1010  I = 1 ,  310                                             COLEASE
      NLANES(I) = 0                                                     COLEASE
      IAT(3,I) = LSHIFT(1,IAT(3,I)) - 1                                 COLEASE
      IAT(3,I) = LSHIFT(IAT(3,I),IAT(2,I))                              COLEASE
 1010 CONTINUE                                                          COLEASE
      DO 1020  I = 1 , 31                                               COLEASE
      J = IFU(2,I)                                                      COLEASE
      IFU(2,I) = IAT(2,J)                                               COLEASE
 1020 CONTINUE                                                          COLEASE
      DO 1030  I = 1 , 5821                                             COLEASE
      IS(I) = 0                                                         COLEASE
 1030 CONTINUE                                                          COLEASE
      CALL  EXEC                                                        COLEASE
      CALL  EXIT                                                        COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      BLOCK DATA                                                         COLEASE
      COMMON / ATTB   / IAT1(300),IAT2(300),IAT3(300),IAT4( 30)          COLEASE
      COMMON / ENTITY / IEN (9,     8)                                   COLEASE
      COMMON / FUN    / IFU (2,    31)                                   COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / NOATTB / NOATTB( 8)                                       COLEASE
      DATA      IAT1   /  0, 0, 3, 0, 3, 6, 0, 9, 6, 0,15, 6, 0,21, 6,   COLEASE
     *                    0,27, 6, 0,33, 6, 0,39, 6, 0,45, 6, 0,51, 6,   COLEASE
     *                    1, 0, 6, 1, 6, 6, 1,12, 6, 1,18, 7, 1,25, 4,   COLEASE
     *                    1,29, 3, 1,32, 5, 1,37, 5, 1,42, 5, 1,47, 5,   COLEASE
     *                    1,52, 5, 2, 0, 4, 2, 4, 4, 2, 8, 4, 2,12, 4,   COLEASE
     *                    2,16, 4, 0, 0, 7, 0, 7, 7, 0,14, 4, 0,18, 4,   COLEASE
     *                    0,22, 8, 0,30, 8, 0,38, 9, 0,47, 6, 0,53, 6,   COLEASE
     *                    1, 0, 8, 1, 8, 8, 1,16, 0, 0, 0, 4, 0, 4, 6,   COLEASE
     *                    0,10, 6, 0,16, 4, 0,20, 3, 0,23, 7, 0,30, 7,   COLEASE
     *                    0,37, 7, 0,44, 7, 0,51, 7, 1, 0, 7, 1, 7, 7,   COLEASE
     *                    1,14, 8, 1,22, 8, 1,30, 3, 1,33, 4, 1,37,10,   COLEASE
     *                    1,47,10, 2, 0,10, 2,10,10, 2,20, 3, 2,23, 5,   COLEASE
     *                    2,28, 5, 2,33, 5, 2,38, 5, 2,43, 5, 2,48, 5,   COLEASE
     *                    2,53, 0, 0, 0, 8, 0, 8, 1, 0, 9, 6, 0,15, 6,   COLEASE
     *                    0,21, 8, 0,29, 8, 0,37, 7, 0,44, 4, 0,48, 6,   COLEASE
     *                    0,54, 6, 1, 0, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1,   COLEASE
     *                    1, 4, 1, 1, 5, 1, 1, 6, 1, 1, 7, 1, 1, 8, 1,   COLEASE
     *                    1, 9, 1, 1,10, 1, 1,11, 1, 1,12, 1, 1,13, 1,   COLEASE
     *                    1,14, 1, 1,15, 1, 1,16, 1, 1,17, 1, 1,18, 1,   COLEASE
     *                    1,19, 1, 1,20, 1, 1,21, 1, 1,22, 1. 1,23, 1/   COLEASE
      DATA      IAT2   /  1,24, 1, 1,25, 1, 1,26; 1, 1,27, 1, 1,28, 1,   COLEASE
     *                    1,29, 1, 1,30, 1, 1,31, 1, 1,32, 1, 1,33, 1,   COLEASE
     *                    1,34, 1, 1,35, 1, 1,36, 1, 1,37, 1, 1,38, 1,   COLEASE
     *                    1,39, 1, 1,40, 1, 1,41, 1, 1,42, 1, 1,43, 1,   COLEASE
     *                    1,44, 1, 1,45, 1, 1,46, 1, 1,47, 1, 1,48, 1,   COLEASE
     *                    1,49, 1, 1,50, 1, 1,51, 1, 1,52, 1, 1,53, 1,   COLEASE
     *                    1,54, 1, 1,55, 1, 1,56, 1, 1,57, 1, 1,58, 1,   COLEASE
     *                    1,59, 1, 2, 0, 4, 2, 4, 1, 2, 5,10, 2,15,10,   COLEASE
     *                    2,25,10, 2,35,10, 2,45,10, 3, 0,10, 3,10,10,   COLEASE
     *                    3,20,10, 3,30,10, 3,40,10, 3,50,10, 4, 0,10,   COLEASE
     *                    4,10,10, 4,20,10, 4,30,10, 4,40,10, 4,50,10,   COLEASE
     *                    5, 0,10, 5,10,10, 5,20,10, 5,30,10, 5,40,10,   COLEASE
     *                    5,50,10, 6, 0,10, 6,10,10, 6,20,10, 6,30,10,   COLEASE
     *                    6,40,10, 6,50,10, 7, 0,10, 7,10,10, 7,20,10,   COLEASE
     *                    7,30,10, 7,40,10, 7,50,10, 8, 0,10, 8,10,10,   COLEASE
     *                    8,20,10, 8,30,10, 8,40,10, 8,50,10, 9, 0,10,   COLEASE
     *                    9,10,10, 9,20,10, 9,30,10, 9,40,10, 9,50,10,   COLEASE
     *                   10, 0,10,10,10,10,10,20,10,10,30,10,10,40,10,   COLEASE
     *                   10,50,10,11, 0,10,11,10,10,11,20,10,11,30,10,   COLEASE
     *                   11,40,10,11,50,10,12, 0,10, 0, 0,10, 0,10,10/   COLEASE
      DATA      IAT3   /  0,20,10, 0,30,10, 0,40,10, 0,50,10, 1, 0,10,   COLEASE
     *                    1,10,10, 1,20,10, 1,30,10, 1,40,10, 1,50,10,   COLEASE
     *                    2, 0,10, 2,10,10, 2,20,10, 2,30,10, 2,40,10,   COLEASE
     *                    2,50,10, 3, 0,10, 3,10,10, 3,20,10, 3,30,10,   COLEASE
     *                    3,40,10, 3,50;10, 4, 0,10, 4,10.10, 4,20,10,   COLEASE
     *                    4,30,10, 4,40,10, 4,50,10, 5, 0,10, 5,10,10,   COLEASE
     *                    5,20,10, 5,30,10, 5,40,10, 5,50,10, 6, 0,10,   COLEASE
     *                    6,10,10, 6,20,10, 6,30,10, 0, 0,13, 0,13,14,   COLEASE
     *                    0,27,12, 0,39,15, 0,54, 3, 0,57, 2, 0,59, 1,   COLEASE
     *                    1, 0, 5, 1, 5, 4, 1, 9,11, 1,20,11, 1,31,18,   COLEASE
     *                    1,49,11, 2, 0,11, 2,11, 6, 2,17, 9, 2,26, 9,   COLEASE
     *                    2,35, 8, 2,43,16, 3, 0, 3, 3, 3, 8, 3,11, 4,   COLEASE
     *                    4, 0, 2, 4, 2, 2, 4, 4, 2, 4, 6, 2, 4, 8, 2,   COLEASE
     *                    4,10, 2, 4,12, 2, 4,14, 2, 4,16, 2, 4,18, 2,   COLEASE
     *                    4,20, 2, 5, 0, 2, 5, 2, 2, 5, 4, 2, 5, 6, 2,   COLEASE
     *                    5, 8, 2, 5,10, 2, 5,12, 2, 0, 0, 3, 0, 3, 4,   COLEASE
     *                    0, 7, 8, 0,15, 8, 0,23, 8, 0,31, 7, 0,38, 7,   COLEASE
     *                    0,45, 2, 0,47, 3, 0,50, 1, 0,51, 5, 0,56, 4,   COLEASE
     *                    0, 0, 2, 0, 2, 2, 0, 4, 2, 0, 6, 2, 0, 8, 2,   COLEASE
     *                    0,10, 2, 0,12, 2, 0,14, 2, 0,16, 2, 0,18, 2/   COLEASE
      DATA      IAT4   /  0,20, 2, 0,22, 2, 1, 0, 2, 1, 2, 2, 1, 4, 2,   COLEASE
     *                    1, 6, 2, 1, 8, 2, 1,10, 2, 1,12, 2, 1,14, 2/   COLEASE
      DATA      IEN    /   12,  26,   3,    1,   0,   0,   0,   0,   1,  COLEASE
     *                   1000,  12,   2,   37,   0,   0,   0,   0,  27,  COLEASE
     *                     50,  28,   3, 2037,   0,   0,   0,   0,  39,  COLEASE
     *                    125, 132,  13, 2187,   0,   0,   0,   0,  67,  COLEASE
     *                     30,  40,   7, 3812,   0,   0,   0,   0, 199,  COLEASE
     *                    200,  40,   6, 4022,   1,   5,  18,   1, 239,  COLEASE
     *                    200,  12,   1, 5222,   0,   0,   0,   0, 279,  COLEASE
     *                    200,  20,   2, 5422,   1,   1,  13,  19, 291/  COLEASE
      DATA      IFU    /              6, 272,      8330, 272,    589930, 273, COLEASE
     *                            34881, 273,     39041, 273,     84033, 273, COLEASE
     *                             4490, 274,       346, 274,    262506, 274, COLEASE
     *                           655722, 274,      4746, 275,       602, 275, COLEASE
     *                           262762, 275,    655978, 275,      1025, 276, COLEASE
     *                            10369, 277,     22657, 278,    149569, 278, COLEASE
     *                                2, 303,         9, 304,        85, 305, COLEASE
     *                              293, 306,      5669, 307.      9765, 308, COLEASE
     *                           150053, 308,   2394661, 308,   2097301, 308, COLEASE
     *                            84517, 309,   1346085, 309,   1048725, 309, COLEASE
     *                           559653, 310/                                 COLEASE
      DATA      LTRUE  / 1 /                                                  COLEASE
      DATA      LFALSE / 2 /                                                  COLEASE
      DATA      NOATTB /  26, 12, 28,132, 40, 40, 12, 20 /                    COLEASE
C
C=====USER DEFINED BLOCK DATA
C
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),THAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
C6    COMMON / PRTPVA / DISTAD(200)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / ZTEMPD(110)
      DATA      AUTOL  / 1.7 /
      DATA      DUTOL  / 2.66666666666667 /
      DATA      FACTOR / 2.0 /
      DATA      IARRPH / 0 /
      DATA      ICONUP / 0 /
      DATA      IGEOP  / 8 /
      DATA      INPUT  / 5 /
      DATA      IPFLAG / 4H     /
      DATA      IPTHUP / 0 /
      DATA      IQ     / 200*0 /
      DATA      IQF    / 0 /
      DATA      IVEHP  / 9 /
```

```
      DATA    JPFLAG / 4H      /
      DATA    KPFLAG / 4H      /
      DATA    LIBAR  / 12*-100000000 /
      DATA    LOBAR  / 12*-100000000 /
      DATA    LQ     / 36*0 /
      DATA    MNVSY  / 0 /
      DATA    MSGR   / 4H NRN,4HAME ,4HGT 3,4H5     /
      DATA    NR     / 13 /
      DATA    NRNAMM / 35 /
      DATA    NUMV   / 1 /
      DATA    QTIME  / 25*-1.0 /
      DATA    TMTIME / 5*0.0 /
      END
```

BLKDATA

```
      SUBROUTINE EXEC
      COMMON / INTER   / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                   LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                   NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                   LIBAR(12),LOBAR(12)
      COMMON / QUE     / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                   NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                   NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                   STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                   VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                   LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                   PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE   / ITITLE(20)
      COMMON / USER    / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                   CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                   APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
C¬    DIMENSION          MSG(6)
C6    DIMENSION          IFET1(8),IBUF1(513),MSG1(3),
C6   *                   IFET2(8),IBUF2(513),MSG2(3),
C6   *                   IFET3(8),IBUF3(513),MSG3(3)
C7    DIMENSION          IFET4(8),IBUF4(513),MSG4(3)
C¬    DATA     MSG     / 4H FAT,4HAL E,4HXECU,4HTION,4H ERR,4HOR    /
C6    DATA     MSG1    / 21L TAPE1 ISLCPF ERROR  /
C6    DATA     MSG2    / 21L TAPE2 ISLCPF ERROR  /
C6    DATA     MSG3    / 21L TAPE3 ISLCPF ERROR  /
C7    DATA     MSG4    / 21L TAPE4 ISLCPF ERROR  /
      DATA     N1,N2   / 4HEXEC,2H    /
C¬601 FORMAT(1H2)
C;601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
C;   *         14HLATION PACKAGE,//,1X,20A4,//)
C6701 FORMAT(*0*F7.2* (*)
C
C-----SUBROUTINE EXEC IS THE MAIN DRIVER FOR SIMPRO AND CONTROLS THE
C-----CALLING OF THE VARIOUS OTHER ROUTINES
C
C-----CA = OUTBOUND APPROACH VEHICLE PRINT FOR CDC
C-----CB = OUTBOUND APPROACH VEHICLE PRINT FOR IBM
C-----CC = OUTBOUND APPROACH VEHICLE PRINT TIME CHECK
C-----CD = OUTBOUND APPROACH VEHICLE PRINT IPRTLO CHECK
C-----CE = OUTBOUND APPROACH ENTITY PRINT
C-----CF = OUTBOUND APPROACH ROUTINE NAME PRINT
C-----CG = OUTBOUND APPROACH ENTITY AND ROUTINE NAME PRINT IPRTLO CHECK
C-----CH = OUTBOUND APPROACH POS/VEL/ACC VS TIME PLOT
C-----CI = OUTBOUND APPROACH POS/VEL/ACC VS TIME PLOT IPRTLO CHECK
C-----CJ = INTERSECTION VEHICLE PRINT FOR CDC
C-----CK = INTERSECTION VEHICLE PRINT FOR IBM
C-----CL = INTERSECTION VEHICLE PRINT TIME CHECK
C-----CM = INTERSECTION VEHICLE PRINT IPRTLO CHECK
C-----CN = INTERSECTION ENTITY PRINT
C-----CO = INTERSECTION ROUTINE NAME PRINT
C-----CP = INTERSECTION ENTITY AND ROUTINE NAME PRINT IPRTLO CHECK
C-----CQ = INTERSECTION POS/VEL/ACC VS TIME PLOT
C-----CR = INTERSECTION POS/VEL/ACC VS TIME PLOT IPRTLO CHECK
C-----CS = INBOUND APPROACH VEHICLE PRINT FOR CDC
C-----CT = INBOUND APPROACH VEHICLE PRINT FOR IBM
C-----CU = INBOUND APPROACH VEHICLE PRINT TIME CHECK
C-----CV = INBOUND APPROACH VEHICLE PRINT IPRTLO CHECK
C-----CW = INBOUND APPROACH ENTITY PRINT
C-----CX = INBOUND APPROACH ROUTINE NAME PRINT
C-----CY = INBOUND APPROACH ENTITY AND ROUTINE NAME PRINT IPRTLO CHECK
C-----CZ = INBOUND APPROACH POS/VEL/ACC VS TIME PLOT
C-----C0 = INBOUND APPROACH POS/VEL/ACC VS TIME PLOT IPRTLO CHECK
C-----C1 = ECHO-PRINT OF INPUT
C-----C2 = ECHO-PRINT OF INPUT IPRTLO CHECK
C-----C3 = FLAG SETTING FOR VEHICLE PRINT FOR CDC
C-----C4 = DEBUG PRINT
C-----C5 = DEBUG PRINT IPRTLO CHECK
C-----C6 = POS/VEL/ACC VS TIME PLOT SETUP
C-----C7 = PAGE PLOT OF POSITION (FOR USE WITH PLTSIM)
```

```
C-----C8 = PRINT MARCH-OUT HEADWAYS ON OUTPUT
C-----C9 = INTERMEDIATE STATISTICS
C-----C/ = DEBUG PRINT FOR SIGNAL (MAINLY ACTUATED INFO)
C-----C¬ = CDC ONLY CODE
C-----C; = IBM ONLY CODE
C
      NRNAME = 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
C-----GET TM TIME FOR THIS JOB AT ITS BEGINING
      CALL  EXTIME  ( 1 )
C-----SET UP DEBUG FILES (CDC ONLY)
C6    IRET = ISLCPF  ( 5LTAPE1,5LTAPE1,IFET1,8,IBUF1,513 )
C6                 IF ( IRET . NE . 0 )          CALL  ABORT ( MSG1 )
C6    IRET = ISLCPF  ( 5LTAPE2,5LTAPE2,IFET2,8,IBUF2,513 )
C6                 IF ( IRET . NE . 0 )          CALL  ABORT ( MSG2 )
C6    IRET = ISLCPF  ( 5LTAPE3,5LTAPE3,IFET3,8,IBUF3,513 )
C6                 IF ( IRET . NE . 0 )          CALL  ABORT ( MSG3 )
C7    IRET = ISLCPF  ( 5LTAPE4,5LTAPE4,IFET4,8,IBUF4,513 )
C7                 IF ( IRET . NE . 0 )          CALL  ABORT ( MSG4 )
      PRINT 601
C-----INITIALIZE THE PARAMETERS FOR THE SIMULATION
      CALL  INITAL
C¬    PRINT 601
C¬    IPAGE = 2
C;    PRINT 601 , ITITLE
C;    IPAGE = 1
C9    ITIM = TSTATS/DT + 0.5
C-----GET TM TIME FOR THIS JOB AT THE END OF INITIALIZATION
      CALL  EXTIME  ( 2 )
C-----GET TM TIME FOR THIS JOB AT THE END OF START-UP TIME
      CALL  EXTIME  ( 3 )
C-----SET RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
C¬    ASSIGN 101 TO NRECAD
C¬    CALL  XMIT    ( NRECAD )
 1010 CONTINUE
C-----SUM THE NUMBER OF VEHICLES IN THE SYSTEM DURING SIMULATION TIME
                   IF ( TIME . GT . STRTIM )    NVSYA = NVSYA + NVSY
      MNVSY = MAX0(MNVSY,NVSY)
C-----GET TM TIME FOR THIS JOB AT THE END OF START-UP TIME
                   IF ( TIME . LE . STRTIM )     CALL  EXTIME  ( 3 )
C-----IF THE TIME INTO THE SIMULATION IS GT THE SIMULATION TIME THEN END
                   IF ( TIME . GT . SIMTIM )    GO TO 4010
C6    WRITE (1,701)  TIME
C6    WRITE (2,701)  TIME
C6    WRITE (3,701)  TIME
C-----DETERMINE WHICH VEHICLES IN THE QUEUE BUFFERS ARE TO BE LOGGED
C-----INTO THE SYSTEM THIS DT
      CALL  QUEUE
C-----IF THERE ARE NO VEHICLES IN THE SYSTEM AND THERE ARE NO VEHICLES
C-----IN THE QUEUE BUFFERS TO BE LOGGED INTO THE SYSTEM THEN END
                   IF ( NVSY+IQF . LE . 0 )      GO TO 4010
C-----IF THERE ARE NO VEHICLES IN THE SYSTEM BUT THERE ARE VEHICLES
C-----IN THE QUEUE BUFFERS TO BE LOGGED INTO THE SYSTEM THEN GO TO 2010
C-----AND PROCESS ONLY THE INBOUND APPROACHES THIS DT
                   IF ( NVSY . LE . 0 )          GO TO 2010
                   IF ( NVOBA . LE . 0 )         GO TO 1020
C-----PROCESS THE VEHICLES ON THE OUTBOUND APPROACHES
      CALL  OBAP
 1020 CONTINUE
                   IF ( NVIN . LE . 0 )          GO TO 2010
C-----PROCESS THE VEHICLES ON THE INTERSECTION PATHS
      CALL  INTERP
 2010 CONTINUE
                   IF ( NVIBA+IQF . LE . 0 )    GO TO 3010
C-----PROCESS THE VEHICLES ON THE INBOUND APPROACHES AND LOG NEW
C-----VEHICLES INTO THE SYSTEM FROM THE QUEUE BUFFERS AS REQUIRED
      CALL  IBAP
 3010 CONTINUE
C-----IF THE INTERSECTION IS PRE-TIMED SIGNAL CONTROLLED THEN SIMULATE
C-----THE PRE-TIMED SIGNAL CONTROLLER
                   IF ( ICONTR . EQ . 5 )        CALL  PRESIG
C-----IF THE INTERSECTION IS SEMI-ACTUATED OR FULL-ACTUATED SIGNAL
C-----CONTROLLED THEN SIMULATE THE SEMI-ACTUATED OR FULL-ACTUATED SIGNAL
C-----CONTROLLER
                   IF ( ICONTR . GE . 6 )        CALL  ACTSIG
C-----IF THE TIME INTO THE SIMULATION IS AN INTEGER MULTIPLE OF THE TIME
C-----INTERVAL FOR INTERMEDIATE STATISTICS THEN PRINT THE INTERMEDIATE
C-----STATISTICS
C9    ITNOW = (TIME-STRTIM)/DT + 0.5
C9          IF ( ((ITNOW/ITIM)*ITIM).EQ.ITNOW )  CALL  INTSTA ( IPAGE )
C-----INCREMENT THE TIME INTO THE SIMULATION AND RECYCLE
      TIME = TIME + DT
      GO TO 1010
 4010 CONTINUE
C-----PRINT THE SUMMARY STATISTICS
      CALL  SUMARY
      RETURN
C-----PROCESS THE SYSTEM ERROR AND STOP (CDC ONLY)
C¬101 CONTINUE
C¬    CALL  ABORTR  ( MSG,22 )
C¬    STOP
C¬102 GO TO NRECAD                                                  *DEBUG*
      END                                                           EXEC
```

```
      SUBROUTINE INITAL
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / I,JTITLE(20),KTITLE(20),ZTEMPD(69)
      DIMENSION         ICOM1(1),ICOM2(1),ICOM3(1)
      EQUIVALENCE       (NVATIN,ICOM1(1)),(TCAMSP(1),ICOM2(1)),
     *                  (TD(1,1),ICOM3(1))
      DATA     N1,N2  / 4HINIT,2HAL /
  501 FORMAT(20A4)
  601 FORMAT(44H ECHO-PRINT OF TITLE FROM GEOMETRY PROCESSOR,/,1X,20A4)
  602 FORMAT(50H0ECHO-PRINT OF TITLE FROM DRIVER-VEHICLE PROCESSOR,/,
     *       1X,20A4)
  603 FORMAT(52H0ECHO-PRINT OF TITLE FROM SIMULATION PROCESSOR INPUT,/,
     *       1X,20A4,//)
C6701 FORMAT(*1 POSITION (20.0 FT PER COL)*/
C6   *       *  CAREQL =*F10.4*  CAREQM =*F10.4*  CAREQA =*F10.4/
C6   *       10X*0       200       400       600       800      1000*
C6   *           *      1200      1400      1600      1800      2000*
C6   *           *      2200      2400*/
C6   *       10X*↓IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓*
C6   *           *IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓IIIIIIIII↓*
C6   *           *IIIIIIIII↓IIIIIIIII↓*)
C6702 FORMAT(*1 VELOCITY (0.5 FT/SEC PER COL)*/
C6   *       *  CAREQL =*F10.4*  CAREQM =*F10.4*  CAREQA =*F10.4/
C6   *       10X*0                   10                   20*
C6   *           *                   30                   40*
C6   *           *                   50                   60*/
C6   *       10X*↓IIIIIIIIIIIIIIIIIII↓IIIIIIIIIIIIIIIIIII↓*
C6   *           *IIIIIIIIIIIIIIIIIII↓IIIIIIIIIIIIIIIIIII↓*
C6   *           *IIIIIIIIIIIIIIIIIII↓IIIIIIIIIIIIIIIIIII↓*)
C6703 FORMAT(*1 ACCELERATION/DECELERATION (0.2 FT/SEC/SEC PER COL)*/
C6   *       *  CAREQL =*F10.4*  CAREQM =*F10.4*  CAREQA =*F10.4/
C6   *       9X*10    9    8    7    6    5    4    3    2    1    0*
C6   *           *   -1   -2   -3   -4   -5   -6   -7   -8   -9  -10*
C6   *           *  -11  -12  -13  -14  -15*/
C6   *       9X* ↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓*
C6   *           *IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓IIII↓*
C6   *           *IIII↓IIII↓IIII↓IIII↓IIII↓*)
  801 FORMAT(50H0END-OF-FILE ON FIRST READ OF GEOPRO INPUT ON TAPE,I2)
  802 FORMAT(49H0END-OF-FILE ON FIRST READ OF DVPRO INPUT ON TAPE,I2)
  803 FORMAT(50H0END-OF-FILE ON FIRST READ OF SIMPRO INPUT ON TAPE,I2)
C
C-----SUBROUTINE INITAL INITIALIZES THE PARAMETERS FOR THE SIMULATION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE COMMON BLOCK INTER
      DO 1010  I = 1 , 212
      ICOM1(I) = 0
 1010 CONTINUE
C-----INITIALIZE COMMON BLOCK SIGCAM
      DO 1020  I = 1 , 1951
      ICOM2(I) = 0
 1020 CONTINUE
      ICAMPC = 1
C-----INITIALIZE COMMON BLOCK SUMSTA (EXCEPT TMTIME)
      DO 1030  I = 1 , 370
      ICOM3(I) = 0
 1030 CONTINUE
C-----READ AND ECHO-PRINT THE TITLE FROM THE GEOMETRY PROCESSOR TAPE
      READ  (IGEOP,501,END=8010)  KTITLE
      PRINT 601 ,                 KTITLE
C-----READ AND ECHO-PRINT THE TITLE FROM THE DRIVER-VEHICLE PROCESSOR
C-----TAPE
      READ  (IVEHP,501,END=8020)  JTITLE
      PRINT 602 ,                 JTITLE
C-----READ AND ECHO-PRINT THE TITLE FROM THE INPUT DIRECTLY TO THE
C-----SIMULATION PROCESSOR
      READ  (INPUT,501,END=8030)  ITITLE
      PRINT 603 ,                 ITITLE
C-----READ THE USER DATA FROM CARD 2 OF THE INPUT DIRECTLY TO THE
C-----SIMULATION PROCESSOR AND CHECK FOR ERRORS
      CALL  RUSERD
C6    WRITE (1,701)  CAREQL,CAREQM,CAREQA
C6    WRITE (2,702)  CAREQL,CAREQM,CAREQA
C6    WRITE (3,703)  CAREQL,CAREQM,CAREQA
C-----READ THE GEOMETRY PROCESSOR DATA FROM THE GEOMETRY PROCESSOR TAPE
C-----AND READ THE LANE CONTROL INFORMATION FROM CARD 3 OF THE INPUT
C-----DIRECTLY TO THE SIMULATION PROCESSOR AND CHECK FOR ERRORS
      CALL  RGEOPD
      NPHASE = 0
C-----IF THE INTERSECTION IS NOT SIGNAL CONTROLLED THEN GO TO 2010 ELSE
C-----READ THE CAM STACK INFORMATION FROM THE INPUT DIRECTLY TO THE
C-----SIMULATION PROCESSOR AND CHECK FOR ERRORS
                      IF ( ICONTR . LT . 5 )        GO TO 2010
      CALL  RCAMSD
C-----IF THE INTERSECTION IS NOT SEMI-ACTUATED OR FULL-ACTUATED SIGNAL
C-----CONTROLLED THEN GO TO 2010 ELSE READ THE SIGNAL PHASE INFORMATION
C-----FROM THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND CHECK FOR
C-----ERRORS
                      IF ( ICONTR . LT . 6 )        GO TO 2010
      CALL  RPHASD
C-----IF NO DETECTORS WERE DECLARED FOR ANY OF THE SEMI-ACTUATED OR
C-----FULL-ACTUATED SIGNAL PHASES THEN GO TO 2010 ELSE READ THE DETECTOR
C-----INFORMATION FROM THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR
C-----AND CHECK FOR ERRORS
                      IF ( NLOOPS . LE . 0 )        GO TO 2010
      CALL  RLOOPD
 2010 CONTINUE
C-----READ THE DRIVER-VEHICLE PROCESSOR DATA FROM THE DRIVER-VEHICLE
C-----PROCESSOR TAPE, INITIALIZE THE QUEUE BUFFERS, AND CHECK FOR ERRORS
      CALL  RDVPRD
      RETURN
C-----PROCESS THE INPUT ERRORS AND STOP
 8010 CONTINUE
      PRINT 801 , IGEOP
      STOP  801
 8020 CONTINUE
      PRINT 802 , IVEHP
      STOP  802
 8030 CONTINUE
      PRINT 803 , INPUT
      STOP  803
      END
```

```
      SUBROUTINE RUSERD
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),JTITLE(20),KTITLE(20),ISTATS,XMPH,
     *                  ZTEMPD(67)
      DATA      IBLNK1 / 1H  /
      DATA      INO    / 2HNO /
      DATA      IYES   / 3HYES /
      DATA      JXXX   / 1HX /
      DATA      N1,N2  / 4HRUSE,2HRD /
  501 FORMAT(F4.2,F6.2,F5.2,2F3.0,2F6.3,F5.0,I2,4(1X,A3),2F5.2,4X,I3,
     *       F6.1,A1)
  502 FORMAT(20A4)
  601 FORMAT(/,
     *54H START-UP TIME (MINUTES) -------------------------- =,F10.2/,
     *54H SIMULATION TIME (MINUTES) ------------------------ =,F10.2/,
     *54H STEP INCREMENT FOR SIMULATION TIME (SECONDS) ------ =,F10.2//,
     *54H SPEED FOR DELAY BELOW XX MPH (MPH) --------------- =,F10.2//,
     *54H MAXIMUM CLEAR DISTANCE FOR BEING IN A QUEUE (FT) -- =,F10.2//,
     *54H CAR FOLLOWING EQUATION LAMBDA ------------------- =,F13.5/,
     *54H CAR FOLLOWING EQUATION MU ----------------------- =,F13.5/,
     *54H CAR FOLLOWING EQUATION ALPHA -------------------- =,F13.5//,
     *54H SUMMARY STATISTICS PRINTED BY TURNING MOVEMENTS --- =,4X,A3/,
     *54H SUMMARY STATISTICS PRINTED BY INBOUND APPROACH ---- =,4X,A3//,
     *54H PUNCHED OUTPUT OF STATISTICS --------------------- =,4X,A3//,
     *54H WRITE TAPE FOR POLLUTION DISPERSION MODEL --------- =,4X,A3//,
     *54H LEAD TIME GAP FOR CONFLICT CHECKING (SECONDS) ----- =,F10.2/,
     *54H LAG TIME GAP FOR CONFLICT CHECKING (SECONDS) ------ =,F10.2//,
     *54H INTERSECTION TRAFFIC CONTROL --------------------- =,I7)
  602 FORMAT(1H+,62X,14H(UNCONTROLLED))
  603 FORMAT(1H+,62X,12H(YIELD SIGN))
  604 FORMAT(1H+,62X,29H(LESS-THAN-ALL-WAY STOP SIGN))
  605 FORMAT(1H+,62X,19H(ALL-WAY STOP SIGN))
  606 FORMAT(1H+,62X,18H(PRE-TIMED SIGNAL))
  607 FORMAT(1H+,62X,22H(SEMI-ACTUATED SIGNAL))
  608 FORMAT(1H+,62X,22H(FULL-ACTUATED SIGNAL))
  701 FORMAT(/,
     *54H TIME INTO SIMULATION FOR DEBUG PRINTING (SECONDS) - =,F10.2,/,
     *54H TIME INTERVAL FOR INTERMEDIATE STATISTICS (SECONDS) =,I7)
  804 FORMAT(16H0START-UP TIME =,F7.2,20H IS LT 2.0 OR GT 5.0)
  805 FORMAT(18H0SIMULATION TIME =,F7.2,22H IS LT 10.0 OR GT 60.0)
  806 FORMAT(37H0STEP INCREMENT FOR SIMULATION TIME =,F7.2,
     *       20H IS LT 0.5 OR GT 1.5)
  807 FORMAT(31H0SPEED FOR DELAY BELOW XX MPH =,F7.2,
     *       21H IS LT 0.0 OR GT 40.0)
  808 FORMAT(46H0MAXIMUM CLEAR DISTANCE FOR BEING IN A QUEUE =,F7.2,
     *       21H IS LT 4.0 OR GT 40.0)
  809 FORMAT(32H0CAR FOLLOWING EQUATION LAMBDA =,F9.5,
     *       20H IS LT 0.0 OR GT 4.0)
  810 FORMAT(28H0CAR FOLLOWING EQUATION MU =,F9.5,
     *       20H IS LT 0.0 OR GT 4.0)
  811 FORMAT(31H0CAR FOLLOWING EQUATION ALPHA =,F9.5,
     *       23H IS LT 0.0 OR GT 9999.9)
  812 FORMAT(31H0INTERSECTION TRAFFIC CONTROL =,I3,16H IS LT 1 OR GT 7)
  813 FORMAT(52H0SUMMARY STATISTICS PRINTED BY TURNING MOVEMENTS = (,A3,
     *       23H) IS NOT (YES) OR (NO ))
  814 FORMAT(51H0SUMMARY STATISTICS PRINTED BY INBOUND APPROACH = (,A3,
     *       23H) IS NOT (YES) OR (NO ))
  815 FORMAT(38H0LEAD TIME GAP FOR CONFLICT CHECKING =,F6.2,
     *       20H IS LT 1.0 OR GT 3.0)
  816 FORMAT(37H0LAG TIME GAP FOR CONFLICT CHECKING =,F6.2,
     *       20H IS LT 1.0 OR GT 3.0)
  817 FORMAT(33H0PUNCHED OUTPUT OF STATISTICS = (,A3,
     *       23H) IS NOT (YES) OR (NO ))
  818 FORMAT(46H0WRITE TAPE FOR POLLUTION DISPERSION MODEL = (,A3,
     *       23H) IS NOT (YES) OR (NO ))
C
C-----SUBROUTINE RUSERD READS THE USER DATA FROM CARD 2 OF THE INPUT
C-----DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----READ THE USER DATA FROM CARD 2 OF THE INPUT DIRECTLY TO THE
C-----SIMULATION PROCESSOR
      READ  (INPUT,501)  STRTIM,SIMTIM,DT,XMPH,XQDIST,CAREQL,CAREQM,
     *                   CAREQA,ICONTR,IPTC,IPAP,IPUNCH,IPOLL,TLEAD,
     *                   TLAG,ISTATS,TPRINT,IXXX
C-----SET THE DEFAULTS FOR THE USER DATA
                      IF ( IPTC . EQ . IBLNK1 )    IPTC = IYES
                      IF ( IPAP . EQ . IBLNK1 )    IPAP = IYES
                      IF ( IPTC . EQ . IYES )      IPAP = IYES
                      IF ( IPUNCH . EQ . IBLNK1 )  IPUNCH = IYES
                      IF ( IPOLL . EQ . IBLNK1 )   IPOLL = INO
C-----ECHO-PRINT THE USER DATA
      PRINT 601 , STRTIM,SIMTIM,DT,XMPH,XQDIST,CAREQL,CAREQM,CAREQA,
     *            IPTC,IPAP,IPUNCH,IPOLL,TLEAD,TLAG,ICONTR
C-----CHECK USER DATA FOR ERRORS
                      IF ( IXXX . EQ . JXXX )        GO TO 1010
                      IF ( STRTIM . LT .     2.0 )   GO TO 8040
                      IF ( STRTIM . GT .     5.0 )   GO TO 8040
                      IF ( SIMTIM . LT .    10.0 )   GO TO 8050
                      IF ( SIMTIM . GT .    60.0 )   GO TO 8050
                      IF ( DT     . LT .     0.5 )   GO TO 8060
                      IF ( DT     . GT .     1.5 )   GO TO 8060
                      IF ( XMPH   . LT .     0.0 )   GO TO 8070
                      IF ( XMPH   . GT .    40.0 )   GO TO 8070
                      IF ( XQDIST . LT .     4.0 )   GO TO 8080
                      IF ( XQDIST . GT .    40.0 )   GO TO 8080
                      IF ( CAREQL . LT .     0.0 )   GO TO 8090
                      IF ( CAREQL . GT .     4.0 )   GO TO 8090
                      IF ( CAREQM . LT .     0.0 )   GO TO 8100
                      IF ( CAREQM . GT .     4.0 )   GO TO 8100
                      IF ( CAREQA . LT .     0.0 )   GO TO 8110
                      IF ( CAREQA . GT .  9999.9 )   GO TO 8110
                      IF ( ICONTR . LT .     1   )   GO TO 8120
                      IF ( ICONTR . GT .     7   )   GO TO 8120
                IF ( IPTC.NE.IYES.AND.IPTC.NE.INO )  GO TO 8130
                IF ( IPAP.NE.IYES.AND.IPAP.NE.INO )  GO TO 8140
                      IF ( TLEAD  . LT .     1.0 )   GO TO 8150
                      IF ( TLEAD  . GT .     3.0 )   GO TO 8150
                      IF ( TLAG   . LT .     1.0 )   GO TO 8160
                      IF ( TLAG   . GT .     3.0 )   GO TO 8160
      IF ( IPUNCH.NE.IYES.AND.IPUNCH.NE.INO    )     GO TO 8170
      IF ( IPOLL.NE.IYES . AND . IPOLL.NE.INO )      GO TO 8180
 1010 CONTINUE
      GO TO ( 1020,1030,1040,1050,1060,1070,1080 ) , ICONTR
 1020 CONTINUE
      PRINT 602
      GO TO 1090
 1030 CONTINUE
      PRINT 603
      GO TO 1090
 1040 CONTINUE
      PRINT 604
      GO TO 1090
 1050 CONTINUE
      PRINT 605
```

```
      GO TO 1090
 1060 CONTINUE
      PRINT 606
      GO TO 1090
 1070 CONTINUE
      PRINT 607
      GO TO 1090
 1080 CONTINUE
      PRINT 608
 1090 CONTINUE
C-----CALCULATE SEVERAL SIMULATION PARAMETERS FROM THE USER DATA
      STRTIM = STRTIM*60.0 + 0.000001
      SIMTIM = SIMTIM*60.0 + STRTIM + 0.000001
      TIME = 0.0
                    IF ( TPRINT . LE . 0.0 )     TPRINT = 999999.99
                    IF ( ISTATS . LE . 0 )       ISTATS = 999999
      IF ( IXXX . EQ . JXXX )
     *PRINT 701 , TPRINT,ISTATS
      DTSQ = DT*DT
      DTCU = DTSQ*DT
      XFPS = XMPH*88.0/60.0
      TSTATS = ISTATS
                    IF ( IPUNCH . NE . IYES )    RETURN
      WRITE (7,502) KTITLE
      WRITE (7,502) JTITLE
      WRITE (7,502) ITITLE
      RETURN
C-----PROCESS THE INPUT ERRORS AND STOP
 8040 CONTINUE
      PRINT 804 , STRTIM
      STOP  804
 8050 CONTINUE
      PRINT 805 , SIMTIM
      STOP  805
 8060 CONTINUE
      PRINT 806 , DT
      STOP  806
 8070 CONTINUE
      PRINT 807 , XMPH
      STOP  807
 8080 CONTINUE
      PRINT 808 , XQDIST
      STOP  808
 8090 CONTINUE
      PRINT 809 , CAREQL
      STOP  809
 8100 CONTINUE
      PRINT 810 , CAREQM
      STOP  810
 8110 CONTINUE
      PRINT 811 , CAREQA
      STOP  811
 8120 CONTINUE
      PRINT 812 , ICONTR
      STOP  812
 8130 CONTINUE
      PRINT 813 , IPTC
      STOP  813
 8140 CONTINUE
      PRINT 814 , IPAP
      STOP  814
 8150 CONTINUE
      PRINT 815 , TLEAD
      STOP  815
 8160 CONTINUE
      PRINT 816 , TLAG
      STOP  816
 8170 CONTINUE
      PRINT 817 , IPUNCH
      STOP  817
 8180 CONTINUE
      PRINT 818 , IPOLL
      STOP  818
      END
```

```
      SUBROUTINE RGEOPD                                                 COLEASE
C     TASK,RGEOPD
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,     COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN   ,     COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA     ,     COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL     ,     COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL     ,     COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL     ,     COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT      ,     COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP    ,     COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),I,IDX,ITEST,IT1,IT2,JA,K,
     *                  LCNTRI(50),NAP,NUM,ZTEMPD(50)
      DIMENSION         IENT1(1),IENT2(1),IENT3(1),IENT4(1),IENT5(1)
      EQUIVALENCE       (NLANES,IENT1(1)),(ICONP(1),IENT2(1)),
     *                  (LWID,IENT3(1)),(LENP,IENT4(1)),
     *                  (ICANSE(1),IENT5(1))
      DATA      N1,N2  / 4HRGEO,2HPD /
  501 FORMAT(20I4)
  502 FORMAT(I4,12X,4I4)
  503 FORMAT(51I1)
  504 FORMAT(8X,I4,/,24X,8I4)
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *       14HLATION PACKAGE,///,1X,20A4,//)
  602 FORMAT(21H LANE CONTROL FOR THE,I3,8H LANES =,50I2)
  603 FORMAT(51H0      WHERE  1 = OUTBOUND (OR BLOCKED INBOUND) LANE,/,
     *       29H              2 = UNCONTROLLED,/,
     *       27H              3 = YIELD SIGN,/,
     *       26H              4 = STOP SIGN,/,
     *       23H              5 = SIGNAL,/,
     *       45H              6 = SIGNAL WITH LEFT TURN ON RED,/,
     *       46H              7 = SIGNAL WITH RIGHT TURN ON RED,//)
C1701 FORMAT(11H A TOTAL OF,I3,32H INBOUND AND OUTBOUND APPROACHES,/)
C1702 FORMAT(11H A TOTAL OF,I3,27H INBOUND AND OUTBOUND LANES,/)
C1703 FORMAT(11H A TOTAL OF,I3,28H SIGHT-DISTANCE RESTRICTIONS,/)
C1704 FORMAT(11H A TOTAL OF,I4,26H PATHS IN THE INTERSECTION,/)
C1705 FORMAT(11H A TOTAL OF,I5,29H INTERSECTION CONFLICT POINTS,/)
C1751 FORMAT(8H APPRO  I3,1X,26I4)
C1752 FORMAT(8H CONFLT I3,1X,12I4)
C1753 FORMAT(8H LANE   I3,1X,28I4)
C1754 FORMAT(8H PATH   I3,1X,10I4,1X,60I1,2I3,2(/,30I4))
C1755 FORMAT(8H SDR    I3,1X,20I4,/,12X,20I4)
  819 FORMAT(37H0LANE CONTROL SPECIFIED FOR MORE THAN,I3,6H LANES)
  820 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,16H IS LT 1 OR GT 7)
  821 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,20H IS EQ 1 FOR INBOUND,
     *       5H LANE)
  822 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,21H IS NE 1 FOR OUTBOUND,
     *       5H LANE)
  823 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,8H IS GT 2,
     *       37H FOR INTERSECTION TRAFFIC CONTROL = 1)
  824 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,8H IS GT 3,
     *       37H FOR INTERSECTION TRAFFIC CONTROL = 2)
  825 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,8H IS GT 4,
     *       37H FOR INTERSECTION TRAFFIC CONTROL = 3)
  826 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,16H IS LT 3 OR GT 4,
```

```
     *       37H FOR INTERSECTION TRAFFIC CONTROL = 4)
  827 FORMAT(5H0LANE,I3,15H LANE CONTROL =,I2,16H IS LT 3 OR EQ 4,
     *       38H FOR INTERSECTION TRAFFIC CONTROL GE 5)
  828 FORMAT(5H LANE,I3,40H SIGNAL WITH LEFT TURN ON RED SPECIFIED ,
     *       26HFOR OTHER THAN MEDIAN LANE)
  829 FORMAT(5H LANE,I3,41H SIGNAL WITH RIGHT TURN ON RED SPECIFIED ,
     *       24HFOR OTHER THAN CURB LANE)
C
C-----SUBROUTINE RGEOPD READS THE GEOMETRY PROCESSOR DATA FROM THE
C-----GEOMETRY PROCESSOR TAPE AND READS THE LANE CONTROL INFORMATION
C-----FROM CARD 3 OF THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND
C-----CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----READ THE ARC INFORMATION TO SCRATCH
      READ  (IGEOP,501)  IT1
                    IF ( IT1 . LE . 0 )          GO TO 1020
      DO 1010  I = 1 , IT1
      READ  (IGEOP,501)  IT2
 1010 CONTINUE
 1020 CONTINUE
C-----READ THE LINE INFORMATION TO SCRATCH
      READ  (IGEOP,501)  IT1
                    IF ( IT1 . LE . 0 )          GO TO 2020
      DO 2010  I = 1 , IT1
      READ  (IGEOP,501)  IT2
 2010 CONTINUE
 2020 CONTINUE
C-----READ THE APPROACH INDEXING INFORMATION
      READ  (IGEOP,501)  NIBA
      READ  (IGEOP,501)  (LIBA(IAN),IAN=1,NIBA)
      DO 3010  IAN = 1 , NIBA
      IA = LIBA(IAN)
      LIBAR(IA) = IAN
 3010 CONTINUE
      READ  (IGEOP,501)  NOBA
      READ  (IGEOP,501)  (LOBA(IAN),IAN=1,NOBA)
      DO 3020  IAN = 1 , NOBA
      IA = LOBA(IAN)
      LOBAR(IA) = IAN
 3020 CONTINUE
C-----READ THE NUMBER OF APPROACHES
      READ  (IGEOP,501)  NAP
C1    PRINT 601 , ITITLE
C1    PRINT 701 , NAP
      NUM = NOATTB(1)
C-----READ THE INFORMATION FOR EACH APPROACH
      DO 3050  I = 1 , NAP
      DO 3030  K = 1 , NUM
      IENT1(K) = 0
 3030 CONTINUE
C-----READ THE APPROACH INFORMATION
      READ  (IGEOP,502)  JA,ISLIM,NLANES,NSDR,IALEFT
      READ  (IGEOP,501)  (LLANES(K),K=1,NLANES)
                    IF ( NSDR . EQ . 0 )         GO TO 3040
      READ  (IGEOP,501)  (ISDRN(K),ISDRA(K),K=1,NSDR)
 3040 CONTINUE
C1    PRINT 751 , JA,(IENT1(K),K=1,NUM)
C-----STORE THE APPROACH INFORMATION IN ENTRY JA OF ENTITY APPRO
C     COLEASE,REPACK,APPRO,JA
      CALL  REPACK  (       1,JA     )                                  COLEASE
C-----END OF APPROACH LOOP
 3050 CONTINUE
C-----READ THE NUMBER OF LANES
      READ  (IGEOP,501)  NRLAN
C-----READ THE LANE CONTROL INFORMATION FROM CARD 3 OF THE INPUT
C-----DIRECTLY TO THE SIMULATION PROCESSOR
      READ  (INPUT,503)  (LCNTRI(I),I=1,NRLAN),ITEST
```

```
      PRINT 601 , ITITLE
      PRINT 602 , NRLAN,(LCNTRI(I),I=1,NRLAN)
      PRINT 603
C1    PRINT 702 , NRLAN
                    IF ( ITEST . NE . 0 )          GO TO 8190
      NUM = NOATTB(3)
C-----READ THE INFORMATION FOR EACH LANE
      DO 4040  I = 1 , NRLAN
      DO 4010  K = 1 , NUM
      IENT3(K) = 0
 4010 CONTINUE
C-----READ THE LANE INFORMATION
      READ  (IGEOP,501)  LWID,LTURN,NPINT,NLL,NLR,ISNA,LGEOM,IDX,IBLN
      LCONTR = LCNTRI(I)
C-----CHECK THE LANE CONTROL FOR ERRORS
                    IF ( LCONTR . LT . 1 )         GO TO 8200
                    IF ( LCONTR . GT . 7 )         GO TO 8200
            IF ( LCONTR.EQ.1.AND.LTURN.NE.0 )      GO TO 8210
            IF ( LCONTR.NE.1.AND.LTURN.EQ.0 )      GO TO 8220
                    IF ( LCONTR . EQ . 1 )         GO TO 4020
            IF ( LGEOM(3) . EQ . LGEOM(4) )        GO TO 4020
            IF ( ICONTR.EQ.1.AND.LCONTR.GT.2 )     GO TO 8230
            IF ( ICONTR.EQ.2.AND.LCONTR.GT.3 )     GO TO 8240
            IF ( ICONTR.EQ.3.AND.LCONTR.GT.4 )     GO TO 8250
            IF ( ICONTR.EQ.4.AND.LCONTR.LT.3 )     GO TO 8260
            IF ( ICONTR.EQ.4.AND.LCONTR.GT.4 )     GO TO 8260
            IF ( ICONTR.GE.5.AND.LCONTR.LT.3 )     GO TO 8270
            IF ( ICONTR.GE.5.AND.LCONTR.EQ.4 )     GO TO 8270
            IF ( LCONTR.EQ.6.AND.NLL.NE.0 )        GO TO 8280
            IF ( LCONTR.EQ.7.AND.NLR.NE.0 )        GO TO 8290
 4020 CONTINUE
      NIBL = MAX0(NIBL,IBLN)
                    IF ( NPINT . EQ . 0 )          GO TO 4030
      READ  (IGEOP,501)  (LINTP(K),K=1,NPINT)
 4030 CONTINUE
C1    PRINT 753 , I,(IENT3(K),K=1,NUM)
C-----STORE THE LANE INFORMATION IN ENTRY I OF ENTITY LANE
C   COLEASE,REPACK,LANE,I
      CALL  REPACK  (      3,I       )                            COLEASE
C-----END OF LANE LOOP
 4040 CONTINUE
C-----READ THE NUMBER OF SIGHT DISTANCE RESTRICTIONS
      READ  (IGEOP,501)  NUMSDR
                    IF ( NUMSDR . LE . 0 )         GO TO 5030
C1    PRINT 601 , ITITLE
C1    PRINT 703 , NUMSDR
      NUM = NOATTB(5)
C-----READ THE INFORMATION FOR EACH SIGHT DISTANCE RESTRICTION
      DO 5020  I = 1 , NUMSDR
      DO 5010  K = 1 , NUM
      IENT5(K) = 0
 5010 CONTINUE
C-----READ THE SIGHT DISTANCE RESTRICTION INFORMATION
      READ  (IGEOP,501)  ICANSE
C1    PRINT 755 , I,(IENT5(K),K=1,NUM)
C-----STORE THE SIGHT DISTANCE RESTRICTION INFORMATION IN ENTRY I OF
C-----ENTITY SDR
C   COLEASE,REPACK,SDR,I
      CALL  REPACK  (      5,I       )                            COLEASE
C-----END OF SIGHT DISTANCE RESTRICTION LOOP
 5020 CONTINUE
 5030 CONTINUE
C-----READ THE NUMBER OF INTERSECTION PATHS
      READ  (IGEOP,501)  NPATHS
C1    PRINT 601 , ITITLE
C1    PRINT 704 , NPATHS
      NUM = NOATTB(4)
C-----READ THE INFORMATION FOR EACH INTERSECTION PATH
      DO 6030  I = 1 , NPATHS
      DO 6010  K = 1 , NUM
      IENT4(K) = 0
 6010 CONTINUE
C-----READ THE INTERSECTION PATH INFORMATION
      READ  (IGEOP,504)  LOBAP,LENP,IPT,LIMP,IOPT,ILCH,LIBL,LOBL,NGEOCP
                    IF ( NGEOCP . EQ . 0 )         GO TO 6020
      READ  (IGEOP,501)  (IGEOCP(K),K=1,NGEOCP)
 6020 CONTINUE
C1    PRINT 754 , I,(IENT4(K),K=1,NUM)
C-----STORE THE INTERSECTION PATH INFORMATION IN ENTRY I OF ENTITY PATH
C   COLEASE,REPACK,PATH,I
      CALL  REPACK  (      4,I       )                            COLEASE
C-----END OF INTERSECTION PATH LOOP
 6030 CONTINUE
C-----READ THE NUMBER OF INTERSECTION CONFLICTS
      READ  (IGEOP,501)  NOCONF
C1    PRINT 601 , ITITLE
C1    PRINT 705 , NOCONF
      NUM = NOATTB(2)
C-----READ THE INFORMATION FOR EACH INTERSECTION CONFLICT
      DO 7020  I = 1 , NOCONF
      DO 7010  K = 1 , NUM
      IENT2(K) = 0
 7010 CONTINUE
C-----READ THE INTERSECTION CONFLICT INFORMATION
      READ  (IGEOP,501)  ICONP,ICONA,ICOND,ICONAN,ICONI
C1    PRINT 752 , I,(IENT2(K),K=1,NUM)
C-----STORE THE INTERSECTION CONFLICT INFORMATION IN ENTRY I OF ENTITY
C-----CONFLT
C   COLEASE,REPACK,CONFLT,I
      CALL  REPACK  (      2,I       )                            COLEASE
C-----END OF INTERSECTION CONFLICT LOOP
 7020 CONTINUE
      RETURN
C-----PROCESS THE INPUT ERRORS AND STOP
 8190 CONTINUE
      PRINT 819 , NRLAN
      STOP  819
 8200 CONTINUE
      PRINT 820 , I,LCONTR
      STOP  820
 8210 CONTINUE
      PRINT 821 , I,LCONTR
      STOP  821
 8220 CONTINUE
      PRINT 822 , I,LCONTR
      STOP  822
 8230 CONTINUE
      PRINT 823 , I,LCONTR
      STOP  823
 8240 CONTINUE
      PRINT 824 , I,LCONTR
      STOP  824
 8250 CONTINUE
      PRINT 825 , I,LCONTR
      STOP  825
 8260 CONTINUE
      PRINT 826 , I,LCONTR
      STOP  826
 8270 CONTINUE
      PRINT 827 , I,LCONTR
      STOP  827
 8280 CONTINUE
      PRINT 828 , I
      STOP  828
 8290 CONTINUE
      PRINT 829 , I
      STOP  829
      END
```

```
      SUBROUTINE RCAMSD
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
C1    COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),I,II,IPHTIM,J,JBLN,JJ,K,KK,LANESS(75),
     *                  MCONTR,NLC,ZTEMPD(24)
      DIMENSION         IISIGN(4),IITURN(3),ISVAL(3,4,3)
      DATA     IBLNK1 / 1H  /
      DATA     IISIGN / 1HG,1HA,1HR,1HP /
      DATA     IITURN / 1HL,1HS,1HR /
      DATA     ILETTA / 1HA /
      DATA     ILETTN / 1HN /
      DATA     ILETTS / 1HS /
      DATA     ILETTU / 1HU /
C-----DATA     ISVAL  / LGG SGG RGG LAG SAG RAG LRG SRG RRG LPG SPG RPG
C-----                  LGA SGA RGA LAA SAA RAA LRA SRA RRA LPA SPA RPA
C-----                  LGR SGR RGR LAR SAR RAR LRR SRR RRR LPR SPR RPR
      DATA     ISVAL  /   1,  1,  1,  7, 13, 19,  9, 15, 21, 23, -1, -1,
     *                    5, 11, 17,  2,  2,  2, 10, 16, 22, 24, -1, -1,
     *                    6, 12, 18,  8, 14, 20,  3,  3,  3, 15, -1, -1/
      DATA     N1,N2  / 4HRCAM,2HSD /
  501 FORMAT(20I4)
  502 FORMAT(I2,I3,75A1)
C1601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
C1   *       14HLATION PACKAGE,//,1X,20A4,//)
  602 FORMAT(11H A TOTAL OF,I3,18H CAM STACK ENTRIES,/)
  603 FORMAT(6H ENTRY,I3,6H PHASE,I3,7H TIME =,I4,1X,25(1X,3A1))
  604 FORMAT(6H ENTRY,I3,6H PHASE,I3,1X,25(1X,3A1))
  830 FORMAT(30H0NUMBER OF CAM STACK ENTRIES =,I4,17H IS LT 4 OR GT 72)
  831 FORMAT(10H0CAM STACK,I3,22H SIGNAL PHASE NUMBER =,I3,
     *       16H IS LT 1 OR GT 8)
  832 FORMAT(10H0CAM STACK,I3,13H PHASE TIME =,I4,8H IS LT 1)
  833 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       20H FIRST CHARACTER = (,A1,28H) IS NOT (L) (S) (R) (A) (U),
     *       7H OR ( ))
  834 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       21H SECOND CHARACTER = (,A1,27H) IS NOT (G) (A) (R) (P) (N,
     *       8H) OR ( ))
  835 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       20H THIRD CHARACTER = (,A1,28H) IS NOT (G) (A) (R) (S) OR ,
     *       3H( ))
  836 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       20H FIRST CHARACTER = (,A1,22H) SECOND CHARACTER = (,A1,
     *       21H) THIRD CHARACTER = (,A1,27H) IS AN ILLEGAL COMBINATION)
  837 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       21H SECOND CHARACTER = (,A1,27H) IS NOT (G) (A) (R) OR (P),
     *       27H WHEN FIRST CHARACTER = (A))
  838 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       47H FIRST CHARACTER = (A) AND SECOND CHARACTER = (,A1,
     *       25H) BUT THIRD CHARACTER = (,A1,12H) IS NOT ( ))
  839 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       47H FIRST CHARACTER = ( ) BUT SECOND CHARACTER = (,A1,
     *       17H) IS NOT ( ) ALSO)
  840 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       49H FIRST CHARACTER = ( ) AND SECOND CHARACTER = ( ),
     *       24H BUT THIRD CHARACTER = (,A1,17H) IS NOT ( ) ALSO)
  841 FORMAT(5H0LANE,I3,13H INBOUND LANE,I3,22H FIRST CHARACTER = ( ),
     *       53H AND SECOND CHARACTER = ( ) AND THIRD CHARACTER = ( ),
     *       16H FOR CAM STACK 1)
  842 FORMAT(10H0CAM STACK,I3,5H LANE,I3,13H INBOUND LANE,I3,
     *       20H FIRST CHARACTER = (,A1,22H) SECOND CHARACTER = (,A1,
     *       21H) THIRD CHARACTER = (,A1,27H) IS ILLEGAL FOR UNSIGNALIZ,
     *       7HED LANE)
```

```
C
C-----SUBROUTINE RCAMSD READS THE CAM STACK INFORMATION FROM THE INPUT
C-----DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----READ THE NUMBER OF CAM STACK POSITIONS
      READ  (INPUT,501)  NCAMSP
C1    PRINT 601 , ITITLE
      PRINT 602 , NCAMSP
                      IF ( NCAMSP . LT .  4 )      GO TO 8300
                      IF ( NCAMSP . GT . 72 )      GO TO 8300
      NLC = NIBL*3
C-----READ THE INFORMATION FOR EACH CAM STACK POSITION
      DO 7020  I = 1 , NCAMSP
C-----READ THE CAM STACK INFORMATION
      READ  (INPUT,502)  ICAMPH(I),IPHTIM,(LANESS(J),J=1,NLC)
                      IF ( ICONTR . GT . 5 )       GO TO 1010
      PRINT 603 , I,ICAMPH(I),IPHTIM,(LANESS(J),J=1,NLC)
      GO TO 1020
 1010 CONTINUE
      PRINT 604 , I,ICAMPH(I),(LANESS(J),J=1,NLC)
 1020 CONTINUE
                      IF ( ICAMPH(I) . LT . 1 )    GO TO 8310
                      IF ( ICAMPH(I) . GT . 8 )    GO TO 8310
               IF ( ICONTR.EQ.5.AND.IPHTIM.LT.1 )  GO TO 8320
      TCAMSP(I) = IPHTIM
      K = 1
C-----CHECK EACH LANE FOR THREE CHARACTER SIGNAL SETTING
      DO 7010  J = 1 , NRLAN
C-----IF THIS IS NOT AN INBOUND LANE THEN SKIP TO THE NEXT LANE
C   COLEASE,FIND,JBLN,LANE,J,IBLN
      CALL  FIND    (JBLN   ,      3,J      ,     27)                      COLEASE
                      IF ( JBLN . EQ . 0 )         GO TO 7010
C   COLEASE,FIND,MCONTR,LANE,J,LCONTR
      CALL  FIND    (MCONTR,       3,J      ,     15)                      COLEASE
C-----IF THE FIRST CHARACTER = ( ) THEN GO TO 5010 AND USE THE SIGNAL
C-----SETTING FROM THE LAST CAM STACK POSITION FOR THIS LANE
                      IF ( LANESS(K).EQ.IBLNK1 )   GO TO 5010
C-----IF THE FIRST CHARACTER = (A) THEN GO TO 4010 AND CHECK THE SECOND
C-----CHARACTER FOR (G) (A) (R) OR (P)
                      IF ( LANESS(K).EQ.ILETTA )   GO TO 4010
C-----IF THE THREE CHARACTERS = (UNS) THEN GO TO 6020 WITH ISISET=0
      IF ( LANESS(K  ) . EQ . ILETTU . AND .
     *     LANESS(K+1) . EQ . ILETTN . AND .
     *     LANESS(K+2) . EQ . ILETTS . AND .
     *     MCONTR      . LT . 5        )           GO TO 6020
C-----CHECK FIRST CHARACTER FOR (L) (S) OR (R)
      DO 1030  II = 1 , 3
               IF ( LANESS(K  ).EQ.IITURN(II) )    GO TO 1040
 1030 CONTINUE
      GO TO 8330
 1040 CONTINUE
C-----CHECK SECOND CHARACTER FOR (G) (A) (R) OR (P)
      DO 2010  JJ = 1 , 4
               IF ( LANESS(K+1).EQ.IISIGN(JJ) )    GO TO 2020
 2010 CONTINUE
      GO TO 8340
 2020 CONTINUE
C-----CHECK THIRD CHARACTER FOR (G) (A) OR (R)
      DO 3010  KK = 1 , 3
               IF ( LANESS(K+2).EQ.IISIGN(KK) )    GO TO 3020
 3010 CONTINUE
      GO TO 8350
 3020 CONTINUE
C-----SET SIGNAL SETTING NUMBER FOR THIS CAM STACK POSITION AND INBOUND
C-----LANE BASED ON THE THREE CHARACTER SIGNAL CODE
      ISISET(I,JBLN) = ISVAL(II,JJ,KK)
                      IF ( ISISET(I,JBLN).LE.0 )   GO TO 8360
```

```
      GO TO 6010
 4010 CONTINUE
C-----CHECK THE SECOND CHARACTER FOR (G) (A) (R) OR (P) WHEN THE FIRST
C-----CHARACTER = (A)
      DO 4020  II = 1 , 4
          IF ( LANESS(K+1).EQ.IISIGN(II) )      GO TO 4030
 4020 CONTINUE
      GO TO 8370
 4030 CONTINUE
                      IF ( LANESS(K+2).NE.IBLNK1 ) GO TO 8380
C-----SET SIGNAL SETTING NUMBER FOR THIS CAM STACK POSITION AND INBOUND
C-----LANE BASED ON THE SECOND CHARACTER WHEN FIRST CHARACTER = (A)
      ISISET(I,JBLN) = II
      GO TO 6010
 5010 CONTINUE
                      IF ( LANESS(K+1).NE.IBLNK1 ) GO TO 8390
                      IF ( LANESS(K+2).NE.IBLNK1 ) GO TO 8400
                      IF ( I . EQ . 1 )            GO TO 8410
C-----SET SIGNAL SETTING NUMBER FOR THIS CAM STACK POSITION AND INBOUND
C-----LANE TO THE SIGNAL SETTING NUMBER FOR THE LAST CAM STACK POSITION
      ISISET(I,JBLN) = ISISET(I-1,JBLN)
 6010 CONTINUE
      IF ( MCONTR.LT.5.AND.ISISET(I,JBLN).NE.0 ) GO TO 8420
 6020 CONTINUE
C-----INCREMENT POINTER FOR NEXT THREE CHARACTERS
      K = K + 3
C-----END OF LANE LOOP
 7010 CONTINUE
C-----END OF CAM STACK INFORMATION LOOP
 7020 CONTINUE
C-----INITIALIZE SIGNAL SETTINGS FOR PRE-TIMED SIGNAL
      ICAMPC = 1
      ICAMPO = NCAMSP
      ICPHAS = ICAMPH(ICAMPC)
      TP = 0.0
      TR = TCAMSP(ICAMPC)
      RETURN
C-----PROCESS INPUT ERRORS AND STOP
 8300 CONTINUE
      PRINT 830 , NCAMSP
      STOP  830
 8310 CONTINUE
      PRINT 831 , I,ICAMPH(I)
      STOP  831
 8320 CONTINUE
      PRINT 832 , I,IPHTIM
      STOP  832
 8330 CONTINUE
      PRINT 833 , I,J,JBLN,LANESS(K)
      STOP  833
 8340 CONTINUE
      PRINT 834 , I,J,JBLN,LANESS(K+1)
      STOP  834
 8350 CONTINUE
      PRINT 835 , I,J,JBLN,LANESS(K+2)
      STOP  835
 8360 CONTINUE
      PRINT 836 , I,J,JBLN,LANESS(K),LANESS(K+1),LANESS(K+2)
      STOP  836
 8370 CONTINUE
      PRINT 837 , I,J,JBLN,LANESS(K+1)
      STOP  837
 8380 CONTINUE
      PRINT 838 , I,J,JBLN,LANESS(K+1),LANESS(K+2)
      STOP  838
 8390 CONTINUE
      PRINT 839 , I,J,JBLN,LANESS(K+1)
      STOP  839
 8400 CONTINUE
      PRINT 840 , I,J,JBLN,LANESS(K+2)
      STOP  840
 8410 CONTINUE
      PRINT 841 , J,JBLN
      STOP  841
 8420 CONTINUE
      PRINT 842 , I,J,JBLN,LANESS(K),LANESS(K+1),LANESS(K+2)
      STOP  842
      END                                                               RCAMSD
```

```
      SUBROUTINE RPHASD
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMRC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSG,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),I,IT1,IUSED(8),J,JP,JPP1,JPP2,K,MCAM,
     *                  N,NCAM,NN,TEST,ZTEMPD(89)
      DATA     IBLNK1 / 1H  /
      DATA     INO    / 2HNO /
      DATA     IOFF   / 3HOFF /
      DATA     ION    / 2HON /
      DATA     IOR    / 2HOR /
      DATA     IYES   / 3HYES /
      DATA     JAND   / 3HAND /
      DATA     N1,N2  / 4HRPHA,2HSD /
  501 FORMAT(20I4)
  502 FORMAT(I2,4F5.1,F6.1,5(1X,A3),2I4,7I2)
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *       14HLATION PACKAGE,//,1X,20A4,//)
  602 FORMAT(11H A TOTAL OF,I2,14H SIGNAL PHASES)
  603 FORMAT(//,45H SEMI-ACTUATED SIGNAL MAIN STREET INFORMATION,/,
     *58H MAIN STREET PHASE NUMBER -------------------------- =    1/,
     *54H MAIN STREET MINIMUM ASSURED GREEN (SECONDS) ------- =F6.1/,
     *54H MAIN STREET AMBER CLEARANCE INTERVAL (SECONDS) ---- =F6.1/,
     *54H MAIN STREET ALL-RED CLEARANCE INTERVAL (SECONDS) -- =F6.1/,
     *54H MAIN STREET NUMBER OF PHASES CLEARED TO ------------=I4/,
     *54H MAIN STREET LIST OF PHASES CLEARED TO --------------=7I4)
  604 FORMAT(//,
     *54H SIGNAL PHASE NUMBER ------------------------------- =I4/,
     *54H INITIAL INTERVAL (SECONDS) ------------------------ =F6.1/,
     *54H VEHICLE INTERVAL (SECONDS) ------------------------ =F6.1/,
     *54H AMBER CLEARANCE INTERVAL (SECONDS) ---------------- =F6.1/,
     *54H ALL-RED CLEARANCE INTERVAL (SECONDS) -------------- =F6.1/,
     *54H MAXIMUM EXTENSION AFTER DEMAND ON RED (SECONDS) --- =F6.1/,
     *54H SKIP-PHASE SWITCH (ON/OFF) ------------------------ =3X,A3/,
     *54H AUTO-RECALL SWITCH (ON/OFF) ----------------------- =3X,A3/,
     *54H PARENT/MINOR MOVEMENT PHASE OPTION (YES/NO) ------- =3X,A3/,
     *54H DUAL LEFT OPTION (YES/NO) ------------------------- =3X,A3/,
     *54H DETECTOR CONNECTION TYPE (AND/OR) ----------------- =3X,A3/,
     *54H NUMBER OF DETECTORS CONNECTED TO PHASE ------------ =I4/,
     *54H NUMBER OF PHASES CLEARED TO ----------------------- =I4/,
     *54H LIST OF PHASES CLEARED TO ------------------------- =7I4)
  605 FORMAT(54H LIST OF DETECTORS CONNECTED TO PHASE -------------- =,
     *       5I4,/,54X,5I4)
  606 FORMAT(34H0PHASE TIMING SET FOR ALL-RED REST)
  607 FORMAT(1H1)
  608 FORMAT(27H0INITIAL INTERVAL FOR PHASE,I2,8H RESET =,F6.1,
     *       45H SECONDS SO THAT DUAL LEFT PHASE WOULD HAVE A,
     *       24H MINIMUM ASSURED GREEN =,F6.1,8H SECONDS)
  609 FORMAT(35H0AMBER CLEARANCE INTERVAL FOR PHASE,I2,
     *       8H RESET =,F6.1,32H SECONDS SO THAT DUAL LEFT PHASE,
     *       29H WOULD HAVE THE MAXIMUM VALUE)
  610 FORMAT(37H0ALL-RED CLEARANCE INTERVAL FOR PHASE,I2,
     *       8H RESET =,F6.1,32H SECONDS SO THAT DUAL LEFT PHASE,
     *       29H WOULD HAVE THE MAXIMUM VALUE)
  611 FORMAT(48H0MAXIMUM EXTENSION AFTER DEMAND ON RED FOR PHASE,I2,
     *       8H RESET =,F6.1,32H SECONDS SO THAT DUAL LEFT PHASE,
     *       29H WOULD HAVE THE MINIMUM VALUE)
  843 FORMAT(26H0NUMBER OF SIGNAL PHASES =,I4,16H IS LT 2 OR GT 8)
  844 FORMAT(22H0SIGNAL PHASE NUMBER =,I2,16H IS LT 1 OR GT 8)
  845 FORMAT(41H0MORE THAN 1 SET OF DATA FOR SIGNAL PHASE,I2)
  846 FORMAT(13H0SIGNAL PHASE,I2,24H IS NOT IN THE CAM STACK)
  847 FORMAT(13H0SIGNAL PHASE,I2,27H AMBER CLEARANCE INTERVAL =,F8.1,
     *       10H IS LT 0.0)
  848 FORMAT(13H0SIGNAL PHASE,I2,29H ALL-RED CLEARANCE INTERVAL =,F8.1,
     *       10H IS LT 0.0)
  849 FORMAT(13H0SIGNAL PHASE,I2,34H MAXIMUM EXTENSION AFTER DEMAND ON,
     *       6H RED =,F8.1,10H IS LT 0.0)
  850 FORMAT(13H0SIGNAL PHASE,I2,22H SKIP PHASE SWITCH = (,A3,
     *       29H) IS NOT (ON ) (OFF) OR (   ))
  851 FORMAT(13H0SIGNAL PHASE,I2,23H AUTO-RECALL SWITCH = (,A3,
     *       29H) IS NOT (ON ) (OFF) OR (   ))
  852 FORMAT(13H0SIGNAL PHASE,I2,24H PARENT/MINOR OPTION = (,A3,
     *       29H) IS NOT (YES) (NO ) OR (   ))
  853 FORMAT(13H0SIGNAL PHASE,I2,21H DUAL LEFT OPTION = (,A3,
     *       29H) IS NOT (YES) (NO ) OR (   ))
  854 FORMAT(13H0SIGNAL PHASE,I2,29H DETECTOR CONNECTION TYPE = (,A3,
     *       29H) IS NOT (AND) (OR ) OR (   ))
  855 FORMAT(13H0SIGNAL PHASE,I2,32H NUMBER OF DETECTORS FOR PHASE =,I4,
     *       17H IS LT 0 OR GT 10)
  856 FORMAT(13H0SIGNAL PHASE,I2,33H IS ACTUATED BUT HAS NO DETECTORS,
     *       35H AND THE AUTO-RECALL SWITCH = (OFF))
  857 FORMAT(13H0SIGNAL PHASE,I2,31H AUTO-RECALL SWITCH = (ON ) BUT,
     *       27H NUMBER OF LOOP DETECTORS =,I3,8H IS NE 0)
  858 FORMAT(13H0SIGNAL PHASE,I2,16H DETECTOR NUMBER,I2,4H = 0)
  859 FORMAT(13H0SIGNAL PHASE,I2,35H POSITIVE CONNECTED DETECTOR IS NOT,
     *       14H FIRST ON LIST)
  860 FORMAT(13H0SIGNAL PHASE,I2,30H NUMBER OF PHASES CLEARED TO =,I4,
     *       16H IS LT 1 OR GT 7)
  861 FORMAT(13H0SIGNAL PHASE,I2,33H DUAL LEFT OPTION = (YES) BUT THE,
     *       30H NUMBER OF PHASES CLEARED TO =,I4,8H IS LT 3)
  862 FORMAT(13H0SIGNAL PHASE,I2,24H CAN NOT CLEAR TO ITSELF)
  863 FORMAT(13H0SIGNAL PHASE,I2,19H PHASE CLEARED TO =,I4,
     *       24H IS NOT IN THE CAM STACK)
  864 FORMAT(13H0SIGNAL PHASE,I2,35H NUMBER OF ENTRIES IN THE CAM STACK,
     *       2H =,I2,47H IS NE 1+(NUMBER OF PHASES CLEARED TO)+(ALL-RED,
     *       3H) =,I2)
  865 FORMAT(13H0SIGNAL PHASE,I2,33H DUAL LEFT OPTION = (YES) BUT THE,
     *       25H FIRST PHASE CLEARED TO =,I2,7H IS NOT,I2)
  866 FORMAT(13H0SIGNAL PHASE,I2,33H DUAL LEFT OPTION = (YES) BUT THE,
     *       26H SECOND PHASE CLEARED TO =,I2,7H IS NOT,I2)
  867 FORMAT(13H0SIGNAL PHASE,I2,35H IS IN THE CAM STACK FOR THE SIGNAL,
     *       30H BUT NO OTHER DATA WAS ENTERED)
  868 FORMAT(13H0SIGNAL PHASE,I2,35H DID NOT HAVE THE ALL-RED REST PHAS,
     *       53HE AS THE LAST PHASE ON ITS LIST OF PHASES TO CLEAR TO)
C
C-----SUBROUTINE RPHASD READS THE SIGNAL PHASE INFORMATION FROM THE
C-----INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                  IF ( NRNAME .GT. NRNAMM )  CALL ABORTR ( MSGR,NR )
      NLOOPS = 0
C-----READ THE NUMBER OF SIGNAL PHASES
      READ  (INPUT,501)  NPHASE
      PRINT 601 , ITITLE
      PRINT 602 , NPHASE
                  IF ( NPHASE .LT. 2 )        GO TO 8430
                  IF ( NPHASE .GT. 8 )        GO TO 8430
      DO 1010  I = 1 , 8
      IUSED(I) = 0
 1010 CONTINUE
C-----READ THE INFORMATION FOR EACH SIGNAL PHASE
      DO 5010  I = 1 , NPHASE
C-----READ THE SIGNAL PHASE INFORMATION
      READ  (INPUT,502)  JP,TII(JP),TVI(JP),TCI(JP),TAR(JP),TMX(JP),
     *                   ISKP(JP),IREC(JP),IMINOR(JP),IDUALL(JP),
```

```
     *              IANDOR(JP),N,NN,(LPHNXT(J,JP),J=1,NN)
C-----SET THE DEFAULTS FOR THE SIGNAL PHASE INFORMATION
                    IF ( ISKP(JP)   .EQ.IBLNK1 )  ISKP(JP)    = IOFF
                    IF ( IREC(JP)   .EQ.IBLNK1 )  IREC(JP)    = IOFF
                    IF ( IMINOR(JP).EQ.IBLNK1 )  IMINOR(JP) = INO
                    IF ( IDUALL(JP).EQ.IBLNK1 )  IDUALL(JP) = INO
                    IF ( IANDOR(JP).EQ.IBLNK1 )  IANDOR(JP) = IOR
                    IF ( ICONTR . EQ . 7 )        GO TO 2010
                    IF ( JP . NE . 1 )            GO TO 2010
C-----SET THE SIGNAL INFORMATION FOR MAIN STREET PHASE OF THE SEMI-
C-----ACTUATED SIGNAL
      TVI(1) = 0.0
      TMX(1) = 0.0
      ISKP(1) = IOFF
      IREC(1) = ION
      IMINOR(1) = INO
      IDUALL(1) = INO
      IANDOR(1) = IOR
      PRINT 603 , TII(1),TCI(1),TAR(1),NN,(LPHNXT(J,1),J=1,NN)
      GO TO 2020
 2010 CONTINUE
      PRINT 604 , JP,TII(JP),TVI(JP),TCI(JP),TAR(JP),TMX(JP),ISKP(JP),
     *            IREC(JP),IMINOR(JP),IDUALL(JP),IANDOR(JP),N,NN,
     *            (LPHNXT(J,JP),J=1,NN)
 2020 CONTINUE
C-----CHECK THE SIGNAL PHASE INFORMATION FOR ERRORS
                    IF ( JP . LT . 1 )            GO TO 8440
                    IF ( JP . GT . 8 )            GO TO 8440
                    IF ( IUSED(JP) . NE . 0 )     GO TO 8450
      IUSED(JP) = 1
C-----FIND THE FIRST CAM STACK POSITION WITH THIS SIGNAL PHASE NUMBER
      DO 2030  J = 1 , NCAMSP
                    IF ( JP . EQ . ICAMPH(J) )    GO TO 2040
 2030 CONTINUE
      GO TO 8460
 2040 CONTINUE
      ICAMPS(JP) = J
C-----SET THE VALUES FOR SEVERAL OF THE SIGNAL PHASE PARAMETERS
      LPHASE(I) = JP
      TII(JP) = AMAX1( TII(JP),DT )
      TVI(JP) = AMAX1( TVI(JP),DT )
      NLD(JP) = N
      NPHNXT(JP) = NN
      NMAXO(JP) = 0
      TMAXO(JP) = 0.0
      NGAPO(JP) = 0
      TGAPO(JP) = 0.0
C-----CHECK THE SIGNAL PHASE INFORMATION FOR ERRORS
                    IF ( TCI(JP) . LT . 0.0 )     GO TO 8470
                    IF ( TAR(JP) . LT . 0.0 )     GO TO 8480
                    IF ( TMX(JP) . LT . 0.0 )     GO TO 8490
      IF ( ISKP(JP).NE.ION.AND.ISKP(JP).NE.IOFF )GO TO 8500
      IF ( IREC(JP).NE.ION.AND.IREC(JP).NE.IOFF )GO TO 8510
      IF ( IMINOR(JP).NE.IYES . AND . IMINOR(JP).NE. INO )
     *                                            GO TO 8520
      IF ( IDUALL(JP).NE.IYES . AND . IDUALL(JP).NE.INO )
     *                                            GO TO 8530
      IF ( IANDOR(JP).NE.JAND . AND . IANDOR(JP).NE.IOR )
     *                                            GO TO 8540
                    IF ( N . LT .  0 )            GO TO 8550
                    IF ( N . GT . 10 )            GO TO 8550
            IF ( IREC(JP).EQ.IOFF.AND.N.EQ.0 )    GO TO 8560
            IF ( IREC(JP).EQ.ION .AND.N.NE.0 )    GO TO 8570
                    IF ( N . EQ .  0 )            GO TO 2060
C-----READ THE LIST OF DETECTORS FOR THIS SIGNAL PHASE
      READ  (INPUT,501)  (LLD(J,JP),J=1,N)
      PRINT 605 , (LLD(J,JP),J=1,N)
      NLOOPS = NLOOPS + N
      ITEST = 0
      DO 2050  J = 1 , N
                    IF ( LLD(J,JP) . EQ . 0 )     GO TO 8580
                    IF ( LLD(J,JP) . GT . 0 )     ITEST = 1
 2050 CONTINUE
      IF ( ITEST.EQ.1 . AND . LLD(1,JP).LT.0 )    GO TO 8590
                    IF ( ITEST . EQ . 1 )         GO TO 2060
C-----SET THE SIGNAL INFORMATION FOR ALL-RED REST PHASE OF THE FULL-
C-----ACTUATED SIGNAL
      IARRPH = JP
      TII(JP) = 0.0
      TVI(JP) = DT
      TCI(JP) = 0.0
      TAR(JP) = 0.0
      TMX(JP) = 0.0
      ISKP(JP) = ION
      IREC(JP) = IOFF
      IMINOR(JP) = INO
      IDUALL(JP) = INO
      IANDOR(JP) = JAND
      PRINT 606
 2060 CONTINUE
                    IF ( NN . LT . 1 )            GO TO 8600
                    IF ( NN . GT . 7 )            GO TO 8600
            IF ( IDUALL(JP).EQ.IYES.AND.NN.LT.3 )GO TO 8610
C-----CHECK TO MAKE SURE THAT THIS PHASE NUMBER IS NOT ON ITS OWN LIST
C-----OF PHASES THAT IT CAN CLEAR TO AND THAT EACH PHASE THAT IT CAN
C-----CLEAR TO IS IN THE CAM STACK
      DO 3020  J = 1 , NN
                    IF ( JP . EQ . LPHNXT(J,JP) )GO TO 8620
      DO 3010  K = 1 , NCAMSP
            IF ( LPHNXT(J,JP) . EQ . ICAMPH(K) ) GO TO 3020
 3010 CONTINUE
      GO TO 8630
 3020 CONTINUE
      IT1 = ICAMPS(JP)
C-----CHECK TO MAKE SURE THAT THERE IS A CAM STACK POSITION FOR THE
C-----GREEN INTERVAL, THE AMBER CLEARANCE INTERVAL FOR EACH PHASE THAT
C-----THIS PHASE CAN CLEAR TO, AND THE ALL-RED CLEARANCE INTERVAL (IF
C-----TAR(JP) GT 0.0) FOR THIS PHASE
      DO 4010  J = IT1 , NCAMSP
                    IF ( ICAMPH(J) . NE . JP )    GO TO 4020
 4010 CONTINUE
      J = NCAMSP + 1
 4020 CONTINUE
      NCAM = J - IT1
      MCAM = 1 + NPHNXT(JP)
                    IF ( TAR(JP) . GT . 0.0 )     MCAM = MCAM + 1
                    IF ( NCAM . NE . MCAM )       GO TO 8640
                    IF ( IDUALL(JP) . NE . IYES ) GO TO 4030
C-----CHECK TO MAKE SURE THAT THE FIRST PHASE THAT THIS PHASE CAN CLEAR
C-----TO IS (THIS PHASE NUMBER+1) WHEN THE DUAL LEFT OPTION = (ON )
      JPP1 = JP + 1
                    IF ( LPHNXT(1,JP).NE.JPP1 )  GO TO 8650
C-----CHECK TO MAKE SURE THAT THE SECOND PHASE THAT THIS PHASE CAN CLEAR
C-----TO IS (THIS PHASE NUMBER+2) WHEN THE DUAL LEFT OPTION = (ON )
      JPP2 = JP + 2
                    IF ( LPHNXT(2,JP).NE.JPP2 )  GO TO 8660
 4030 CONTINUE
            IF ( (I/3)*3.EQ.I.AND.I.NE.NPHASE )  PRINT 607
C-----END OF SIGNAL PHASE LOOP
 5010 CONTINUE
C-----CHECK TO MAKE SURE THAT DATA WAS ENTERED FOR EACH SIGNAL PHASE IN
C-----THE CAM STACK
      DO 6020  I = 1 , NCAMSP
      DO 6010  J = 1 , NPHASE
                    IF ( ICAMPH(I).EQ.LPHASE(J) )GO TO 6020
 6010 CONTINUE
      GO TO 8670
 6020 CONTINUE
C-----CHECK EACH SIGNAL PHASE FOR DUAL LEFT OPTION
      DO 7040  I = 1 , NPHASE
      JP = LPHASE(I)
                    IF ( IDUALL(JP) . NE . IYES )GO TO 7040
```

```
      JPP1 = JP + 1
      JPP2 = JP + 2
C-----CHECK TO MAKE SURE THAT THE MINIMUM ASSURED GREEN (TII+TVI) FOR
C-----THE DUAL LEFT PHASE IS EQ TO THE MINIMUM OF THE MINIMUM ASSURED
C-----GREEN FOR THE FIRST 2 PHASES THAT THIS PHASE CAN CLEAR TO
      TEST = TII(JP)
      TII(JP) = AMIN1(TII(JPP1)+TVI(JPP1),TII(JPP2)+TVI(JPP2)) - TVI(JP)
                    IF ( TEST . EQ . TII(JP) )    GO TO 7010
      TEST = TII(JP) + TVI(JP)
      PRINT 608 , JP,TII(JP),TEST
 7010 CONTINUE
C-----CHECK TO MAKE SURE THAT THE AMBER CLEARANCE INTERVAL FOR THE DUAL
C-----LEFT PHASE IS EQ TO THE MAXIMUM OF THE AMBER CLEARANCE INTERVAL
C-----FOR THE FIRST 2 PHASES THAT THIS PHASE CAN CLEAR TO
      TEST = TCI(JP)
      TCI(JP) = AMAX1(TCI(JPP1),TCI(JPP2))
                    IF ( TEST . EQ . TCI(JP) )    GO TO 7020
      PRINT 609 , JP,TCI(JP)
 7020 CONTINUE
C-----CHECK TO MAKE SURE THAT THE ALL-RED CLEARANCE INTERVAL FOR THE
C-----DUAL LEFT PHASE IS EQ TO THE MAXIMUM OF THE ALL-RED CLEARANCE
C-----INTERVAL FOR THE FIRST 2 PHASES THAT THIS PHASE CAN CLEAR TO
      TEST = TAR(JP)
      TAR(JP) = AMAX1(TAR(JPP1),TAR(JPP2))
                    IF ( TEST . EQ . TAR(JP) )    GO TO 7030
      PRINT 610 , JP,TAR(JP)
 7030 CONTINUE
C-----CHECK TO MAKE SURE THAT THE MAXIMUM EXTENSION AFTER DEMAND ON RED
C-----FOR THE DUAL LEFT PHASE IS EQ TO THE MINIMUM OF THE MAXIMUM
C-----EXTENSION AFTER DEMAND ON RED FOR THE FIRST 2 PHASES THAT THIS
C-----PHASE CAN CLEAR TO
      TEST = TMX(JP)
      TMX(JP) = AMIN1(TMX(JPP1),TMX(JPP2))
                    IF ( TEST . EQ . TMX(JP) )    GO TO 7040
      PRINT 611 , JP,TMX(JP)
C-----END OF DUAL LEFT PHASE LOOP
 7040 CONTINUE
C-----INITIALIZE THE SIGNAL SETTINGS FOR THE ACTUATED SIGNAL
      ICPHAS = LPHASE(1)
      ICAMPC = ICAMPS(ICPHAS)
      ICAMPO = NCAMSP
      TP = 0.0
      TR = TII(ICPHAS) + TVI(ICPHAS)
                    IF ( IARRPH . EQ . 0 )        RETURN
      DO 7050  I = 1 , NPHASE
                    IF ( I . EQ . IARRPH )        GO TO 7050
      N = NPHNXT(I)
             IF ( LPHNXT(N,I) . NE . IARRPH )     GO TO 8680
 7050 CONTINUE
      RETURN
C-----PROCESS THE INPUT ERRORS AND STOP
 8430 CONTINUE
      PRINT 843 , NPHASE
      STOP  843
 8440 CONTINUE
      PRINT 844 , JP
      STOP  844
 8450 CONTINUE
      PRINT 845 , JP
      STOP  845
 8460 CONTINUE
      PRINT 846 , JP
      STOP  846
 8470 CONTINUE
      PRINT 847 , JP,TCI(JP)
      STOP  847
 8480 CONTINUE
      PRINT 848 , JP,TAR(JP)
      STOP  848
 8490 CONTINUE
      PRINT 849 , JP,TMX(JP)
      STOP  849
 8500 CONTINUE
      PRINT 850 , JP,ISKP(JP)
      STOP  850
 8510 CONTINUE
      PRINT 851 , JP,IREC(JP)
      STOP  851
 8520 CONTINUE
      PRINT 852 , JP,IMINOR(JP)
      STOP  852
 8530 CONTINUE
      PRINT 853 , JP,IDUALL(JP)
      STOP  853
 8540 CONTINUE
      PRINT 854 , JP,IANDOR(JP)
      STOP  854
 8550 CONTINUE
      PRINT 855 , JP,N
      STOP  855
 8560 CONTINUE
      PRINT 856 , JP
      STOP  856
 8570 CONTINUE
      PRINT 857 , JP,N
      STOP  857
 8580 CONTINUE
      PRINT 858 , JP,N
      STOP  858
 8590 CONTINUE
      PRINT 859 , JP
      STOP  859
 8600 CONTINUE
      PRINT 860 , JP,NN
      STOP  860
 8610 CONTINUE
      PRINT 861 , JP,NN
      STOP  861
 8620 CONTINUE
      PRINT 862 , JP
      STOP  862
 8630 CONTINUE
      PRINT 863 , JP,LPHNXT(J,JP)
      STOP  863
 8640 CONTINUE
      PRINT 864 , JP,NCAM,MCAM
      STOP  864
 8650 CONTINUE
      PRINT 865 , JP,LPHNXT(1,JP),JPP1
      STOP  865
 8660 CONTINUE
      PRINT 866 , JP,LPHNXT(2,JP),JPP2
      STOP  866
 8670 CONTINUE
      PRINT 867 , ICAMPH(I)
      STOP  867
 8680 CONTINUE
      PRINT 868 , I
      STOP  868
      END                                                               RPHASD
```

```
      SUBROUTINE RLOOPD
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),I,ID,II,ILDLN,IT1,ITEST,IUSED(20),J,
     *                  JJ,JL,K,LDA,LDSTOP,LDSTRT,LGEOM3,LGEOM4,
     *                  LLDLN(6),MLANES,N,NLDL,NLDLN,ZTEMPD(64)
      DATA      IBLNK1 / 1H  /
      DATA      IENCE  / 4HENCE /
      DATA      IPRES  / 4HPRES /
      DATA      IPULS  / 4HPULS /
      DATA      N1,N2  / 4HRLOO,2HPD /
  501 FORMAT(20I4)
  502 FORMAT(I2,1X,2A4,1X,10I4)
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *       14HLATION PACKAGE,//,1X,20A4,//)
  602 FORMAT(11H A TOTAL OF,I3,10H DETECTORS)
  603 FORMAT(//,
     *       31H DETECTOR NUMBER ============ =,I5,/,
     *       31H DETECTOR TYPE ============== =,1X,2A4,/,
     *       31H STARTING POSITION (FEET) === =,I5,/,
     *       31H STOPPING POSITION (FEET) === =,I5,/,
     *       31H APPROACH NUMBER ============ =,I5,/,
     *       31H NUMBER OF LANES ============ =,I5,/,
     *       31H LIST OF LANE NUMBERS ======= =,6I5)
  604 FORMAT(1H1)
  869 FORMAT(22H0NUMBER OF DETECTORS =,I3,17H IS LT 1 OR GT 20)
  870 FORMAT(18H0DETECTOR NUMBER =,I3,17H IS LT 1 OR GT 20)
  871 FORMAT(37H0MORE THAN 1 SET OF DATA FOR DETECTOR,I3)
  872 FORMAT(9H0DETECTOR,I3,18H DETECTOR TYPE = (,2A4,
     *       44H) IS NOT (PULSE   ) (PRESENCE) OR (        ))
  873 FORMAT(9H0DETECTOR,I3,20H STARTING POSITION =,I5,8H IS LT 0)
  874 FORMAT(9H0DETECTOR,I3,20H STOPPING POSITION =,I5,
     *       26H IS LT STARTING POSITION =,I5)
  875 FORMAT(9H0DETECTOR,I3,18H APPROACH NUMBER =,I3,
     *       37H IS NOT ON LIST OF INBOUND APPROACHES)
  876 FORMAT(9H0DETECTOR,I3,25H NUMBER OF LANE NUMBERS =,I4,
     *       16H IS LT 1 OR GT 6)
  877 FORMAT(9H0DETECTOR,I3,14H LANE NUMBER =,I4,
     *       43H IS LT 1 OR GT NUMBER OF LANES FOR APPROACH,I3,2H =,I2)
  878 FORMAT(9H0APPROACH,I3,29H NUMBER OF DETECTORS FOR LANE,I2,2H =,I2,
     *       8H IS GT 5)
  879 FORMAT(9H0DETECTOR,I3,9H APPROACH,I3,5H LANE,I2,
     *       37H IS NOT AVAILABLE AT THE INTERSECTION)
  880 FORMAT(9H0DETECTOR,I3,20H STOPPING POSITION =,I5,
     *       31H IS GT END OF LANE FOR APPROACH,I3,5H LANE,I2,2H =,I5)
  881 FORMAT(9H0DETECTOR,I3,34H IS ON LIST OF DETECTORS FOR PHASE,I2,
     *       30H BUT NO OTHER DATA WAS ENTERED)
  882 FORMAT(9H0DETECTOR,I3,36H DATA WAS ENTERED BUT DID NOT APPEAR,
     *       51H ON THE LIST OF DETECTORS FOR ANY PHASE AS POSITIVE)
C
C-----SUBROUTINE RLOOPD READS THE DETECTOR INFORMATION FROM THE INPUT
C-----DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----READ THE NUMBER OF DETECTORS
      READ  (INPUT,501)  NLOOPS
      PRINT 601 , ITITLE
      PRINT 602 , NLOOPS
                    IF ( NLOOPS . LT .  1 )       GO TO 8690
                    IF ( NLOOPS . GT . 20 )       GO TO 8690
      DO 1010  I = 1 , 20
      IUSED(I) = 0
 1010 CONTINUE
C-----READ THE INFORMATION FOR EACH DETECTOR
      DO 2030  I = 1 , NLOOPS
C-----READ THE DETECTOR INFORMATION
      READ  (INPUT,502)  ID,ITYPLD(ID),IT1,LDSTRT,LDSTOP,LDA,
     *                   NLDLN,(LLDLN(K),K=1,NLDLN)
C-----SET THE DEFAULTS FOR THE DETECTOR INFORMATION
                    IF ( ITYPLD(ID).NE.IBLNK1 )  GO TO 2010
      ITYPLD(ID) = IPRES
      IT1        = IENCE
 2010 CONTINUE
      PRINT 603 , ID,ITYPLD(ID),IT1,LDSTRT,LDSTOP,LDA,
     *            NLDLN,(LLDLN(K),K=1,NLDLN)
C-----CHECK THE DETECTOR INFORMATION FOR ERRORS
                    IF ( ID . LT .  1 )           GO TO 8700
                    IF ( ID . GT . 20 )           GO TO 8700
                    IF ( IUSED(ID) . NE . 0 )     GO TO 8710
      IUSED(ID) = 1
      IF ( ITYPLD(ID).NE.IPULS . AND . ITYPLD(ID).NE.IPRES )
     *                                            GO TO 8720
                    IF ( LDSTRT . LT . 0 )        GO TO 8730
                    IF ( LDSTOP . LT . LDSTRT )   GO TO 8740
      STRTLD(ID) = LDSTRT
      STOPLD(ID) = LDSTOP
      LLOOPS(I) = ID
      LDTRIP(ID) = .FALSE.
C-----CHECK TO MAKE SURE THAT THE DETECTOR APPROACH NUMBER IS ON THE
C-----LIST OF INBOUND APPROACHES
                    IF ( LIBAR(LDA) . LE . 0 )   GO TO 8750
                    IF ( NLDLN . LT . 1 )        GO TO 8760
                    IF ( NLDLN . GT . 6 )        GO TO 8760
C     COLEASE,FIND,MLANES,APPRO,LDA,NLANES
      CALL  FIND    (MLANES,      1,LDA   ,       1)                    COLEASE
C-----PROCESS EACH LANE THAT THE DETECTOR OCCUPIES
      DO 2020  K = 1 , NLDLN
      ILDLN = LLDLN(K)
                    IF ( ILDLN . LT . 1 )        GO TO 8770
                    IF ( ILDLN . GT . MLANES )   GO TO 8770
C     COLEASE,FIND,JL,APPRO,LDA,LLANES(ILDLN)
      CALL  FIND    (JL     ,     1,LDA   ,       1+ILDLN )             COLEASE
C-----ADD THE DETECTOR FOR LANE JL
C     COLEASE,FIND,NLDL,LANE,JL,NLDL
      CALL  FIND    (NLDL   ,     3,JL    ,      21)                    COLEASE
      NLDL = NLDL + 1
                    IF ( NLDL . GT . 5 )         GO TO 8780
C     COLEASE,STORE,NLDL,LANE,JL,NLDL
      CALL  STORE   (NLDL   ,     3,JL    ,      21)                    COLEASE
C     COLEASE,STORE,ID,LANE,JL,LLDL(NLDL)
      CALL  STORE   (ID     ,     3,JL    ,      21+NLDL  )             COLEASE
C     COLEASE,FIND,LGEOM3,LANE,JL,LGEOM(3)
      CALL  FIND    (LGEOM3,      3,JL    ,      19)                    COLEASE
C     COLEASE,FIND,LGEOM4,LANE,JL,LGEOM(4)
      CALL  FIND    (LGEOM4,      3,JL    ,      20)                    COLEASE
                    IF ( LGEOM3 . EQ . LGEOM4 )  GO TO 8790
                    IF ( LDSTOP . GT . LGEOM4 )  GO TO 8800
C-----END OF LANE LOOP
 2020 CONTINUE
             IF ( (I/6)*6.EQ.I.AND.I.NE.NLOOPS )  PRINT 604
C-----END OF DETECTOR LOOP
 2030 CONTINUE
C-----CHECK EACH SIGNAL PHASE TO MAKE SURE THAT DATA WAS ENTERED FOR
C-----EACH DETECTOR THAT WAS DECLARED FOR THAT SIGNAL PHASE
      DO 3030  II = 1 , NPHASE
```

```
      I = LPHASE(II)
      N = NLD(I)
                        IF ( N . LE . 0 )             GO TO 3030
      DO 3020  J = 1 , N
      JL = IABS(LLD(J,I))
      DO 3010  K = 1 , NLOOPS
                        IF ( JL . EQ . LLOOPS(K) )    GO TO 3020
 3010 CONTINUE
      GO TO 8810
 3020 CONTINUE
 3030 CONTINUE
C-----CHECK EACH DETECTOR TO MAKE SURE THAT IT APPEARED ON AT LEAST ONE
C-----OF THE LIST OF DETECTORS FOR A SIGNAL PHASE AS POSITIVE
      DO 4030  I = 1 , NLOOPS
      JL = LLOOPS(I)
      DO 4020  JJ = 1 , NPHASE
      J = LPHASE(JJ)
      N = NLD(J)
                        IF ( N . LE . 0 )             GO TO 4020
      DO 4010  K = 1 , N
                        IF ( JL . EQ . LLD(K,J) )     GO TO 4030
 4010 CONTINUE
 4020 CONTINUE
      GO TO 8820
 4030 CONTINUE
      RETURN
C-----PROCESS THE INPUT ERRORS AND STOP
 8690 CONTINUE
      PRINT 869 , NLOOPS
      STOP  869
 8700 CONTINUE
      PRINT 870 , JL
      STOP  870
 8710 CONTINUE
      PRINT 871 , JL
      STOP  871
 8720 CONTINUE
      PRINT 872 , JL,ITYPLD(JL),IT1
      STOP  872
 8730 CONTINUE
      PRINT 873 , JL,LDSTRT
      STOP  873
 8740 CONTINUE
      PRINT 874 , JL,LDSTOP,LDSTRT
      STOP  874
 8750 CONTINUE
      PRINT 875 , JL,LDA
      STOP  875
 8760 CONTINUE
      PRINT 876 , JL,NLDLN
      STOP  876
 8770 CONTINUE
      PRINT 877 , JL,ILDLN,LDA,MLANES
      STOP  877
 8780 CONTINUE
      PRINT 878 , LDA,ILDLN,NLDL
      STOP  878
 8790 CONTINUE
      PRINT 879 , JL,LDA,ILDLN
      STOP  879
 8800 CONTINUE
      PRINT 880 , JL,LDSTOP,LDA,ILDLN,LGEOM4
      STOP  880
 8810 CONTINUE
      PRINT 881 , JL,I
      STOP  881
 8820 CONTINUE
      PRINT 882 , JL
      STOP  882
      END                                                             RLOOPD
```

```
      SUBROUTINE RDVPRD                                               COLEASE
C     TASK,RDVPRD
      COMMON / LOGICV / LTRUE,LFALSE                                  COLEASE
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IUF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VINITA(1),I,IAMAX(15),IDCHAR(5),IDMAX(15),
     *                  IVCHAR(15),IVMAX(15),J,NDRICL,NVEHCL,PIJRMI,
     *                  ZTEMPD(39)
      DATA      N1,N2  / 4HRDVP,2HRD /
  501 FORMAT(20I4)
  502 FORMAT(6F5.1)
  503 FORMAT(F10.2,7I5)
C1701 FORMAT(38H1LENGTH OF VEHICLES (FT) --------------15I6)
C7702 FORMAT(38H LENGTH OF VEHICLES (FT) --------------15I6)
C1703 FORMAT(38H VEHICLE OPERATIONAL FACTOR -----------15I6)
C1704 FORMAT(38H MAXIMUM DECELERATION (FT/SEC/SEC) ---15I6)
C1705 FORMAT(38H MAXIMUM ACCELERATION (FT/SEC/SEC) ---15I6)
C1706 FORMAT(38H MAXIMUM VELOCITY (FT/SEC) ------------15I6)
C1707 FORMAT(38H MINIMUM TURNING RADIUS (FT) ----------15I6)
C1708 FORMAT(38H DRIVER OPERATIONAL FACTOR ------------5I6)
C1709 FORMAT(38H DRIVER REACTION TIME (SEC) -----------6F6.1)
C1710 FORMAT()
C1711 FORMAT(13H QUEUE BUFFERI3,9H  VEHICLEI5,10H  READIN =F10.2,7I5)
  883 FORMAT(15H0AVERAGE PIJR =,F4.1,21H IS LT MINIMUM PIJR =,F4.1)
C
C-----SUBROUTINE RDVPRD READS THE DRIVER-VEHICLE PROCESSOR DATA FROM THE
C-----DRIVER-VEHICLE PROCESSOR TAPE, INITIALIZES THE QUEUE BUFFERS, AND
C-----CHECKS FOR ERRORS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                       IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      IEF = LFALSE
C-----READ THE NUMBER OF VEHICLE AND DRIVER CLASSES
      READ  (IVEHP,501)  NVEHCL,NDRICL
C-----READ AND ECHO-PRINT THE VEHICLE CHARACTERISTICS
      READ  (IVEHP,501)  (LENV(I),I=1,NVEHCL)
C1    PRINT 701 ,        (LENV(I),I=1,NVEHCL)
C7    WRITE (4,702)      (LENV(I),I=1,NVEHCL)
      READ  (IVEHP,501)  (IVCHAR(I),I=1,NVEHCL)
C1    PRINT 703 ,        (IVCHAR(I),I=1,NVEHCL)
      READ  (IVEHP,501)  (IDMAX(I),I=1,NVEHCL)
C1    PRINT 704 ,        (IDMAX(I),I=1,NVEHCL)
      READ  (IVEHP,501)  (IAMAX(I),I=1,NVEHCL)
C1    PRINT 705 ,        (IAMAX(I),I=1,NVEHCL)
      READ  (IVEHP,501)  (IVMAX(I),I=1,NVEHCL)
C1    PRINT 706 ,        (IVMAX(I),I=1,NVEHCL)
      READ  (IVEHP,501)  (IRMIN(I),I=1,NVEHCL)
C1    PRINT 707 ,        (IRMIN(I),I=1,NVEHCL)
C-----READ AND ECHO-PRINT THE DRIVER CHARACTERISTICS
      READ  (IVEHP,501)  (IDCHAR(I),I=1,NDRICL)
C1    PRINT 708 ,        (IDCHAR(I),I=1,NDRICL)
      READ  (IVEHP,502)  (PIJR(I),I=1,NDRICL),APIJR
C1    PRINT 709 ,        (PIJR(I),I=1,NDRICL),APIJR
C1    PRINT 710
      DCHARM = 0.0
      PIJRMI = 10.0
      TLEAD = TLEAD - APIJR
      TLAG  = TLAG  - APIJR
C-----COMPUTE DRIVER PARAMETERS FOR THE SIMULATION
      DO 1010  I = 1 , NDRICL
      DCHAR(I) = IDCHAR(I)/100.0
      DCHARM = AMAX1(DCHARM,DCHAR(I))
      PIJRMI = AMIN1(PIJR(I),PIJRMI)
```

```
      IPIJR(I) = MAX0(IFIX(PIJR(I)/DT+0.5),1)
      PIJR(I) = IPIJR(I)*DT
 1010 CONTINUE
                       IF ( APIJR . LT . PIJRMI )    GO TO 8830
C-----COMPUTE VEHICLE PARAMETERS FOR THE SIMULATION
      DO 1020  I = 1 , NVEHCL
      DMAX(I) = -DUTOL*IDMAX(I)
      AMAX(I) = AUTOL*IAMAX(I)
      VMAX(I) = IVMAX(I)
      VCHAR(I) = IVCHAR(I)/100.0
 1020 CONTINUE
C-----INITIALIZE THE QUEUE BUFFERS
      DO 2010  I = 1 , 25
C-----READ THE DRIVER-VEHICLE INFORMATION; IF END-OF-FILE THEN GO TO
C-----2020 AND SET IEF FLAG
      READ  (IVEHP,503,END=2020)  QTIME(I),(IBUF(I,J),J=1,7)
      IBUF(I,8) = NUMV
      NUMV = NUMV + 1
C2                     IF ( IBUF(I,7) . EQ . 0 )     GO TO 101
C1                     IF ( TIME . LT . TPRINT )     GO TO 101
C1    PRINT 711 , I,IBUF(I,8),QTIME(I),(IBUF(I,J),J=1,7)
C1101 CONTINUE
C-----INCREMENT THE NUMBER OF VEHICLES IN THE QUEUE BUFFERS
      IQF = IQF + 1
C-----END OF QUEUE BUFFER LOOP
 2010 CONTINUE
      I = 25
      RETURN
 2020 CONTINUE
C-----SET END-OF-FILE FLAG AND FLAG QUEUE BUFFER I UNUSED
      IEF = LTRUE
      QTIME(I) = -1.0
      RETURN
C-----PROCESS THE INPUT ERROR AND STOP
 8830 CONTINUE
      PRINT 883 , APIJR,PIJRMI
      STOP  883
      END                                                               RDVPRD
```

```
      SUBROUTINE QUEUE
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / IB,JA,JAN,JLN,ZTEMPD(106)
      DATA      N1,N2   / 4HQUEU,2HE   /
C
C-----SUBROUTINE QUEUE DETERMINES WHICH VEHICLES IN THE QUEUE BUFFER
C-----ARE TO BE LOGGED INTO THE SYSTEM THIS DT
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                       IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----CHECK EACH QUEUE BUFFER TO DETERMINE WHICH VEHICLES ARE TO BE
C-----LOGGED INTO THE SYSTEM THIS DT
      DO 1010  IB = 1 , 25
C-----IF QTIME IS NEGATIVE OR QTIME IS GT THE TIME INTO THE SIMULATION
C-----THEN SKIP TO THE NEXT QUEUE BUFFER
                       IF ( QTIME(IB) . LT . 0.0 )  GO TO 1010
                       IF ( QTIME(IB) . GT . TIME ) GO TO 1010
C-----SET THE QUEUE BUFFER INDEX FOR THE INBOUND APPROACH AND LANE THAT
C-----THE VEHICLE IS TO LOG INTO
      JLN = IBUF(IB,6)
      JA = IBUF(IB,5)
      JAN = LIBAR(JA)
      LQ(JAN,JLN) = IB
 1010 CONTINUE
      RETURN
      END                                                               QUEUE
```

```
      SUBROUTINE OBAP                                                   COLEASE
C     TASK,OBAP
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,     COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA     ,     COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL     ,     COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL     ,     COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,     COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL    ,     COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,     COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,     COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,     COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,     COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,     COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,     COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,     COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF      ,     COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN    ,     COLEASE
     *                  IBAPS     ,IPRTLD    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IUF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPULL
      DIMENSION         IENT1(1),IENT3(1),IENT6(1),IENT7(1)
      EQUIVALENCE       (NLANES,IENT1(1)),(LWID,IENT3(1)),
     *                  (ISLP,IENT6(1)),(IDRICL,IENT7(1))
C7    DATA      IONE    / 1 /
CA    DATA      IZERO   / 0 /
CB    DATA      IZERO   / 0 /
      DATA      N1,N2   / 4HOBAP,2H   /
CA701 FORMAT(35H0SUMMARY FOR OUTBOUND APPROACHES AT,F8.2,8H SECONDS,/,
CA   *        51H AP LN VEH  NUM NUF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CA   *        52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT LPOS   SIG)
CB701 FORMAT(35H0SUMMARY FOR OUTBOUND APPROACHES AT,F8.2,8H SECONDS,/,
CB   *        51H AP LN VEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CB   *        52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT LPOS   SIG)
C7702 FORMAT(F7.2,5I4,2F7.1)
CA703 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,2X,I4,
CA   *        3(1X,A10))
CB703 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,2X,I4)
CF704 FORMAT(18(1X,A4,A2))
CE751 FORMAT(8H APPRO  I3,1X, 26I4)
CE753 FORMAT(8H LANE   I3,1X, 28I4)
CE756 FORMAT(8H VEHD   I3,2(I5,I6),3I2,2I3,2I5,I7,2I5,I3,3I4,I6,I2,I4,
CE   *                 I3,2X,11I1,2X,7I1)
CE757 FORMAT(8H VEHF   I3,1X, 12I4)
C
C-----SUBROUTINE OBAP PROCESSES THE VEHICLES ON THE OUTBOUND APPROACHES
C
      NRNAME = 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
CG    IHPRT = LFALSE
CG                    IF ( IHPRT . EQ . LFALSE )    GO TO 101
CC                    IF ( TIME . LT . TPRINT )     GO TO 101
CA    PRINT 701 , TIME
CB    PRINT 701 , TIME
CC101 CONTINUE
      IGO = 0
C-----PROCESS EACH OUTBOUND APPROACH
      DO 6010  IAN = 1 , NOBA
      IA = LOBA(IAN)
C-----IF THERE ARE NO VEHICLES ON THIS OUTBOUND APPROACH THEN SKIP TO
C-----THE NEXT OUTBOUND APPROACH
                      IF ( NVIA(IA) . LE . 0 )      GO TO 6010
C-----EXTRACT OUTBOUND APPROACH IA
C     COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (      1,IA     )                                   COLEASE
CG    IAPRT = LFALSE
CG                    IF ( IAPRT . EQ . LFALSE )    GO TO 102
CE                    IF ( TIME . LT . TPRINT )     GO TO 102
CE    NUM = NOATTB(1)
CE    PRINT 751 , IA,(IENT1(I),I=1,NUM)
CE102 CONTINUE
C-----PROCESS EACH LANE ON THE OUTBOUND APPROACH
      DO 5010  ILN = 1 , NLANES
C-----IF THERE ARE NO VEHICLES IN THIS LANE THEN SKIP TO THE NEXT LANE
                      IF ( NVIL(ILN) . LE . 0 )     GO TO 5010
      IL = LLANES(ILN)
      LCHGE = 1
C-----EXTRACT LANE IL
C     COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (      3,IL     )                                   COLEASE
CG    ILPRT = LFALSE
CG                    IF ( ILPRT . EQ . LFALSE )    GO TO 103
CE                    IF ( TIME . LT . TPRINT )     GO TO 103
CE    NUM = NOATTB(3)
CE    PRINT 753 , ILN,(IENT3(I),I=1,NUM)
CE103 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES OF THE FIRST VEHICLE ON
C-----THE LANE
      IV = IFVL
      NV = NVIL(ILN)
C-----PROCESS EACH VEHICLE ON THE LANE
      DO 4010  IVN = 1 , NV
      NRNAME = 1
      ENDLN = 5000.0
C-----EXTRACT ENTRY IV OF ENTITY VEHF, RESET THE PREVIOUS VEHICLE
C-----PARAMETERS TO THE NEW NOF IF THE VEHICLE IS LANE CHANGING, AND
C-----INITIALIZE SEVERAL PARAMETERS FOR THE VEHICLE
      CALL  PREST1  ( LFALSE )
                      IF ( MFINL . EQ . LFALSE )    GO TO 1010
C-----THIS VEHICLE IS THE FIRST VEHICLE IN THE LANE THUS RESET THE
C-----PREVIOUS VEHICLE PARAMETERS
      PVPOS = ENDLN
      PVVEL = 0.0
      PVACC = 0.0
 1010 CONTINUE
C-----COMPUTE NEW ACC/DEC LOGIC AND EXTRACT ENTRY IV OF ENTITY VEHD FOR
C-----THE VEHICLE
      CALL  PREST2
CG                    IF ( IPRTLD . EQ . 0 )        GO TO 107
CE                    IF ( TIME . LT . TPRINT )     GO TO 107
CG                    IF ( IHPRT . EQ . LTRUE )     GO TO 104
CG    PRINT 701 , TIME
CG    IHPRT = LTRUE
CG104 CONTINUE
CG                    IF ( IAPRT . EQ . LTRUE )     GO TO 105
CG    NUM = NOATTB(1)
CG    PRINT 751 , IA,(IENT1(I),I=1,NUM)
CG    IAPRT = LTRUE
CG105 CONTINUE
CG                    IF ( ILPRT . EQ . LTRUE )     GO TO 106
```

```
CG     NUM = NOATTB(3)
CG     PRINT 753 , ILN,(IENT3(I),I=1,NUM)
CG     ILPRT = LTRUE
CG106  CONTINUE
CE     NUM = NOATTB(7)
CE     PRINT 757 , IV,(IENT7(I),I=1,NUM)
CE     NUM = NOATTB(6)
CE     PRINT 756 , IV,(IENT6(I),I=1,NUM)
CE107  CONTINUE
                     IF ( LALT . NE . 6 )          GO TO 2010
C-----THIS VEHICLE HAS ALREADY BEEN PROCESSED IN THIS DT THUS RESET THE
C-----PREVIOUS VEHICLE PARAMETERS AND SKIP TO THE NEXT VEHICLE
      LALT = 5
      PVPOS = IPOS/25.0 - LENV(IVEHCL) - 4.0
      PVVEL = IVEL/25.0
      PVACC = IACC/312.5 - 32.0
      NXVEH = NOR
      GO TO 3020
 2010 CONTINUE
C-----UNBIAS THE VEHICLE ATTRIBUTES AND PREDICT THE NEW POS/VEL/ACC
      CALL  UNBIAS
      NXVEH = NOR
                     IF ( ISET . NE . 1 )          GO TO 2020
C-----COMPUTE THE NEW LATERAL POSITION FOR A LANE CHANGE USING A COSINE
C-----CURVE AND IF FINISHED THEN END THE LANE CHANGE
      CALL  LCHGEO
              IF ( ISET.NE.1.AND.MBLOCK.EQ.LFALSE )ISET = 6
      GO TO 2030
 2020 CONTINUE
                     IF ( ISET . GE . 6 )          GO TO 2030
C-----DETERMINE IF A LANE CHANGE IS DESIRABLE
      CALL  LCHDES
 2030 CONTINUE
C-----CHECK THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES, CALL THE
C-----APPROPRIATE ACC/DEC ROUTINES, AND COMPUTE THE VEHICLES NEW POS/
C-----VEL/ACC
      CALL  ACDCP
C7     POSLAT = LATPOS/8.0 - 15.0
C7                   IF ( LCHGE . NE . 2 )         POSLAT = 0.0
C7     IF ( ABS(LEGAL/2.0-ABS(POSLAT)).LE.0.1 )    POSLAT = 0.0
C7     WRITE (4,702)  TIME,IQ(IV),IONE,IA,IL,IVEHCL,POSNEW,POSLAT
CI                   IF ( IPRTLO . EQ . 0 )        GO TO 108
C-----PRINT POS/VEL/ACC FOR THE VEHICLE
CH     CALL  PVAPRT
CI108 CONTINUE
C-----IF THE VEHICLE LEFT THE OUTBOUND APPROACH THEN GO TO 3030 AND LOG
C-----THE VEHICLE OUT OF THE SYSTEM
              IF ( POSNEW . GT . FLOAT(LGEOM(4)) ) GO TO 3030
C-----UPDATE THE VEHICLES SIMULATION STATISTICS ON THE OUTBOUND APPROACH
      CALL  SSOBAP
                     IF ( MFINL . EQ . LTRUE )     GO TO 3010
                     IF ( PVPOS+4.0.GT.POSNEW )    GO TO 3010
C-----PRINT THE COLLISION INFORMATION AND RESET THE VEHICLES POS/VEL/ACC
      CALL  BANGS   ( 3 )
 3010 CONTINUE
C-----BIAS THE VEHICLE ATTRIBUTES, SET THE PREVIOUS VEHICLE PARAMETERS,
C-----AND UPDATE THE MAXIMUM ACC/DEC FOR THE VEHICLE
      CALL  BIAS
C-----PRINT SELECTED ATTRIBUTES FOR THE VEHICLE
                     IF ( JPRTM . NE . 0 )         IPRTM = JPRTM
CD                   IF ( IPRTLO . EQ . 0 )        GO TO 109
CC                   IF ( TIME . LT . TPRINT )     GO TO 109
C3                   IF ( JPRTM . GT . 0 )         JPFLAG = 10HPIJR TIME
CA     IDESPD = DESVEL + 0.5
CA     POSLAT = LATPOS/8.0 - 15.0
CA                   IF ( LCHGE . NE . 2 )         POSLAT = 0.0
CA     PRINT 703 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
CA    *              SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
CA    *              LOGFLG,LCHGE,IPRTM,POSLAT,IZERO,IPFLAG,JPFLAG,KPFLAG
CB     IDESPD = DESVEL + 0.5
CB     POSLAT = LATPOS/8.0 - 15.0
CB                   IF ( LCHGE . NE . 2 )         POSLAT = 0.0
CB     PRINT 703 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
CB    *              SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISFT,LEGAL,
CB    *              LOGFLG,LCHGE,IPRTM,POSLAT,IZERO
CC109 CONTINUE
 3020 CONTINUE
C-----REPACK THE ATTRIBUTES FOR VEHICLE IV
C   COLEASE,REPACK,VEHD,IV
      CALL  REPACK  (      6,IV     )                                COLEASE
                     IF ( IREPFX . EQ . LFALSE )  GO TO 3040
C   COLEASE,REPACK,VEHF,IV
      CALL  REPACK  (      7,IV     )                                COLEASE
      GO TO 3040
 3030 CONTINUE
C-----ADD THE VEHICLES SIMULATION STATISTICS FOR THE INBOUND APPROACH
C-----AND TURN CODE AND LOG THE VEHICLE OUT OF THE SYSTEM, THE OUTBOUND
C-----APPROACH, AND THE OUTBOUND LANE
      CALL  LOGOUT
 3040 CONTINUE
CG                   IF ( IPRTLO . EQ . 0 )        GO TO 110
CF                   IF ( TIME . LT . TPRINT )     GO TO 110
CF     PRINT 704 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
CF110 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES OF THE NEXT VEHICLE TO
C-----BE PROCESSED FOR THIS LANE
      IV = NXVEH
C-----END OF VEHICLE LOOP
 4010 CONTINUE
C-----END OF OUTBOUND LANE LOOP
 5010 CONTINUE
C-----END OF OUTBOUND APPROACH LOOP
 6010 CONTINUE
      RETURN
      END                                                            OBAP
```

```
      SUBROUTINE SSOBAP                                                 COLEASE
C     TASK,SSOBAP
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / ZTEMPD / ZTEMPD(110)
      DATA     N1,N2  / 4HSSOB,2HAP /
C
C-----SUBROUTINE SSOBAP UPDATES THE VEHICLES SIMULATION STATISTICS ON
C-----THE OUTBOUND APPROACH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INCREMENT THE TRAVEL TIME
      ITIMV = ITIMV + 1
C-----IF THE VELOCITY IS LE XFPS THEN INCREMENT THE DELAY BELOW XX MPH
                      IF ( VELNEW . LE . XFPS )    IDVS = IDVS + 1
C-----ADD THE DESIRED SPEED FOR THIS DT FOR THE AVERAGE DESIRED SPEED
      ISPDS = ISPDS + ISPD
      RETURN
      END                                                               SSOBAP
```

```
      SUBROUTINE LOGOUT                                                 COLEASE
C     TASK,LOGOUT
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(18),NTD(18),QD(18),NQD(18),SD(18),MNVSY,
     *                  NSD(18),DMPH(18),NDMPH(18),VMT(18),
     *                  STIME(18),NUMPRO(18),ASPEED(18),ADESPD(18),
     *                  VMAXA(18),VMAXD(18),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / AMAXV,AVGSPD,AVGVEL,DESPD,DMAXV,INDEX,NVILL,
     *                  XDISTL,XDMPH,XQD,XSD,XSTIME,XTD,XVMT,ZTEMPD(96)
      DATA     N1,N2  / 4HLOGO,2HUT /
  601 FORMAT(12H ***** VEH =I4,7H IBAP =I3,7H TURN =I2,2X,10F9.4,
     *         6H *****)
C
C-----SUBROUTINE LOGOUT ADDS THE VEHICLES SIMULATION STATISTICS FOR THE
C-----INBOUND APPROACH AND TURN CODE AND LOGS THE VEHICLE OUT OF THE
C-----SYSTEM, THE OUTBOUND APPROACH, AND THE OUTBOUND LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      NUMPSU = NUMPSU + 1
C-----IF THE TIME INTO THE SIMULATION IS LE THE START-UP TIME FOR THE
C-----SIMULATION THEN DO NOT ADD THE VEHICLES SIMULATION STATISTICS
                      IF ( TIME . LE . STRTIM )    GO TO 1050
C-----COMPUTE THE INDEX FOR THE ARRAYS DIMENSIONED TO (6,3) IN /SUMSTA/
C-----BASED ON THE INBOUND APPROACH AND TURN CODE OF THE VEHICLE
      INDEX = (ITURN-1)*6 + IBAPS
      NUMPRO(INDEX) = NUMPRO(INDEX) + 1
C-----COMPLETE THE VEHICLES SIMULATION STATISTICS
      XDISTL = LGEOM(4) - POSOLD
      XSTIME = DT*(IEXTIM/25.0+ITIMV) + XDISTL/VELOLD
      AVGSPD = FLOAT(ISPDS)/FLOAT(ITIMV)
      IDTS = IDTS + XDISTL*25.0 + 0.5
C-----COMPUTE THE VEHICLES TOTAL DELAY = (THE ACTUAL TRAVEL TIME) -
```

```
C-----(THE TIME TO TRAVEL THE SAME DISTANCE AT THE AVERAGE DESIRED
C-----SPEED)
      XTD = XSTIME - 0.04*IDTS/AVGSPD
                    IF ( XTD . LE . 0.0 )          GO TO 1010
      TD(INDEX) = TD(INDEX) + XTD
      NTD(INDEX) = NTD(INDEX) + 1
 1010 CONTINUE
C-----COMPUTE THE VEHICLES QUEUE DELAY
      XQD = IQDS*DT
                    IF ( XQD . LE . 0.0 )          GO TO 1020
      QD(INDEX) = QD(INDEX) + XQD
      NQD(INDEX) = NQD(INDEX) + 1
 1020 CONTINUE
C-----COMPUTE THE VEHICLES STOPPED DELAY
      XSD = ISDS*DT
                    IF ( XSD . LE . 0.0 )          GO TO 1030
      SD(INDEX) = SD(INDEX) + XSD
      NSD(INDEX) = NSD(INDEX) + 1
 1030 CONTINUE
C-----COMPUTE THE VEHICLES DELAY BELOW XX MPH
      XDMPH = IDVS*DT
                    IF ( XDMPH . LE . 0.0 )        GO TO 1040
      DMPH(INDEX) = DMPH(INDEX) + XDMPH
      NDMPH(INDEX) = NDMPH(INDEX) + 1
 1040 CONTINUE
C-----COMPUTE THE VEHICLES MILES OF TRAVEL
      XVMT = IDTS/(5280.0*25.0)
      VMT(INDEX) = VMT(INDEX) + XVMT
C-----ADD THE VEHICLES TRAVEL TIME
      STIME(INDEX) = STIME(INDEX) + XSTIME
C-----COMPUTE THE VEHICLES AVERAGE VELOCITY
      AVGVEL = 3600.0*XVMT/XSTIME
      ASPEED(INDEX) = ASPEED(INDEX) + AVGVEL
C-----COMPUTE THE VEHICLES AVERAGE DESIRED SPEED
      DESPD = AVGSPD*60.0/88.0
      ADESPD(INDEX) = ADESPD(INDEX) + DESPD
C-----COMPUTE THE VEHICLES MAXIMUM ACC/DEC
      AMAXV = (IVMAXA/10.0)/AUTOL
      VMAXA(INDEX) = VMAXA(INDEX) + AMAXV
      DMAXV = (IVMAXD/10.0)/DUTOL
      VMAXD(INDEX) = VMAXD(INDEX) + DMAXV
                    IF ( IPRTLO . EQ . 0 )         GO TO 1050
C-----PRINT THE VEHICLES SIMULATION STATISTICS
      PRINT 601 , IQ(IV),LIBA(IBAPS),ITURN,XTD,XQD,XSD,XDMPH,XVMT,
     *            XSTIME,AVGVEL,DESPD,AMAXV,DMAXV
 1050 CONTINUE
C-----LOG THE VEHICLE OUT OF THE SYSTEM, THE OUTBOUND APPROACH, AND THE
C-----OUTBOUND LANE
      NVSY = NVSY - 1
      NVOBA = NVOBA - 1
      IQ(IV) = 0
      NVILL = NVIL(ILN) - 1
C   COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
      CALL  STORE   (NVILL ,      1,ISNA   ,      7+ILN    )              COLEASE
      NVIL(ILN) = NVILL
      NVIA(ISNA) = NVIA(ISNA) - 1
C-----SET THE FIRST VEHICLE IN THE LANE TO THIS VEHICLES NOR
C   COLEASE,STORE,NOR,LANE,IL,IFVL
      CALL  STORE   (NOR   ,      3,IL     ,      13)                     COLEASE
      IFVL = NOR
                    IF ( NOR . NE . 0 )           GO TO 1060
C-----IF THERE IS NO VEHICLE TO THE REAR THEN SET THE LAST VEHICLE IN
C-----THE LANE TO ZERO (NOR EQ 0)
C   COLEASE,STORE,0,LANE,IL,ILVL
      CALL  STORE   (0     ,      3,IL     ,      14)                     COLEASE
      ILVL = 0
      RETURN
 1060 CONTINUE
C-----SET MFINL AND MOASF TO LTRUE, RESET IACC TO SLIGHTLY DECELERATING
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE 0 FOR NOF FOR THE NOR VEHICLE
C-----(NOR NE 0)
      CALL  FLGNOR ( LTRUE,0 )
      RETURN
      END                                                              LOGOUT
```

```
      SUBROUTINE FLGNOR (LTF,    NEWNOF)                                COLEASE
C     TASK,FLGNOR,LTF,NEWNOF
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHF   / IDRICL     ,IVEHCL     ,ISPD       ,NOF      ,  COLEASE
     *                  NOR        ,LNEXT      ,LPRES      ,ITURN    ,  COLEASE
     *                  IBAPS      ,IPRTLO     ,IEXTIM     ,NOBAPD      COLEASE
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      DATA      N1,N2   / 4HFLGN,2HOR /
C
C-----SUBROUTINE FLGNOR SETS MFINL AND MOASF TO LTF, RESETS IACC TO
C-----SLIGHTLY DECELERATING IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT
C-----DECELERATING, SETS MSFLG TO LFALSE, AND FINALLY STORES NEWNOF FOR
C-----NOF FOR THE NOR VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                   IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET MFINL AND MOASF TO LTF FOR THE NOR VEHICLE
C     COLEASE,STORE,LTF,VEHD,NOR,MFINL
      CALL  STORE    (LTF     ,      6,NOR    ,      26)               COLEASE
C     COLEASE,STORE,LTF,VEHD,NOR,MOASF
      CALL  STORE    (LTF     ,      6,NOR    ,      29)               COLEASE
C-----RESET IACC TO SLIGHTLY DECELERATING IF MSFLG EQ LTRUE AND THE
C-----VEHICLE IS NOT DECELERATING FOR THE NOR VEHICLE
C     COLEASE,FIND,NSFLG,VEHD,NOR,MSFLG
      CALL  FIND     (NSFLG ,        6,NOR    ,      27)               COLEASE
C     COLEASE,FIND,JACC,VEHD,NOR,IACC
      CALL  FIND     (JACC    ,      6,NOR    ,       2)               COLEASE
      IF ( NSFLG.EQ.LTRUE.AND.JACC.GE.10000 )     JACC = 9999
C     COLEASE,STORE,JACC,VEHD,NOR,IACC
      CALL  STORE    (JACC    ,      6,NOR    ,       2)               COLEASE
C-----SET MSFLG TO LFALSE FOR THE NOR VEHICLE
C     COLEASE,STORE,LFALSE,VEHD,NOR,MSFLG
      CALL  STORE    (LFALSE,        6,NOR    ,      27)               COLEASE
C-----STORE NEWNOF FOR NOF FOR THE NOR VEHICLE
C     COLEASE,STORE,NEWNOF,VEHF,NOR,NOF
      CALL  STORE    (NEWNOF,        7,NOR    ,       4)               COLEASE
      RETURN
      END                                                               FLGNOR
```

```
      SUBROUTINE INTERP                                                 COLEASE
C     TASK,INTERP
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / PATH   / LENP       ,IOPT       ,LIBL       ,LOBL     ,  COLEASE
     *                  IFVP       ,ILVP       ,LIMP       ,IPT      ,  COLEASE
     *                  NGEOCP     ,NCPSET     ,ICPSET(60),LOBAP     ,  COLEASE
     *                  ILCH       ,IGEOCP(60)                          COLEASE
      COMMON / VEHD   / ISLP       ,IACC       ,IVEL       ,IPOS     ,  COLEASE
     *                  ISET       ,LCHGE      ,ISPDP      ,LEGAL    ,  COLEASE
     *                  IPRTM      ,ITIMV      ,IQDS       ,ISPDS    ,  COLEASE
     *                  ISDS       ,IDVS       ,ISTCON     ,IVMAXA   ,  COLEASE
     *                  IVMAXD     ,LATPOS     ,IDTS       ,LALT     ,  COLEASE
     *                  NORC       ,LOGFLG     ,MSTPF      ,MLAG     ,  COLEASE
     *                  MTCARS     ,MFINL      ,MSFLG      ,MPOBS    ,  COLEASE
     *                  MOASF      ,MSAOR      ,MPRO       ,MBLOCK   ,  COLEASE
     *                  MININT     ,IFVA       ,IACDS      ,ICDFS    ,  COLEASE
     *                  ISDEC      ,ISTMO      ,IACLDS     ,IRSTOP      COLEASE
      COMMON / VEHF   / IDRICL     ,IVEHCL     ,ISPD       ,NOF      ,  COLEASE
     *                  NOR        ,LNEXT      ,LPRES      ,ITURN    ,  COLEASE
     *                  IBAPS      ,IPRTLO     ,IEXTIM     ,NOBAPD      COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         IENT4(1),IENT6(1),IENT7(1)
      EQUIVALENCE       (LENP,IENT4(1)),(ISLP,IENT6(1)),
     *                  (IDRICL,IENT7(1))
C7    DATA      ITWO    / 2 /
CJ    DATA      IZERO   / 0 /
CK    DATA      IZERO   / 0 /
      DATA      N1,N2   / 4HINTE,2HRP /
CJ701 FORMAT(35H0SUMMARY FOR INTERSECTION PATHS AT ,F8.2,8H SECONDS,/,
CJ   *          51H PATH  VEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CJ   *          52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT SCON   SIG)
CK701 FORMAT(35H0SUMMARY FOR INTERSECTION PATHS AT ,F8.2,8H SECONDS,/,
CK   *          51H PATH  VEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CK   *          52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT SCON   SIG)
C7702 FORMAT(F7.2,5I4,2F7.1)
CJ703 FORMAT(I4,I6,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,2I5,2X,I4,
CJ   *          3(1X,A10))
CK703 FORMAT(I4,I6,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,2I5,2X,I4)
CO704 FORMAT(18(1X,A4,A2))
CN754 FORMAT(8H PATH    I3,1X,10I4,1X,60I1,2I3)
CN756 FORMAT(8H VEHD    I3,2(I5,I6),3I2,2I3,2I5,I7,2I5,I3,3I4,I6,I2,I4,
CN   *                  I3,2X,11I1,2X,7I1)
CN757 FORMAT(8H VEHF    I3,1X,12I4)
C
C-----SUBROUTINE INTERP PROCESSES THE VEHICLES ON THE INTERSECTION PATHS
C
      NRNAME = 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
CP    IHPRT = LFALSE
CP                   IF ( IHPRT . EQ . LFALSE )    GO TO 101
CL                   IF ( TIME . LT . TPRINT )     GO TO 101
```

```
CJ     PRINT 701 , TIME
CK     PRINT 701 , TIME
CL101 CONTINUE
      IGO = 0
C-----PROCESS EACH INTERSECTION PATH
      DO 5010  IP = 1 , NPATHS
      NV = NVIP(IP)
C-----IF THERE ARE NO VEHICLES ON THE INTERSECTION PATH THEN SKIP TO THE
C-----NEXT INTERSECTION PATH
                      IF ( NV . LE . 0 )            GO TO 5010
                      IF ( IPTHUP . EQ . IP )       GO TO 1010
C-----EXTRACT INTERSECTION PATH IP
C   COLEASE,EXTRAC,PATH,IP
      CALL  EXTRAC  (      4,IP     )                                  COLEASE
      IPTHUP = IP
 1010 CONTINUE
CP    IPPRT = LFALSE
CP                    IF ( IPPRT . EQ . LFALSE )    GO TO 102
CN                    IF ( TIME . LT . TPRINT )     GO TO 102
CN    NUM = NOATTB(4) - 60
CN    PRINT 754 , IP,(IENT4(I),I=1,NUM)
CN102 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES OF THE FIRST VEHICLE ON
C-----THE INTERSECTION PATH
      IV = IFVP
C-----PROCESS EACH VEHICLE ON THE INTERSECTION PATH
      DO 4010  IVN = 1 , NV
      NRNAME = 1
      ENDLN = LENP
C-----EXTRACT ENTRY IV OF ENTITY VEHF AND INITIALIZE SEVERAL PARAMETERS
C-----FOR THE VEHICLE
      CALL  PREST1  ( LTRUE )
      JFINL = MFINL
                      IF ( MFINL . EQ . LFALSE )    GO TO 1020
C-----LOOK AHEAD INTO THE LINKING OUTBOUND LANE FOR THE INTERSECTION
C-----PATH AND IF THERE IS A VEHICLE ON THE LANE THEN RESET THE PREVIOUS
C-----VEHICLE PARAMETERS TO THAT VEHICLE ELSE RESET THE PREVIOUS VEHICLE
C-----PARAMETERS TO THE END OF THE LANE
      CALL  LOKIOB
 1020 CONTINUE
C-----COMPUTE NEW ACC/DEC LOGIC AND EXTRACT ENTRY IV OF ENTITY VEHD FOR
C-----THE VEHICLE
      CALL  PREST2
      MFINL = JFINL
CP                    IF ( IPRTLO . EQ . 0 )        GO TO 105
CN                    IF ( TIME . LT . TPRINT )     GO TO 105
CP                    IF ( IHPRT . EQ . LTRUE )     GO TO 103
CP    PRINT 701 , TIME
CP    IHPRT = LTRUE
CP103 CONTINUE
CP                    IF ( IPPRT . EQ . LTRUE )     GO TO 104
CP    NUM = NOATTB(4) - 60
CP    PRINT 754 , IP,(IENT4(I),I=1,NUM)
CP    IPPRT = LTRUE
CP104 CONTINUE
CN    NUM = NOATTB(7)
CN    PRINT 757 , IV,(IENT7(I),I=1,NUM)
CN    NUM = NOATTB(6)
CN    PRINT 756 , IV,(IENT6(I),I=1,NUM)
CN105 CONTINUE
C-----UNBIAS THE VEHICLE ATTRIBUTES AND PREDICT THE NEW POS/VEL/ACC
      CALL  UNBIAS
      NXVEH = NOR
C-----CHECK THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES, CALL THE
C-----APPROPRIATE ACC/DEC ROUTINES, AND COMPUTE THE VEHICLES NEW POS/
C-----VEL/ACC
      CALL  ACDCP
C7    POSLAT = 0.0
C7    WRITE (4,702)  TIME,IQ(IV),ITWO,IP,IP,IVEHCL,POSNEW,POSLAT
CR                    IF ( IPRTLO . EQ . 0 )        GO TO 106
C-----PRINT POS/VEL/ACC FOR THE VEHICLE
CQ    CALL  PVAPRT
CR106 CONTINUE
C-----UPDATE THE VEHICLES SIMULATION STATISTICS IN THE INTERSECTION
      CALL  SSINTR
                      IF ( ISTCON . GT . NGEOCP )   GO TO 2010
C-----CLEAR THE INTERSECTION CONFLICTS AS THE REAR BUMPER PASSES THEM
      CALL  CLRCON
 2010 CONTINUE
C-----IF THE VEHICLE LEFT THE INTERSECTION PATH THEN GO TO 2020 AND LOG
C-----THE VEHICLE OUT OF THE INTERSECTION PATH AND INTO THE LINKING
C-----OUTBOUND LANE
                      IF ( POSNEW.GE.FLOAT(LENP) ) GO TO 2020
                      IF ( PVPOS+4.0.GT.POSNEW )    GO TO 3010
C-----PRINT THE COLLISION INFORMATION AND RESET THE VEHICLES POS/VEL/ACC
      CALL  BANGS   ( 2 )
      GO TO 3010
 2020 CONTINUE
C-----LOG THE VEHICLE OUT OF THE INTERSECTION PATH AND INTO THE LINKING
C-----OUTBOUND LANE
      CALL  LOGIOB
C3    KPFLAG = 10HLEAVE INTR
 3010 CONTINUE
C-----BIAS THE VEHICLE ATTRIBUTES, SET THE PREVIOUS VEHICLE PARAMETERS,
C-----AND UPDATE THE MAXIMUM ACC/DEC FOR THE VEHICLE
      CALL  BIAS
C-----PRINT SELECTED ATTRIBUTES FOR THE VEHICLE
                      IF ( JPRTM . NE . 0 )        IPRTM = JPRTM
CM                    IF ( IPRTLO . EQ . 0 )       GO TO 107
CL                    IF ( TIME . LT . TPRINT )    GO TO 107
C3                    IF ( JPRTM . GT . 0 )        JPFLAG = 10HPIJR TIME
CJ    IDESPD = DESVEL + 0.5
CJ    PRINT 703 , IP,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,SLPNEW,
CJ   *            IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,IZERO,
CJ   *            LCHGE,IPRTM,ISTCON,IZERO,IPFLAG,JPFLAG,KPFLAG
CK    IDESPD = DESVEL + 0.5
CK    PRINT 703 , IP,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,SLPNEW,
CK   *            IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,IZERO,
CK   *            LCHGE,IPRTM,ISTCON,IZERO
CL107 CONTINUE
C-----REPACK THE ATTRIBUTES FOR VEHICLE IV
C   COLEASE,REPACK,VEHD,IV
      CALL  REPACK  (      6,IV     )                                  COLEASE
                      IF ( IREPFX . EQ . LFALSE )  GO TO 3020
C   COLEASE,REPACK,VEHF,IV
      CALL  REPACK  (      7,IV     )                                  COLEASE
 3020 CONTINUE
CP                    IF ( IPRTLO . EQ . 0 )       GO TO 108
CO                    IF ( TIME . LT . TPRINT )    GO TO 108
CO    PRINT 704 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
CO108 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES FOR THE NEXT VEHICLE ON
C-----THE INTERSECTION PATH TO BE PROCESSED
      IV = NXVEH
C-----END OF VEHICLE LOOP
 4010 CONTINUE
C-----END OF INTERSECTION PATH LOOP
 5010 CONTINUE
      RETURN
      END                                                              INTERP
```

```
      SUBROUTINE LOKIOB                                                 COLEASE
C     TASK,LOKIOB
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / JACC,JPOS,JVEHCL,JVEL,LGEOM1,ZTEMPD(105)
      DATA      N1,N2   / 4HLOKI,2HOB /
C
C-----SUBROUTINE LOKIOB LOOKS AHEAD INTO THE LINKING OUTBOUND LANE FOR
C-----THE INTERSECTION PATH AND IF THERE IS A VEHICLE ON THE LANE THEN
C-----RESETS THE PREVIOUS VEHICLE PARAMETERS TO THAT VEHICLE ELSE RESETS
C-----THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----FIND THE ENTRY NUMBER FOR THE LAST VEHICLE ON THE LINKING OUTBOUND
C-----LANE FOR THE INTERSECTION PATH
C     COLEASE,FIND,IVPV,LANE,LNEXT,ILVL
      CALL  FIND     (IVPV   ,       3,LNEXT  ,      14)                COLEASE
                      IF ( IVPV . LE . 0 )           GO TO 1010
C-----RESET THE PREVIOUS VEHICLE PARAMETERS TO THE LAST VEHICLE ON THE
C-----LINKING OUTBOUND LANE FOR THE INTERSECTION PATH
C     COLEASE,STORE,LFALSE,VEHD,IV,MFINL
      CALL  STORE    (LFALSE,        6,IV     ,      26)                COLEASE
      MFINL = LFALSE
C     COLEASE,FIND,LGEOM1,LANE,LNEXT,LGEOM(1)
      CALL  FIND     (LGEOM1,        3,LNEXT  ,      17)                COLEASE
C     COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
      CALL  FIND     (JVEHCL,        7,IVPV   ,       2)                COLEASE
C     COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
      CALL  FIND     (JPOS   ,       6,IVPV   ,       4)                COLEASE
      PVPOS = LENP + JPOS/25.0 - LGEOM1 - LENV(JVEHCL) - 4.0
C     COLEASE,FIND,JVEL,VEHD,IVPV,IVEL
      CALL  FIND     (JVEL   ,       6,IVPV   ,       3)                COLEASE
      PVVEL = JVEL/25.0
C     COLEASE,FIND,JACC,VEHD,IVPV,IACC
      CALL  FIND     (JACC   ,       6,IVPV   ,       2)                COLEASE
      PVACC = JACC/312.5 - 32.0
      RETURN
 1010 CONTINUE
C-----RESET THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE LANE
      PVPOS = 5000.0
      PVVEL = 0.0
      PVACC = 0.0
      RETURN
      END                                                               LOKIOB
```

```
      SUBROUTINE SSINTR                                                 COLEASE
C     TASK,SSINTR
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / ZTEMPD / JAN,JL,JLN,JSNA,MLANES,ZTEMPD(105)
      DIMENSION         MSG901(10)
      DATA      MSG901 / 4H LIB,4HL NO,4HT ON,4H LLA,4HNES ,4HFOR ,
     *                   4HJSNA,4H = S,4HSINT,4HR    /
      DATA      N1,N2   / 4HSSIN,2HTR /
C
C-----SUBROUTINE SSINTR UPDATES THE VEHICLES SIMULATION STATISTICS IN
C-----THE INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INCREMENT THE TRAVEL TIME
      ITIMV = ITIMV + 1
C-----IF THE VELOCITY IS LE XFPS THEN INCREMENT THE DELAY BELOW XX MPH
                      IF ( VELNEW . LE . XFPS )     IDVS = IDVS + 1
C-----ADD THE DESIRED SPEED FOR THIS DT FOR THE AVERAGE DESIRED SPEED
      ISPDS = ISPDS + ISPD
                      IF ( POSNEW . GT . 5.0 )      RETURN
                      IF ( VELNEW . GT . 3.0 )      RETURN
C-----THE VEHICLE IS STILL STOPPED AT THE START OF THE INTERSECTION PATH
C-----THUS INCREMENT QUEUE DELAY AND STOPPED DELAY FOR THE VEHICLE AND
C-----INCREMENT THE QUEUE LENGTH FOR THE VEHICLES INBOUND APPROACH AND
C-----LANE
      IQDS = IQDS + 1
      ISDS = ISDS + 1
      JSNA = LIBA(IBAPS)
C     COLEASE,FIND,MLANES,APPRO,JSNA,NLANES
      CALL  FIND     (MLANES,        1,JSNA   ,       1)                COLEASE
      DO 1010  JLN = 1 , MLANES
C     COLEASE,FIND,JL,APPRO,JSNA,LLANES(JLN)
      CALL  FIND     (JL     ,       1,JSNA   ,     1+JLN   )           COLEASE
                      IF ( JL . EQ . LIBL )         GO TO 1020
 1010 CONTINUE
      GO TO 9010
 1020 CONTINUE
      LQUEUE(IBAPS,JLN) = LQUEUE(IBAPS,JLN) + 1
```

```
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9010 CONTINUE
      CALL  ABORTR  ( MSG901,37 )
      STOP  901
      END                                                               SSINTR
```

```
      SUBROUTINE CLRCON                                                 COLEASE
C   TASK,CLRCON
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / I,IK,IPOSRB,J,JCONI,JGEOCP,JP,MCPSET,NUM,
     *                  ZTEMPD(101)
      DIMENSION         IENT2(1)
      EQUIVALENCE       (ICONP(1),IENT2(1))
      DATA      N1,N2  / 4HCLRC,2HON /
CN752 FORMAT(8H CONFLT I3,1X,12I4)
C
C-----SUBROUTINE CLRCON CLEARS THE INTERSECTION CONFLICTS AS THE REAR
C-----BUMPER PASSES THEM
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      IPOSRB = POSNEW - LENV(IVEHCL) + 0.5
CN    NUM = NOATTB(2)
C-----CHECK THE INTERSECTION CONFLICTS THAT THE VEHICLE HAS NOT CLEARED
      DO 1030  IK = ISTCON , NGEOCP
      ISTCON = IK
      JGEOCP = IGEOCP(IK)
                    IF ( ICONUP . EQ . JGEOCP )  GO TO 1010
C   COLEASE,EXTRAC,CONFLT,JGEOCP
      CALL  EXTRAC  (      2,JGEOCP)                                    COLEASE
      ICONUP = JGEOCP
 1010 CONTINUE
CP                  IF ( IPRTLO . EQ . 0 )       GO TO 101
CN                  IF ( TIME . LT . TPRINT )    GO TO 101
CN    PRINT 752 , JGEOCP,(IENT2(I),I=1,NUM)
CN101 CONTINUE
      J = 1
                    IF ( IP . EQ . ICONP(2) )     J = 2
C-----IF THE VEHICLE IS TO LEAVE THE INTERSECTION PATH THIS DT THEN
C-----CLEAR ALL REMAINING INTERSECTION CONFLICTS
                    IF ( POSNEW.GE.FLOAT(LENP) ) GO TO 1020
C-----IF THE POSITION OF THE REAR BUMPER IS LT THE DISTANCE TO THE
C-----INTERSECTION CONFLICT THEN DO NOT CLEAR THE INTERSECTION CONFLICT
                    IF ( IPOSRB.LT.ICOND(J) )    RETURN
 1020 CONTINUE
```

```
C-----SET THE VEHICLES NORC AS THE NEXT VEHICLE THAT HAS NOT CLEARED THE
C-----INTERSECTION CONFLICT
C   COLEASE,STORE,NORC,CONFLT,JGEOCP,ICONV(J)
      CALL  STORE   (NORC   ,      2,JGEOCP,      9+J      )            COLEASE
      ICONV(J) = NORC
                    IF ( NORC . NE . 0 )          GO TO 1050
C-----UNSET THE INTERSECTION CONFLICT FOR THE OTHER INTERSECTION PATH
      J = 3 - J
      JP = ICONP(J)
C   COLEASE,FIND,MCPSET,PATH,JP,NCPSET
      CALL  FIND    (MCPSET,      4,JP     ,      10)                 COLEASE
      MCPSET = MAX0(MCPSET-1,0)
C   COLEASE,STORE,MCPSET,PATH,JP,NCPSET
      CALL  STORE   (MCPSET,      4,JP     ,      10)                 COLEASE
                    IF ( IPTHUP . EQ . JP )      NCPSET = MCPSET
      JCONI = ICONI(J)
C   COLEASE,STORE,0,PATH,JP,ICPSET(JCONI)
      CALL  STORE   (0      ,      4,JP     ,     10+JCONI )          COLEASE
                    IF ( IPTHUP . EQ . JP )      ICPSET(JCONI) = 0
C-----END OF INTERSECTION CONFLICT LOOP
 1030 CONTINUE
C-----ALL THE INTERSECTION CONFLICTS HAVE BEEN PASSED BY THE VEHICLE
      ISTCON = NGEOCP + 1
      RETURN
      END                                                             CLRCON
```

```
      SUBROUTINE LOGIOB                                               COLEASE
C   TASK,LOGIOB
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,   COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)   COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA     ,   COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL     ,   COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLOL     ,   COLEASE
     *                  LLOL  ( 5),IBLN      ,IDUMLA                  COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                  COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL     ,   COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT      ,   COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP    ,   COLEASE
     *                  ILCH      ,IGEOCP(60)                         COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,   COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL    ,   COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,   COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,   COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,   COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,   COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,   COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,   COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,   COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP       COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF      ,   COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN    ,   COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD       COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
C6    COMMON / PRTPVA / DISTAD(200)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / JPOS,JVEL,NVILL,ZTEMPD(107)
      DIMENSION         MSG902(10)
      DATA      MSG902 / 4H LNE,4HXT I,4HS NO,4HT ON,4H LLA,4HNES ,
     *                   4HLIST,4H - L,4HOGIO,4HB     /
      DATA      N1,N2  / 4HLOGI,2HOB /
C
C-----SUBROUTINE LOGIOB LOGS THE VEHICLE OUT OF THE INTERSECTION PATH
C-----AND INTO THE LINKING OUTBOUND APPROACH AND LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----EXTRACT LINKING OUTBOUND LANE LNEXT
C   COLEASE,EXTRAC,LANE,LNEXT
      CALL  EXTRAC  (      3,LNEXT )                              COLEASE
C-----EXTRACT LINKING OUTBOUND APPROACH ISNA
C   COLEASE,EXTRAC,APPRO,ISNA
      CALL  EXTRAC  (      1,ISNA  )                              COLEASE
C-----SET NOF TO THE LAST VEHICLE IN THE LINKING OUTBOUND LANE
      NOF = ILVL
      POSNEW = POSNEW - LENP + LGEOM(1)
C-----INCREMENT THE NUMBER OF VEHICLES ON THE LINKING OUTBOUND APPROACH
C-----AND LANE
      NVOBA = NVOBA + 1
      NVIA(ISNA) = NVIA(ISNA) + 1
      DO 1010  ILN = 1 , NLANES
                    IF ( LNEXT.EQ.LLANES(ILN) )  GO TO 1020
 1010 CONTINUE
      GO TO 9020
 1020 CONTINUE
      NVILL = NVIL(ILN) + 1
C   COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
      CALL  STORE   (NVILL ,      1,ISNA  ,      7+ILN   )            COLEASE
```

```
      NVIL(ILN) = NVILL
C-----DECREMENT THE NUMBER OF VEHICLES ON THE INTERSECTION PATH
      NVIP(LPRES) = NVIP(LPRES) - 1
      NVIN = NVIN - 1
C-----SET THE FIRST VEHICLE ON THE INTERSECTION PATH TO THIS VEHICLES
C-----NOR
C   COLEASE,STORE,NOR,PATH,LPRES,IFVP
      CALL  STORE   (NOR    ,       4,LPRES ,      5)                   COLEASE
                     IF ( IPTHUP . EQ . LPRES )   IFVP = NOR
                     IF ( NOR . NE . 0 )            GO TO 1030
C-----SET THE LAST VEHICLE ON THE INTERSECTION PATH TO 0 (OLD NOR EQ 0)
C   COLEASE,STORE,0,PATH,LPRES,ILVP
      CALL  STORE   (0      ,       4,LPRES ,      6)                   COLEASE
                     IF ( IPTHUP . EQ . LPRES )   ILVP = 0
      GO TO 2010
 1030 CONTINUE
C-----SET MFINL AND MOASF TO LTRUE, RESET IACC TO SLIGHTLY DECELERATING
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE 0 FOR NOF FOR THE NOR VEHICLE (OLD
C-----NOR NE 0)
      CALL  FLGNOR ( LTRUE,0 )
 2010 CONTINUE
                     IF ( NOF . EQ . 0 )            GO TO 2020
C-----CHECK WHICH VEHICLE ON THE OUTBOUND LANE THAT THIS VEHICLE SHOULD
C-----BE BEHIND (NEW NOF NE 0)
C   COLEASE,FIND,JPOS,VEHD,NOF,IPOS
      CALL  FIND    (JPOS   ,       6,NOF   ,      4)                   COLEASE
C-----IF THE POSITION OF THIS VEHICLE IS LE THE POSITION OF THE NOF
C-----VEHICLE THEN GO TO 2030 AND PUT THIS VEHICLE BEHIND THE NOF
C-----VEHICLE
            IF ( POSNEW . LE . JPOS/25.0 )          GO TO 2030
C-----SET THE VEHICLE AHEAD OF THE NOF VEHICLE AS THE NEW NOF VEHICLE
C   COLEASE,FIND,NOF,VEHF,NOF,NOF
      CALL  FIND    (NOF    ,       7,NOF   ,      4)                   COLEASE
C-----IF THERE WAS A VEHICLE AHEAD OF THE NOF VEHICLE THEN GO TO 2010
C-----AND CHECK THE POSITION ELSE SET THIS VEHICLE AS THE NEW FIRST
C-----VEHICLE ON THE LINKING OUTBOUND LANE
                     IF ( NOF . NE . 0 )            GO TO 2010
 2020 CONTINUE
C-----SET THIS VEHICLE AS THE NEW FIRST VEHICLE ON THE LINKING OUTBOUND
C-----LANE (NEW NOF EQ 0)
      NOR = IFVL
C   COLEASE,STORE,IV,LANE,LNEXT,IFVL
      CALL  STORE   (IV     ,       3,LNEXT ,     13)                   COLEASE
      IFVL = IV
      MFINL = LTRUE
      MOASF = LTRUE
                     IF ( NOR . NE . 0 )            GO TO 2050
      GO TO 2040
 2030 CONTINUE
C-----SET THIS VEHICLE BEHIND THE NOF VEHICLE ON THE LINKING OUTBOUND
C-----APPROACH (NEW NOF NE 0)
      MFINL = LFALSE
      MOASF = LFALSE
C   COLEASE,FIND,JVEL,VEHD,NOF,IVEL
      CALL  FIND    (JVEL   ,       6,NOF   ,      3)                   COLEASE
                     IF ( JVEL . LE . 0 )           MOASF = LTRUE
C   COLEASE,FIND,NOR,VEHF,NOF,NOR
      CALL  FIND    (NOR    ,       7,NOF   ,      5)                   COLEASE
C   COLEASE,STORE,IV,VEHF,NOF,NOR
      CALL  STORE   (IV     ,       7,NOF   ,      5)                   COLEASE
                     IF ( NOR . NE . 0 )            GO TO 2050
 2040 CONTINUE
C-----SET THE LAST VEHICLE ON THE LINKING OUTBOUND LANE TO THIS VEHICLE
C-----(NEW NOR EQ 0)
C   COLEASE,STORE,IV,LANE,LNEXT,ILVL
      CALL  STORE   (IV     ,       3,LNEXT ,     14)                   COLEASE
      ILVL = IV
      GO TO 3010
 2050 CONTINUE
C-----SET MFINL AND MOASF TO LFALSE, RESET IACC TO SLIGHTLY DECELERATING
```

```
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE IV FOR NOF FOR THE NOR VEHICLE (NEW
C-----NOR NE 0)
      CALL  FLGNOR ( LFALSE,IV )
 3010 CONTINUE
      LEGAL = 2
C-----CHECK MY LANE AND IF BLOCKED THEN SET PARAMETERS FOR BLOCKED LANE
      CALL  CHKMLN
      LALT = 1
                     IF ( NLR . NE . 0 )          LALT = LALT + 1
                     IF ( NLL . NE . 0 )          LALT = LALT + 2
                     IF ( LEGAL . EQ . 2 )        ISET = 6
C-----RESET SOME OF THE VEHICLE ATTRIBUTES
      LPRES = LNEXT
      LNEXT = 0
                     IF ( LATPOS . EQ . LTRUE )   ISPD = ISPD/2
      ISPD = FLOAT(ISPD)*FLOAT(ISLIM)/FLOAT(LIMP) + 0.5
      MSFLG = LFALSE
      MININT = LFALSE
      LATPOS = 0
      IREPFX = LTRUE
C6    DISTAD(IV) = DISTAD(IV) + LENP
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9020 CONTINUE
      CALL  ABORTR  ( MSG902,37 )
      STOP  902
      END                                                               LOGIOB
```

```
      SUBROUTINE IBAP                                                   COLEASE
C   TASK,IBAP
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL   ( 6),ISLIM     ,   COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         IENT1(1),IENT3(1),IENT6(1),IENT7(1),IENT8(1)
      EQUIVALENCE       (NLANES,IENT1(1)),(LWID,IENT3(1)),
     *                  (ISLP,IENT6(1)),(IDRICL,IENT7(1)),
     *                  (MDEDIC,IENT8(1))
      LOGICAL           INQUE
C7    DATA      IONE    / 1 /
      DATA      N1,N2   / 4HIBAP,2H    /
CS701 FORMAT(35H0SUMMARY FOR INBOUND APPROACHES AT ,F8.2,8H SECONDS,/,
CS   *          51H AP LN VEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CS   *          52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT LPOS Q SIG)
CT701 FORMAT(35H0SUMMARY FOR INBOUND APPROACHES AT ,F8.2,8H SECONDS,/,
CT   *          51H AP LN VEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
CT   *          52HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT LPOS Q SIG)
C7702 FORMAT(F7.2,5I4,2F7.1)
CS703 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,1X,L1,I4,
CS   *          3(1X,A10))
CT703 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,1X,L1,I4)
CX704 FORMAT(18(1X,A4,A2))
CW751 FORMAT(8H APPRO  I3,1X,26I4)
CW753 FORMAT(8H LANE   I3,1X,28I4)
CW756 FORMAT(8H VEHD   I3,2(I5,I6),3I2,2I3,2I5,I7,2I5,I3,3I4,I6,I2,I4,
CW   *                 I3,2X,11I1,2X,7I1)
CW757 FORMAT(8H VEHF   I3,1X,12I4)
CW758 FORMAT(8H VEHIL  I3,1X,12I2,1X,8I2)
```

```
C
C-----SUBROUTINE IBAP PROCESSES THE VEHICLES ON THE INBOUND APPROACHES
C-----AND LOGS NEW VEHICLES INTO THE SYSTEM FROM THE QUEUE BUFFERS AS
C-----REQUIRED
C
      NRNAME = 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
CY    IHPRT = LFALSE
CY                      IF ( IHPRT . EQ . LFALSE )    GO TO 101
CU                      IF ( TIME . LT . TPRINT )     GO TO 101
CS    PRINT 701 , TIME
CT    PRINT 701 , TIME
CU101 CONTINUE
C-----PROCESS EACH INBOUND APPROACH
      DO 6010  IAN = 1 , NIBA
      IA = LIBA(IAN)
C-----FIND THE NUMBER OF VEHICLES TO BE LOGGED INTO THE SYSTEM THIS DT
C-----FOR THIS INBOUND APPROACH
      NQA = 0
      DO 1010  ILN = 1 , 6
      NQA = NQA + LQ(IAN,ILN)
 1010 CONTINUE
C-----IF THERE ARE NO VEHICLES ON THE APPROACH AND NO VEHICLES TO BE
C-----LOGGED INTO THE APPROACH THEN SKIP TO THE NEXT INBOUND APPROACH
                        IF ( NVIA(IA)+NQA.LE.0 )      GO TO 6010
C-----EXTRACT INBOUND APPROACH IA
C   COLEASE,EXTRAC,APPRO,IA
      CALL  EXTRAC  (     1,IA     )                                COLEASE
CY    IAPRT = LFALSE
CY                      IF ( IAPRT . EQ . LFALSE )   GO TO 102
CW                      IF ( TIME . LT . TPRINT )    GO TO 102
CW    NUM = NOATTB(1)
CW    PRINT 751 , IA,(IENT1(I),I=1,NUM)
CW102 CONTINUE
C-----PROCESS EACH LANE OF THE INBOUND APPROACH
      DO 5010  ILN = 1 , NLANES
C-----IF THERE ARE NO VEHICLES IN THIS LANE OR NO VEHICLES TO BE LOGGED
C-----INTO THIS LANE THIS DT THEN SKIP TO THE NEXT LANE
              IF ( NVIL(ILN)+LQ(IAN,ILN) . LE . 0 ) GO TO 5010
      IL = LLANES(ILN)
      LCHGE = 1
C-----EXTRACT LANE IL
C   COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC  (     3,IL     )                                COLEASE
CY    ILPRT = LFALSE
CY                      IF ( ILPRT . EQ . LFALSE )   GO TO 103
CW                      IF ( TIME . LT . TPRINT )    GO TO 103
CW    NUM = NOATTB(3)
CW    PRINT 753 , ILN,(IENT3(I),I=1,NUM)
CW103 CONTINUE
C-----IF THERE ARE NO VEHICLES IN THIS LANE THEN LOG IN THE NEW VEHICLE
                        IF ( NVIL(ILN) . LE . 0 )    GO TO 4020
      IGO = 1
      JSISET = 0
C-----IF THIS LANE IS NOT SIGNAL CONTROLLED OR THE SIGNAL INDICATION FOR
C-----THIS LANE HAS NOT CHANGED FROM THE OLD CAM STACK POSITION
C-----INDICATION THEN GO TO 1020 ELSE SET THE SIGNAL INDICATION FOR THE
C-----CURRENT CAM STACK POSITION AND INBOUND LANE NUMBER
                        IF ( LCONTR . LT . 5 )       GO TO 1020
      IF ( ISISET(ICAMPC,IBLN) . EQ . ISISET(ICAMPO,IBLN) )
     *                                               GO TO 1020
      JSISET = ISISET(ICAMPC,IBLN)
 1020 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES OF THE FIRST VEHICLE IN
C-----THIS LANE
      IV = IFVL
      NV = NVIL(ILN)
      INQUE = .TRUE.
C-----PROCESS EACH VEHICLE ON THIS LANE
      DO 4010  IVN = 1 , NV
```

```
      NPNAME = 1
      IREPIL = LFALSE
      ENDLN = LGEOM(4) + 1.5
C-----EXTRACT ENTRY IV OF ENTITY VEHF, RESET THE PREVIOUS VEHICLE
C-----PARAMETERS TO THE NEW NOF IF THE VEHICLE IS LANE CHANGING, AND
C-----INITIALIZE SEVERAL PARAMETERS FOR THE VEHICLE
      CALL  PRESTI  ( LFALSE )
      POSCHK = PVPOS
      JFINL = MFINL
                      IF ( MFINL . EQ . LFALSE )    GO TO 1040
                      IF ( IVN . EQ . 1 )           GO TO 1030
                      IF ( PVVEL . GT . 0.1 )       GO TO 1030
      MFINL = LFALSE
      PVPOS = PVPOS + LGEOM(4)
      GO TO 1035
 1030 CONTINUE
C-----THIS VEHICLE IS THE FIRST VEHICLE IN THE LANE THUS RESET THE
C-----PREVIOUS VEHICLE PARAMETERS FOR THE END OF THE LANE
      PVPOS = ENDLN
      PVVEL = 0.0
      PVACC = 0.0
 1035 CONTINUE
      POSCHK = PVPOS
                      IF ( LNEXT . EQ . 0 )         GO TO 1040
C   COLEASE,FIND,MPRO,VEHD,IV,MPRO
      CALL  FIND    (MPRO   ,      6,IV     ,      31)                 COLEASE
                      IF ( MPRO . EQ . LFALSE )     GO TO 1040
C-----THIS VEHICLE MAY PROCEED INTO THE INTERSECTION THUS LOOK AHEAD
C-----INTO THE LINKING INTERSECTION PATH FOR THIS VEHICLE AND IF THERE
C-----IS A VEHICLE ON THE INTERSECTION PATH THEN RESET THE PREVIOUS
C-----VEHICLE PARAMETERS TO THE LAST VEHICLE ON THE INTERSECTION PATH
C-----ELSE RESET THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE
C-----INTERSECTION PATH
      CALL  LOKIBI
 1040 CONTINUE
C-----COMPUTE NEW ACC/DEC LOGIC AND EXTRACT ENTRY IV OF ENTITY VEHD FOR
C-----THE VEHICLE
      CALL  PREST2
      MFINL = JFINL
                      IF ( LOGFLG . GT . 1 )        LOGFLG = LOGFLG - 1
      LOGTMP = LOGFLG
CY                    IF ( IPRTLO . EQ . 0 )        GO TO 107
CW                    IF ( TIME . LT . TPRINT )     GO TO 107
CY                    IF ( IHPRT . EQ . LTRUE )     GO TO 104
CY    PRINT 701 , TIME
CY    IHPRT = LTRUE
CY104 CONTINUE
CY                    IF ( IAPRT . EQ . LTRUE )     GO TO 105
CY    NUM = NOATTB(1)
CY    PRINT 751 , IA,(IENT1(I),I=1,NUM)
CY    IAPRT = LTRUE
CY105 CONTINUE
CY                    IF ( ILPRT . EQ . LTRUE )     GO TO 106
CY    NUM = NOATTB(3)
CY    PRINT 753 , ILN,(IENT3(I),I=1,NUM)
CY    ILPRT = LTRUE
CY106 CONTINUE
CW    NUM = NOATTB(7)
CW    PRINT 757 , IV,(IENT7(I),I=1,NUM)
CW    NUM = NOATTB(6)
CW    PRINT 756 , IV,(IENT6(I),I=1,NUM)
CW107 CONTINUE
                      IF ( LALT . NE . 6 )          GO TO 1050
C-----THIS VEHICLE HAS ALREADY BEEN PROCESSED IN THIS DT THUS RESET THE
C-----PREVIOUS VEHICLE PARAMETERS AND SKIP TO THE NEXT VEHICLE
      LALT = 5
      PVPOS = IPOS/25.0 - LENV(IVEHCL) - 4.0
      PVVEL = IVEL/25.0
      PVACC = IACC/312.5 - 32.0
      NXVEH = NOR
      GO TO 3020
 1050 CONTINUE
                      IF ( LOGFLG . NE . 1 )        GO TO 1060
C-----COMPUTE NEW INTERSECTION CONTROL LOGIC
C   COLEASE,LOGIC,VEHIL,IV
      CALL  LOGIC   (      8,IV     )                                  COLEASE
      LOGTMP = 2
 1060 CONTINUE
C-----EXTRACT ENTRY IV OF ENTITY VEHIL
C   COLEASE,EXTRAC,VEHIL,IV
      CALL  EXTRAC  (      8,IV     )                                  COLEASE
CY                    IF ( IPRTLO . EQ . 0 )        GO TO 108
CW                    IF ( TIME . LT . TPRINT )     GO TO 108
CW    NUM = NOATTB(8)
CW    PRINT 758 , IV,(IENT8(I),I=1,NUM)
CW108 CONTINUE
C-----UNBIAS THE VEHICLE ATTRIBUTES AND PREDICT THE NEW POS/VEL/ACC
      CALL  UNBIAS
      NXVEH = NOR
                      IF ( ISPDP . NE . 0 )         GO TO 1080
                      IF ( MBLOCK . EQ . LTRUE )    GO TO 1080
                      IF ( LNEXT . EQ . 0 )         GO TO 1080
                      IF ( RELEND . LE . 25.0 )     GO TO 1070
                      IF ( VELOLD . LE . 0.0 )      GO TO 1080
 1070 CONTINUE
C-----CHECK TO SEE IF THE VEHICLE SHOULD RESET HIS DESIRED SPEED TO THE
C-----DESIRED SPEED OF HIS INTERSECTION PATH SO THAT HE CAN GRADUALLY
C-----DECELERATE TO HIS NEW DESIRED SPEED BEFORE HE ENTERS THE
C-----INTERSECTION
      CALL  CHKDSP
 1080 CONTINUE
      KSISET = JSISET
      TESTLP = 1.0
 1090 CONTINUE
      JGO = 1
                      IF ( IGO . EQ . 2 )           JGO = 3
                      IF ( KSISET . EQ . 0 )        GO TO 1100
                      IF ( MDEDIC . EQ . LFALSE )   GO TO 1100
                      IF ( MINFLZ . EQ . LFALSE )   GO TO 1100
C-----THE SIGNAL INDICATION HAS CHANGED FOR THIS LANE, THE VEHICLE IS
C-----DEDICATED TO AN INTERSECTION PATH, AND THE VEHICLE IS WITHIN THE
C-----INFLUENCE ZONE OF THE INTERSECTION CONTROL THUS DETERMINE THE
C-----APPROPRIATE DRIVER RESPONSE FOR THE NEW SIGNAL INDICATION
      CALL  SIGRES  ( KSISET )
      JGO = IGO
 1100 CONTINUE
                      IF ( TESTLP . LE . 0.1 )      GO TO 2010
      IGO = JGO
                      IF ( ISET . NE . 1 )          GO TO 2020
C-----THIS VEHICLE IS CHANGING LANES THUS CHECK IF THE SIGNAL RESPONSE
C-----ROUTINE SHOULD BE CALLED
                      IF ( LCONTR . LT . 5 )        GO TO 2010
      TESTLP = ABS(ABS(LATPOS/8.0-15.0)-LEGAL/2.0)
                      IF ( TESTLP . GT . 0.1 )      GO TO 2010
C-----THIS IS THE FIRST DT THAT THE VEHICLE IS BEING PROCESSED IN HIS
C-----NEW LANE AFTER THE LANE CHANGE WAS STARTED AND THE LANE IS SIGNAL
C-----CONTROLLED THUS CALL THE SIGNAL RESPONSE ROUTINE
      KSISET = ISISET(ICAMPC,IBLN)
      GO TO 1090
 2010 CONTINUE
C-----COMPUTE THE NEW LATERAL POSITION FOR A LANE CHANGE USING A COSINE
C-----CURVE AND IF FINISHED THEN END THE LANE CHANGE
      CALL  LCHGEO
                      IF ( ISET . EQ . 1 )          GO TO 2020
C-----THE LANE CHANGE IS FINISHED THUS FIND THE INTERSECTION PATH FOR
C-----THIS VEHICLE BASED ON THE CURRENT APPROACH, CURRENT LANE, AND THE
C-----DESIRED OUTBOUND APPROACH
      CALL  PATHF   ( LFALSE,N1,N2 )
 2020 CONTINUE
                      IF ( ISET . LE . 1 )          GO TO 2050
                      IF ( ISET . GE . 6 )          GO TO 2050
                      IF ( JSISET . NE . 0 )        GO TO 2050
```

```
      FLENV = 4.0*LENV(IVEHCL)
C-----IF THE DISTANCE TO THE END OF THE LANE IS GT 4 VEHICLE LENGTHS
C-----THEN DETERMINE IF A LANE CHANGE IS DESIRABLE
                      IF ( RELEND . GT . FLENV )    GO TO 2040
C-----IF THE DISTANCE TO THE END OF THE LANE IS LT 2 VEHICLE LENGTHS
C-----THEN A LANE CHANGE SHOULD NO LONGER BE CONSIDERED
                      IF ( RELEND.LT.0.5*FLENV )    GO TO 2030
C-----IF THE LANE CHANGE IS FORCED (NOT OPTIONAL) WHEN THE DISTANCE TO
C-----THE END OF THE LANE IS BETWEEN 2 AND 4 VEHICLE LENGTHS THEN
C-----DETERMINE IF A LANE CHANGE IS DESIRABLE ELSE A LANE CHANGE SHOULD
C-----NO LONGER BE CONSIDERED
                      IF ( LEGAL . EQ . 1 )         GO TO 2040
                      IF ( LEGAL . EQ . 3 )         GO TO 2040
 2030 CONTINUE
C-----A LANE CHANGE SHOULD NO LONGER BE CONSIDERED
      ISET = 6
                      IF ( LNEXT . NE . 0 )         GO TO 2050
C-----THE VEHICLE CAN NOT CHANGE LANES AND IT HAS NOT YET FOUND AN
C-----INTERSECTION PATH THUS FORCE AN INTERSECTION PATH TO BE FOUND FOR
C-----THIS VEHICLE BASED ON THE CURRENT APPROACH, CURRENT LANE, AND THE
C-----DESIRED OUTBOUND APPROACH
      CALL  PATHF   ( LTRUE,N1,N2 )
      GO TO 2050
 2040 CONTINUE
                      IF ( VELOLD . LT . 5.0 )      GO TO 2050
C-----DETERMINE IF A LANE CHANGE IS DESIRABLE
      CALL  LCHDES
 2050 CONTINUE
C-----CHECK THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES, CALL THE
C-----APPROPRIATE ACC/DEC ROUTINES, AND COMPUTE THE VEHICLES NEW POS/
C-----VEL/ACC
      CALL  ACDCP
C7    POSLAT = LATPOS/8.0 - 15.0
C7                    IF ( LCHGE . NE . 2 )         POSLAT = 0.0
C7    IF ( ABS(LEGAL/2.0-ABS(POSLAT)).LE.0.1 )     POSLAT = 0.0
C7    WRITE (4,702)  TIME,IQ(IV),IONE,IA,IL,IVEHCL,POSNEW,POSLAT
C0                    IF ( IPRTLO . EQ . 0 )        GO TO 110
C-----PRINT POS/VEL/ACC FOR THE VEHICLE
CZ    CALL  PVAPRT
C0110 CONTINUE
                      IF ( ICONTR . LT . 6 )        GO TO 2060
                      IF ( NLDL . LE . 0 )          GO TO 2060
C-----CHECK EACH DETECTOR FOR THIS LANE TO SEE IF THIS VEHICLE TRIPPED
C-----ANY OF THEM THIS DT
      CALL  CHKLDT
 2060 CONTINUE
C-----UPDATE THE VEHICLES SIMULATION STATISTICS ON THE INBOUND APPROACH
      CALL  SSIBAP  ( POSCHK,INQUE )
                      IF ( LOGFLG - 1 )             2080 , 2070 , 2080
 2070 CONTINUE
C-----CHECK THE INTERSECTION CONTROL LOGICAL DEPENDENT ATTRIBUTES AND
C-----CALL THE APPROPRIATE INTERSECTION CONTROL ROUTINES
      CALL  INTLOG
 2080 CONTINUE
                      IF ( MFINL . EQ . LFALSE )    GO TO 2090
            IF ( FLOAT(LGEOM(4)) . GT . POSNEW ) GO TO 3010
                      IF ( MPRO . EQ . LFALSE )     GO TO 3010
                      IF ( IPRTM . GT . 1 )         GO TO 3010
C-----LOG THE VEHICLE OUT OF THE INBOUND APPROACH AND LANE AND INTO THE
C-----LINKING INTERSECTION PATH FOR THE VEHICLE
      CALL  LOGIBI
C3    KPFLAG = 10HENTER INTR
      GO TO 3010
 2090 CONTINUE
                      IF ( PVPOS+4.0.GT.POSNEW )    GO TO 3010
C-----PRINT THE COLLISION INFORMATION AND RESET THE VEHICLES POS/VEL/ACC
      CALL  BANGS   ( 1 )
 3010 CONTINUE
C-----BIAS THE VEHICLE ATTRIBUTES, SET THE PREVIOUS VEHICLE PARAMETERS,
C-----AND UPDATE THE MAXIMUM ACC/DEC FOR THE VEHICLE
      CALL  BIAS
```

```
C-----PRINT SELECTED ATTRIBUTES FOR VEHICLE IV
                      IF ( JPRTM . NE . 0 )         IPRTM = JPRTM
CV                    IF ( IPRTLO . EQ . 0 )        GO TO 111
CU                    IF ( TIME . LT . TPRINT )     GO TO 111
C3                    IF ( JPRTM . GT . 0 )         JPFLAG = 10HPIJR TIME
CS    IDESPD = DESVEL + 0.5
CS    POSLAT = LATPOS/8.0 - 15.0
CS                    IF ( LCHGE . NE . 2 )         POSLAT = 0.0
CS    PRINT 703 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
CS   *            SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
CS   *            LOGFLG,LCHGE,IPRTM,POSLAT,INQUE,ISISET(ICAMPC,IBLN),
CS   *            IPFLAG,JPFLAG,KPFLAG
CT    IDESPD = DESVEL + 0.5
CT    POSLAT = LATPOS/8.0 - 15.0
CT                    IF ( LCHGE . NE . 2 )         POSLAT = 0.0
CT    PRINT 703 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
CT   *            SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
CT   *            LOGFLG,LCHGE,IPRTM,POSLAT,INQUE,ISISET(ICAMPC,IBLN)
CU111 CONTINUE
 3020 CONTINUE
C-----REPACK THE ATTRIBUTES FOR VEHICLE IV
      LOGFLG = LOGTMP
C   COLEASE,REPACK,VEHD,IV
      CALL  REPACK  (      6,IV      )                                  COLEASE
                      IF ( IREPFX . EQ . LFALSE )  GO TO 3030
C   COLEASE,REPACK,VEHF,IV
      CALL  REPACK  (      7,IV      )                                  COLEASE
 3030 CONTINUE
                      IF ( IREPIL . EQ . LFALSE )  GO TO 3040
C   COLEASE,REPACK,VEHIL,IV
      CALL  REPACK  (      8,IV      )                                  COLEASE
 3040 CONTINUE
C-----SET THE ENTRY NUMBER FOR THE VEH ENTITIES FOR THE NEXT VEHICLE ON
C-----THE INBOUND LANE TO BE PROCESSED
      IV = NXVEH
CY                    IF ( IPRTLO . EQ . 0 )        GO TO 112
CX                    IF ( TIME . LT . TPRINT )     GO TO 112
CX    PRINT 704 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
CX112 CONTINUE
C-----END OF VEHICLE LOOP
 4010 CONTINUE
                      IF ( LQ(IAN,ILN) . LE . 0 )  GO TO 5010
 4020 CONTINUE
      NRNAME = 1
C-----LOG THE NEW VEHICLE INTO THE INBOUND APPROACH AND LANE AND
C-----INITIALIZE THE VEHICLE ATTRIBUTES
      CALL  LOGIN
CY                    IF ( IPRTLO . EQ . 0 )        GO TO 113
CX                    IF ( TIME . LT . TPRINT )     GO TO 113
CX    PRINT 704 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
CX113 CONTINUE
C-----END OF INBOUND LANE LOOP
 5010 CONTINUE
C-----END OF INBOUND APPROACH LOOP
 6010 CONTINUE
      RETURN
      END                                                               IBAP
```

```
      SUBROUTINE LOKIBI                                                  COLEASE
C     TASK,LOKIBI
      COMMON / LANE   / LWID      ,NLL      ,NLR      ,ISNA     ,      COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL    ,ILVL     ,      COLEASE
     *                  LCONTR    ,LTURN    ,LGEOM ( 4),NLDL    ,      COLEASE
     *                  LLDL  ( 5),IBLN     ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                   COLEASE
      COMMON / VEHD   / ISLP      ,IACC     ,IVEL     ,IPOS     ,      COLEASE
     *                  ISET      ,LCHGE    ,ISPDP    ,LEGAL    ,      COLEASE
     *                  IPRTM     ,ITIMV    ,IQDS     ,ISPOS    ,      COLEASE
     *                  ISDS      ,IDVS     ,ISTCON   ,IVMAXA   ,      COLEASE
     *                  IVMAXD    ,LATPOS   ,IDTS     ,LALT     ,      COLEASE
     *                  NORC      ,LOGFLG   ,MSTPF    ,MLAG     ,      COLEASE
     *                  MTCARS    ,MFINL    ,MSFLG    ,MPOBS    ,      COLEASE
     *                  MOASF     ,MSAOR    ,MPRO     ,MBLOCK   ,      COLEASE
     *                  MININT    ,IFVA     ,IACDS    ,ICDFS    ,      COLEASE
     *                  ISDEC     ,ISTMO    ,IACLDS   ,IRSTOP          COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL   ,ISPD     ,NOF      ,      COLEASE
     *                  NOR       ,LNEXT    ,LPRES    ,ITURN    ,      COLEASE
     *                  IBAPS     ,IPRTLO   ,IEXTIM   ,NOBAPD          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / JACC,JPOS,JVEHCL,JVEL,LGEOM1,MENP,MOBL,
     *                  ZTEMPD(103)
      DATA      N1,N2  / 4HLOKI,2HBI /
C
C-----SUBROUTINE LOKIBI LOOKS AHEAD INTO THE LINKING INTERSECTION PATH
C-----FOR THIS VEHICLE AND IF THERE IS A VEHICLE ON THE INTERSECTION
C-----PATH THEN RESET THE PREVIOUS VEHICLE PARAMETERS TO THE LAST
C-----VEHICLE ON THE INTERSECTION PATH ELSE RESET THE PREVIOUS VEHICLE
C-----PARAMETERS TO THE END OF THE INTERSECTION PATH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----FIND THE ENTRY NUMBER FOR THE LAST VEHICLE ON THE LINKING
C-----INTERSECTION PATH FOR THE VEHICLE
C     COLEASE,FIND,IVPV,PATH,LNEXT,ILVP
      CALL  FIND    (IVPV  ,      4,LNEXT ,      6)                    COLEASE
                    IF ( IVPV . EQ . 0 )        GO TO 1020
      MENP = 0
 1010 CONTINUE
C-----RESET THE PREVIOUS VEHICLE PARAMETERS TO THE LAST VEHICLE ON THE
C-----LINKING INTERSECTION PATH FOR THE VEHICLE
C     COLEASE,STORE,LFALSE,VEHD,IV,MFINL
      CALL  STORE   (LFALSE,      6,IV    ,     26)                    COLEASE
      MFINL = LFALSE
C     COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
      CALL  FIND    (JVEHCL,      7,IVPV  ,      2)                    COLEASE
C     COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
      CALL  FIND    (JPOS  ,      6,IVPV  ,      4)                    COLEASE
      PVPOS = LGEOM(4) + MENP + JPOS/25.0 - LENV(JVEHCL) - 4.0
C     COLEASE,FIND,JVEL,VEHD,IVPV,IVEL
      CALL  FIND    (JVEL  ,      6,IVPV  ,      3)                    COLEASE
      PVVEL = JVEL/25.0
C     COLEASE,FIND,JACC,VEHD,IVPV,IACC
      CALL  FIND    (JACC  ,      6,IVPV  ,      2)                    COLEASE
      PVACC = JACC/312.5 - 32.0
      RETURN
 1020 CONTINUE
C     COLEASE,FIND,MOBL,PATH,LNEXT,LOBL
      CALL  FIND    (MOBL  ,      4,LNEXT ,      4)                    COLEASE
C     COLEASE,FIND,IVPV,LANE,MOBL,ILVL
      CALL  FIND    (IVPV  ,      3,MOBL  ,     14)                    COLEASE
C     COLEASE,FIND,MENP,PATH,LNEXT,LENP
      CALL  FIND    (MENP  ,      4,LNEXT ,      1)                    COLEASE
                    IF ( IVPV . EQ . 0 )        GO TO 1030
C     COLEASE,FIND,LGEOM1,LANE,MOBL,LGEOM(1)
      CALL  FIND    (LGEOM1,      3,MOBL  ,     17)                    COLEASE
      MENP = MENP - LGEOM1
      GO TO 1010
 1030 CONTINUE
C-----RESET THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE LINKING
C-----INTERSECTION PATH FOR THE VEHICLE
      PVPOS = LGEOM(4) + MENP
      PVVEL = 0.0
      PVACC = 0.0
      RETURN
      END                                                                LOKIBI
```

```
      SUBROUTINE CHKDSP                                                 COLEASE
C     TASK,CHKDSP
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IGDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / MIMP,SLOPE,SPD,T,XCRIT,ZTEMPD(105)
      DATA     N1,N2  / 4HCHKD,2HSP /
C
C-----SUBROUTINE CHKDSP CHECKS TO SEE IF THE VEHICLE SHOULD RESET HIS
C-----DESIRED SPEED TO THE DESIRED SPEED OF HIS INTERSECTION PATH SO
C-----THAT HE CAN GRADUALLY DECELERATE TO HIS NEW DESIRED SPEED BEFORE
C-----HE ENTERS THE INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C     COLEASE,FIND,MIMP,PATH,LNEXT,LIMP
      CALL  FIND    (MIMP  ,      4,LNEXT ,      7)                     COLEASE
C-----FIND THE DESIRED SPEED FOR THE INTERSECTION PATH
      SPD = FLOAT(ISPD)*FLOAT(MIMP)/FLOAT(ISLIM)
                    IF ( RELEND . LE . 25.0 )     GO TO 1010
                    IF ( VELOLD . LT . SPD )      RETURN
C-----FIND THE DISTANCE REQUIRED TO REDUCE THE PRESENT VELOCITY OF THE
C-----VEHICLE TO THE DESIRED SPEED OF THE INTERSECTION PATH USING SLOPE
      SLOPE = -1.5*DCHAR(IDRICL)
      T = (-ACCOLD-SQRT(ACCOLD**2-2.0*SLOPE*(VELOLD-SPD)))/SLOPE + DT
      XCRIT = VELOLD*T + 0.5*ACCOLD*T**2 + SLOPE*T**3/6.0
                    IF ( RELEND . GT . XCRIT )    RETURN
 1010 CONTINUE
C-----SET THE VEHICLES DESIRED SPEED TO THE DESIRED SPEED FOR THE
C-----INTERSECTION PATH AND SET THE FLAG TO INDICATE THAT THE VEHICLES
C-----DESIRED SPEED HAS BEEN RESET
      ISPD = SPD + 0.5
      DESVEL = ISPD
C     COLEASE,STORE,ISPD,VEHF,IV,ISPD
      CALL  STORE   (ISPD  ,      7,IV    ,      3)                     COLEASE
      ISPDP = 1
      RETURN
      END                                                               CHKDSP
```

```
      SUBROUTINE CHKLDT                                                 COLEASE
C     TASK,CHKLDT
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / ILDL,JLDL,POSNRB,POSORB,STOP,STRT,ZTEMPD(104)
      DATA     IPULS  / 4HPULS /
      DATA     N1,N2  / 4HCHKL,2HDT /
C/701 FORMAT(3H IV,I5,9H POSORB =,F6.1,9H POSOLD =,F6.1,9H POSNRB =,
C/   *       F6.1,9H POSNEW =,F6.1,7H JLDL =,I2,9H STRTLD =,F6.1,
C/   *       9H STOPLD =,F6.1,2X,A4)
C
C-----SUBROUTINE CHKLDT CHECKS EACH DETECTOR FOR THIS LANE TO SEE IF
C-----THIS VEHICLE TRIPPED ANY OF THEM THIS DT
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      POSORB = POSOLD - LENV(IVEHCL)
      POSNRB = POSNEW - LENV(IVEHCL)
C-----CHECK EACH DETECTOR FOR THIS LANE TO SEE IF THIS VEHICLE TRIPPED
C-----ANY OF THEM THIS DT
      DO 1020  ILDL = 1 , NLDL
      JLDL = LLDL(ILDL)
C-----IF THE THE DETECTOR HAS ALREADY BEEN TRIPPED THEN SKIP TO THE NEXT
C-----DETECTOR
                    IF ( LDTRIP(JLDL) )           GO TO 1020
      STRT = STRTLD(JLDL)
      STOP = STOPLD(JLDL)
C-----IF THE FRONT BUMPER CROSSED THE START OF THE DETECTOR THEN THE
C-----DETECTOR IS TRIPPED
      IF ( (STRT-POSOLD)*(STRT-POSNEW).LE.0.0 )  GO TO 1010
C-----IF THE REAR BUMPER CROSSED THE START OF THE DETECTOR THEN THE
C-----DETECTOR IS TRIPPED
      IF ( (STRT-POSORB)*(STRT-POSNRB).LE.0.0 )  GO TO 1010
C-----IF THE FRONT BUMPER IS BETWEEN THE START AND END OF THE DETECTOR
C-----THEN THE DETECTOR IS TRIPPED
      IF ( (POSNEW-STRT)*(POSNEW-STOP).LE.0.0 )  GO TO 1010
C-----IF THE REAR BUMPER IS BETWEEN THE START AND END OF THE DETECTOR
C-----THEN THE DETECTOR IS TRIPPED
      IF ( (POSNRB-STRT)*(POSNRB-STOP).LE.0.0 )  GO TO 1010
C-----IF THE DETECTOR TYPE = (PULSE    ) THEN THE DETECTOR HAS NOT BEEN
C-----TRIPPED AND SKIP TO THE NEXT DETECTOR
                    IF ( ITYPLD(JLDL).EQ.IPULS ) GO TO 1020
C-----THIS DETECTOR TYPE = (PRESENCE) THUS IF THE VEHICLE IS STRADDLING
C-----THE DETECTOR THEN THE DETECTOR IS TRIPPED ELSE THE DETECTOR HAS
C-----NOT BEEN TRIPPED AND SKIP TO THE NEXT DETECTOR
      IF ( (POSNRB-STRT)*(POSNEW-STOP).LE.0.0 )  GO TO 1010
      GO TO 1020
 1010 CONTINUE
C-----SET THE DETECTOR TRIPPED
      LDTRIP(JLDL) = .TRUE.
```

EE M

```
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 102
C/                  IF ( TIME . LT . TPRINT )    GO TO 101
C/    PRINT 701 , IV,POSORB,POSOLD,POSNRB,POSNEW,JLDL,STRT,STOP,
C/   *              ITYPLD(JLDL)
C/101 CONTINUE
C5102 CONTINUE
C-----END OF DETECTOR LOOP
 1020 CONTINUE
      RETURN
      END                                                        CHKLDT
```

```
      SUBROUTINE SSIBAP ( POSCHK,INQUE )                                COLEASE
C   TASK,SSIBAP,POSCHK,INQUE
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,     COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / MIMP,SPFACT,ZTEMPD(108)
      LOGICAL           INQUE
      DATA      N1,N2 / 4HSSIB,2HAP /
C
C-----SUBROUTINE SSIBAP UPDATES THE VEHICLES SIMULATION STATISTICS ON
C-----THE INBOUND APPROACH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                  IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INCREMENT THE TRAVEL TIME
      ITIMV = ITIMV + 1
C-----IF THIS VEHICLE HAS ALREADY GATHERED QUEUE DELAY THEN THE QUEUE
C-----IS NOT BROKEN AND CONTINUES TO GATHER QUEUE DELAY
                  IF ( IQDS . GT . 0 )         GO TO 1010
C-----THIS VEHICLE HAS NOT GATHERED ANY QUEUE DELAY YET THEN IF THE
C-----QUEUE IS ALREADY BROKEN THEN THIS VEHICLE MAY NOT JOIN THE QUEUE
                  IF ( . NOT . INQUE )         GO TO 1010
C-----IF THIS VEHICLE IS MOVING FASTER THAN 3.0 FPS OR THIS VEHICLE IS
C-----MORE THAN XQDIST FEET FROM THE VEHICLE IN FRONT OF HIM (OR THE END
C-----OF THE LANE FOR THE FIRST VEHICLE IN THE LANE) THEN THE QUEUE IS
C-----BROKEN FOR THIS LANE
                  IF ( VELNEW . GT . 3.0 )      INQUE = .FALSE.
          IF ( POSCHK-POSNEW . GT . XQDIST )    INQUE = .FALSE.
 1010 CONTINUE
C-----IF THIS VEHICLE IS IN THE QUEUE THEN INCREMENT QUEUE DELAY
                  IF ( INQUE )                  IQDS = IQDS + 1
C-----IF THE VEHICLE IS STOPPED IN A QUEUE THEN INCREMENT STOPPED DELAY
          IF ( INQUE . AND . VELNEW.LE.3.0 )    ISDS = ISDS + 1
C-----IF THE VELOCITY IS LE XFPS THEN INCREMENT THE DELAY BELOW XX MPH
                  IF ( VELNEW . LE . XFPS )     IDVS = IDVS + 1
      SPFACT = 1.0
                  IF ( ISPDP . EQ . 0 )         GO TO 1020
C-----THE VEHICLE HAS RESET HIS DESIRED SPEED TO THE DESIRED SPEED FOR
C-----HIS INTERSECTION PATH THUS FIND THE FACTOR REQUIRED TO MAKE HIS
C-----CURRENT DESIRED SPEED BE THE VALUE FOR THIS APPROACH FOR SUMMATION
C-----FOR THE AVERAGE DESIRED SPEED
```

549

```
C     COLEASE,FIND,MIMP,PATH,LNEXT,LIMP
      CALL  FIND    (MIMP   ,     4,LNEXT  ,      7)                     COLEASE
      SPFACT = FLOAT(ISLIM)/FLOAT(MIMP)
 1020 CONTINUE
C-----ADD THE DESIRED SPEED FOR THIS DT FOR THE AVERAGE DESIRED SPEED
      ISPDS = ISPDS + ISPD*SPFACT + 0.5
                    IF ( TIME . LE . STRTIM )     RETURN
                    IF ( . NOT . INQUE )          RETURN
C-----THE VEHICLE HAS ACCUMULATED QUEUE DELAY SO UPDATE THE MAXIMUM
C-----QUEUE LENGTH AND INCREMENT THE NUMBER OF VEHICLES IN THE QUEUE
      MQUEUE(IAN,ILN) = MAX0(MQUEUE(IAN,ILN),IVN)
      LQUEUE(IAN,ILN) = LQUEUE(IAN,ILN) + 1
      RETURN
      END                                                                SSIBAP
```

```
      SUBROUTINE LOGIBI                                                  COLEASE
C     TASK,LOGIBI
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,     COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)      COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,     COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,     COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLOL      ,     COLEASE
     *                  LLOL  ( 5),IBLN      ,IDUMLA                     COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,     COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,     COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,     COLEASE
     *                  ILCH      ,IGEOCP(60)                            COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,     COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,     COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,     COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,     COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,     COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,     COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,     COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,     COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,     COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP          COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,     COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,     COLEASE
     *                  ISAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
C6    COMMON / PRTPVA / DISTAD(200)
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / DTIME,I,J,JVEL,LPREV,MOBAP,MOGFLG,NSKP,NVILL,
     *                  POSTOT,ZTEMPD(100)
      DATA      N1,N2  / 4HLOGI,2HBI /
C8601 FORMAT(1X,2I2,1(I4),=X,F6.2)
C
C-----SUBROUTINE LOGIBI LOGS THE VEHICLE OUT OF THE INBOUND APPROACH AND
C-----LANE AND INTO THE LINKING INTERSECTION PATH FOR THE VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----REMOVE THE VEHICLE FROM THE LIST OF VEHICLES AT THE INTERSECTION
      J = 0
      DO 1010  I = 1 , NVATIN
                    IF ( LVATIN(I) . EQ . IV )    J = J + 1
      LVATIN(I) = LVATIN(I+J)
      TVATIN(I) = TVATIN(I+J)
 1010 CONTINUE
      NVATIN = NVATIN - J
                    IF ( LNEXT . EQ . 0 )         GO TO 2010
                    IF ( ISET . NE . 1 )          GO TO 2020
 2010 CONTINUE
C-----END THE LANE CHANGE AND RESET THE LANE CHANGE FLAGS
      CALL  ENDLCH
C-----FORCE AN INTERSECTION PATH TO BE FOUND FOR THIS VEHICLE BASED ON
C-----THE CURRENT APPROACH, CURRENT LANE, AND THE DESIRED OUTBOUND
C-----APPROACH
      CALL  PATHF   ( LTRUE,N1,N2 )
```

```
C-----SET CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
      CALL  SETCON
 2020 CONTINUE
C-----EXTRACT THE LINKING INTERSECTION PATH FOR THE VEHICLE
                    IF ( IPTHUP . EQ . LNEXT )   GO TO 3010
C   COLEASE,EXTRAC,PATH,LNEXT
      CALL  EXTRAC  (      4,LNEXT )                                COLEASE
      IPTHUP = LNEXT
 3010 CONTINUE
C-----SET THE VEHICLES NOF TO THE LAST VEHICLE ON THE LINKING
C-----INTERSECTION PATH
      NOF = ILVP
C-----SET THIS VEHICLE AS THE NEW LAST VEHICLE ON THE LINKING
C-----INTERSECTION PATH
C   COLEASE,STORE,IV,PATH,LNEXT,ILVP
      CALL  STORE   (IV     ,       4,LNEXT ,       6)              COLEASE
      ILVP = IV
C-----DECREMENT THE NUMBER OF VEHICLES ON THE INBOUND APPROACH AND LANE
      NVIA(ISNA) = NVIA(ISNA) - 1
      NVIBA = NVIBA - 1
      NVILL = NVIL(ILN) - 1
C   COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
      CALL  STORE   (NVILL ,        1,ISNA   ,      7+ILN    )      COLEASE
      NVIL(ILN) = NVILL
C-----INCREMENT THE NUMBER OF VEHICLES ON THE INTERSECTION PATH
      NVIN = NVIN + 1
      NVIP(LNEXT) = NVIP(LNEXT) + 1
      MFINL = LFALSE
                    IF ( IFVP . NE . 0 )         GO TO 3020
C-----SET THE VEHICLE AS THE NEW FIRST VEHICLE ON THE INTERSECTION PATH
C   COLEASE,STORE,IV,PATH,LNEXT,IFVP
      CALL  STORE   (IV     ,       4,LNEXT ,       5)              COLEASE
      IFVP = IV
      MFINL = LTRUE
 3020 CONTINUE
C-----UPDATE THE LINK INDICES
      LPREV = LPRES
      LPRES = LNEXT
      LNEXT = LOBL
C-----SET THE FIRST VEHICLE IN THE INBOUND LANE AS THE NOR OF THIS
C-----VEHICLE
C   COLEASE,STORE,NOR,LANE,LPREV,IFVL
      CALL  STORE   (NOR    ,       3,LPREV ,      13)              COLEASE
      IFVL = NOR
                    IF ( NOR . NE . 0 )          GO TO 3030
C-----SET THE LAST VEHICLE IN THE INBOUND LANE = 0 (OLD NOR EQ 0)
C   COLEASE,STORE,0,LANE,LPREV,ILVL
      CALL  STORE   (0      ,       3,LPREV ,      14)              COLEASE
      ILVL = 0
      GO TO 3040
 3030 CONTINUE
C-----SET MFINL AND MOASF TO LTRUE, RESET IACC TO SLIGHTLY DECELERATING
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE 0 FOR NOF OF THE NOR VEHICLE
C-----(OLD NOR NE 0)
      CALL  FLGNOR ( LTRUE,0 )
C-----WAKE THE NOR VEHICLE UP FOR INTERSECTION CONTROL LOGIC
C   COLEASE,FIND,MOGFLG,VEHD,NOR,LOGFLG
      CALL  FIND    (MOGFLG,        6,NOR    ,      22)             COLEASE
                    IF ( MOGFLG . LE . 2 )       GO TO 3040
C   COLEASE,STORE,2,VEHD,NOR,LOGFLG
      CALL  STORE   (      2,       6,NOR    ,      22)             COLEASE
 3040 CONTINUE
C-----SET THIS VEHICLES NOR = 0
      NOR = 0
      MOASF = LTRUE
      ISTCON = 1
                    IF ( ISPDP . NE . 0 )        GO TO 3050
C-----THE VEHICLE HAS NOT PREVIOUSLY RESET HIS DESIRED SPEED THUS SET
C-----THE DESIRED SPEED FOR THE INTERSECTION PATH
      ISPD = FLOAT(ISPD)*FLOAT(LIMP)/FLOAT(ISLIM) + 0.5
 3050 CONTINUE
C-----IF THIS VEHICLE IS AN AMBER-GO FROM A STOPPED POSITION AT THE STOP
C-----LINE AND HAS CLEARED HIS INTERSECTION CONFLICTS THEN DOUBLE THE
C-----DESIRED SPEED FOR THE VEHICLE ON THE INTERSECTION PATH
                    IF ( LATPOS . EQ . LTRUE )   ISPD = 2*ISPD
C-----RESET SOME OF THE VEHICLES ATTRIBUTES
C8    POSTOT = POSNEW - POSOLD
      POSNEW = POSNEW - LGEOM(4)
C8    DTIME = TIME
C8                  IF ( POSTOT . LE . 0.0 )     GO TO 101
C8    DTIME = TIME - (POSNEW/POSTOT)*DT
C8101 CONTINUE
C8    NSKP = (IAN-1)*26 + (ILN-1)*6 + 2
C8    PRINT 601 , IA,ILN,IQ(IV),NSKP,DTIME
      MININT = LTRUE
      LOGTMP = 0
      IREPFX = LTRUE
C6    DISTAD(IV) = LGEOM(4)
                    IF ( NOF . EQ . 0 )          RETURN
C-----SET THIS VEHICLE AS THE NOR OF THE LAST VEHICLE ON THE
C-----INTERSECTION PATH
C   COLEASE,STORE,IV,VEHF,NOF,NOR
      CALL  STORE   (IV     ,       7,NOF    ,      5)              COLEASE
C-----CHECK IF THE VEHICLE AHEAD IS STOPPED
      MOASF = LFALSE
C   COLEASE,FIND,JVEL,VEHD,NOF,IVEL
      CALL  FIND    (JVEL  ,        6,NOF    ,      3)              COLEASE
                    IF ( JVEL . LE . 0 )         MOASF = LTRUE
      RETURN
      END                                                           LOGIBI
```

```
      SUBROUTINE PRESTI (ININT)                                         COLEASE
C     TASK,PRESTI,ININT
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA     ,      COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL     ,      COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL     ,      COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                     COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,      COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL    ,      COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,      COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,      COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,      COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,      COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,      COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,      COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,      COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP          COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF      ,      COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN    ,      COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / JACC,JPOS,JVEHCL,JVEL,ZTEMPD(106)
      DATA      N1,N2 / 4HPRES,2HT1 /
C
C-----SUBROUTINE PRESTI EXTRACTS ENTRY IV OF ENTITY VEHF, RESETS THE
C-----PREVIOUS VEHICLE PARAMETERS TO THE NEW NOF IF THE VEHICLE IS LANE
C-----CHANGING, AND INITIALIZES SEVERAL PARAMETERS FOR THE VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      IREPFX = LFALSE
C3    IPFLAG = 10H
C3    JPFLAG = 10H
C3    KPFLAG = 10H
C-----EXTRACT ENTRY IV OF ENTITY VEHF
C     COLEASE,EXTRAC,VEHF,IV
      CALL  EXTRAC  (      7,IV     )                                  COLEASE
                    IF ( ININT . EQ . LTRUE )    GO TO 1010
C     COLEASE,FIND,MBLOCK,VEHD,IV,MBLOCK
      CALL  FIND    (MBLOCK,      6,IV     ,      32)                  COLEASE
                    IF ( MBLOCK . EQ . LTRUE )   ENDLN = LGEOM(2)
                    IF ( LCHGE . NE . 2 )        GO TO 1010
                    IF ( NOF . EQ . 0 )          GO TO 1010
C-----THE VEHICLE IS CHANGING LANES SO RESET THE PREVIOUS VEHICLE
C-----PARAMETERS TO THE NEW NOF VEHICLE
C     COLEASE,FIND,JPOS,VEHD,NOF,IPOS
      CALL  FIND    (JPOS  ,      6,NOF    ,       4)                  COLEASE
C     COLEASE,FIND,JVEHCL,VEHF,NOF,IVEHCL
      CALL  FIND    (JVEHCL,      7,NOF    ,       2)                  COLEASE
      PVPOS = JPOS/25.0 - LENV(JVEHCL) - 4.0
C     COLEASE,FIND,JVEL,VEHD,NOF,IVEL
      CALL  FIND    (JVEL  ,      6,NOF    ,       3)                  COLEASE
      PVVEL = JVEL/25.0
C     COLEASE,FIND,JACC,VEHD,NOF,IACC
      CALL  FIND    (JACC  ,      6,NOF    ,       2)                  COLEASE
      PVACC = JACC/312.5 - 32.0
 1010 CONTINUE
C-----INITIALIZE SEVERAL PARAMETERS FOR THE VEHICLE
      IVPV = NOF
C     COLEASE,FIND,MFINL,VEHD,IV,MFINL
      CALL  FIND    (MFINL ,      6,IV     ,      26)                  COLEASE
      RETURN
```

```
      SUBROUTINE PREST2                                                  COLEASE
C     TASK,PREST2
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,     COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,     COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,     COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,     COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,     COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,     COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,     COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,     COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,     COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / ZTEMPD(110)
      DATA     N1,N2  / 4HPRES,2HT2 /
C
C-----SUBROUTINE PREST2 COMPUTES NEW ACC/DEC LOGIC AND EXTRACTS ENTRY IV
C-----OF ENTITY VEHD FOR THE VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME .GT. NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET PARAMETERS FOR NEW ACC/DEC LOGIC
      MOASF = LFALSE
                    IF ( PVVEL .LE. 0.0 )      MOASF = LTRUE
C     COLEASE,STORE,MOASF,VEHD,IV,MOASF
      CALL  STORE   (MOASF ,      6,IV     ,     29)                     COLEASE
C     COLEASE,FIND,JPRTM,VEHD,IV,IPRTM
      CALL  FIND    (JPRTM ,      6,IV     ,      9)                     COLEASE
      JPRTM = MAX0(JPRTM-1,0)
                    IF ( JPRTM .GT. 0 )        GO TO 1010
C-----COMPUTE NEW ACC/DEC LOGIC
C     COLEASE,LOGIC,VEHD,IV
      CALL  LOGIC   (      6,IV     )                                    COLEASE
 1010 CONTINUE
C-----EXTRACT ENTRY IV OF ENTITY VEHD
C     COLEASE,EXTRAC,VEHD,IV
      CALL  EXTRAC  (      6,IV     )                                    COLEASE
      IPRTM = JPRTM
      RETURN
      END                                                                PREST2
```

```
      SUBROUTINE UNBIAS                                                  COLEASE
C     TASK,UNBIAS
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,     COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,     COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,     COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,     COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,     COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,     COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,     COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,     COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,     COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP          COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,     COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,     COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / ZTEMPD(110)
      DATA     N1,N2  / 4HUNBI,2HAS /
C
C-----SUBROUTINE UNBIAS UNBIASES THE VEHICLE ATTRIBUTES AND PREDICTS THE
C-----NEW POS/VEL/ACC
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME .GT. NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----UNBIAS THE VEHICLE ATTRIBUTES
      SLPOLD = ISLP/400.0 - 12.0
                    IF ( ISLP .EQ. 4800 )      SLPOLD = 0.0
      ACCOLD = IACC/312.5 - 32.0
                    IF ( IACC .EQ. 10000 )     ACCOLD = 0.0
      VELOLD = IVEL/25.0
      POSOLD = IPOS/25.0
C-----INITIALIZE SEVERAL VEHICLE PARAMETERS
      RELEND = ENDLN - POSOLD
      SLPNEW = SLPOLD
      OLDDTS = IDTS
      DESVEL = ISPD
      SLPLCH = 0.0
      ISIDE = 2
      LEADSP = 0
      LAGSPD = 0
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER DT SECONDS
      CALL  NEWVEL  ( DT,DTSQ,DTCU )
      RETURN
      END                                                                UNBIAS
```

```
      SUBROUTINE NEWVEL (T,      TSQ,      TCU)                             COLEASE
C     TASK,NEWVEL,T,TSQ,TCU
      COMMON / LOGICV / LTRUE,LFALSE                                        COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,         COLEASE
     *                  ISET      ,LCHGE     ,ISPOP     ,LEGAL    ,         COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,         COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,         COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,         COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,         COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,         COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,         COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,         COLEASE
     *                  ISDEC     ,ISTMO     ,IACLOS    ,IRSTOP             COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DATA      N1,N2 / 4HNEWV,2HEL /
C
C-----SUBROUTINE NEWVEL CALCULATES THE POS/VEL/ACC FOR THE VEHICLE AFTER
C-----T SECONDS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                      IF ( SLPLCH . EQ . 0.0 )     GO TO 1030
                      IF ( SLPNEW - SLPLCH )       1010 , 1030 , 1020
 1010 CONTINUE
                      IF ( ISET . EQ . 3 )         ISET = 4
      GO TO 1030
 1020 CONTINUE
C-----THE ACC/DEC SLOPE FOR A LANE CHANGE IS NOT ZERO AND IS LT THE
C-----SLOPE CALCULATED BY ACDCP THUS USE THE ACC/DEC SLOPE FOR A LANE
C-----CHANGE
C3    KPFLAG = 10HSLPLCH MIN
      SLPNEW = SLPLCH
      MSFLG = LFALSE
      IPRTM = 0
      JPRTM = 0
 1030 CONTINUE
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER T SECONDS
      ACCNEW = ACCOLD + SLPNEW*T
      VELNEW = VELOLD + ACCOLD*T + 0.5*SLPNEW*TSQ
      DPOS = VELOLD*T + 0.5*ACCOLD*TSQ + SLPNEW*TCU/6.0
           IF ( VELNEW.LE.-0.01.AND.T.NE.DT )   DPOS = 0.0
      POSNEW = POSOLD + DPOS
C-----UPDATE SOME OF THE VEHICLE PARAMETERS
      IDTS = OLDDTS + DPOS*25.0 + 0.5
      RELVEL = PVVEL - VELNEW
      RELPOS = PVPOS - POSNEW
      RETURN
      END                                                                   NEWVEL
```

```
      SUBROUTINE LCHGEO                                                     COLEASE
C     TASK,LCHGEO
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,         COLEASE
     *                  ISET      ,LCHGE     ,ISPOP     ,LEGAL    ,         COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,         COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,         COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,         COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,         COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,         COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,         COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,         COLEASE
     *                  ISDEC     ,ISTMO     ,IACLOS    ,IRSTOP             COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF      ,         COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN    ,         COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD             COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / DVFACT,POSLAT,TLDIST,XNEW,XOLD,XTOT,ZTEMPD(104)
      DATA      N1,N2 / 4HLCHG,2HEO /
      DATA      PI    / 3.14159265358979 /
C
C-----SUBROUTINE LCHGEO COMPUTES THE NEW LATERAL POSITION FOR A LANE
C-----CHANGE USING A COSINE CURVE AND IF FINISHED THEN ENDS THE LANE
C-----CHANGE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                      IF ( VELOLD . LE . 0.0 )     RETURN
C-----FIND THE OLD LATERAL POSITION AND THE TOTAL LATERAL DISTANCE FOR
C-----THE LANE CHANGE
      DVFACT = DCHAR(IDRICL)*VCHAR(IVEHCL)
      POSLAT = LATPOS/8.0 - 15.0
      TLDIST = LEGAL/2.0
C-----DEFINE THE LENGTH OF THE LANE CHANGE TO BE 3.5 SECONDS AT THE OLD
C-----VELOCITY OF THE VEHICLE WITH A MINIMUM OF THE VEHICLE LENGTH
      XTOT = AMAX1(VELOLD*3.5/DVFACT,FLOAT(LENV(IVEHCL)))
C-----DEFINE THE PRESENT POSITION ON THE COSINE CURVE
C=    XOLD = XTOT*ACOS(2.0*ABS(POSLAT)/TLDIST-1.0)/PI
C;    XOLD = XTOT*ARCOS(2.0*ABS(POSLAT)/TLDIST-1.0)/PI
C-----UPDATE THE POSITION OF THE VEHICLE ON THE COSINE CURVE
      XNEW = XOLD + VELOLD*DT
C-----IF THE NEW POSITION OF THE VEHICLE ON THE COSINE CURVE IS GE
C-----95 PERCENT OF THE TOTAL LENGTH OF THE LANE CHANGE THEN GO TO 1010
C-----AND END THE LANE CHANGE AND RESET THE LANE CHANGE FLAGS
                      IF ( XNEW . GE . 0.95*XTOT )    GO TO 1010
C-----FIND THE NEW LATERAL POSITION FOR THE LANE CHANGE
      POSLAT = SIGN(0.5*TLDIST*(1.0+COS(PI*XNEW/XTOT)),POSLAT)
C-----BIAS THE NEW LATERAL POSITION FOR THE LANE CHANGE
      LATPOS = 8.0*(POSLAT+15.0) + 0.5
C-----IF THE NEW LATERAL POSITION FOR THE LANE CHANGE IS GT 0.3 FEET
C-----THEN RETURN ELSE END THE LANE CHANGE AND RESET THE LANE CHANGE
C-----FLAGS
                      IF ( ABS(POSLAT) . GT . 0.3 )RETURN
 1010 CONTINUE
C-----END THE LANE CHANGE AND RESET THE LANE CHANGE FLAGS
      CALL  ENDLCH
      RETURN
      END                                                                   LCHGEO
```

```
      SUBROUTINE ENDLCH                                                 COLEASE
C     TASK,ENDLCH
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      DATA      N1,N2   / 4HENDL,2HCH /
C
C-----SUBROUTINE ENDLCH ENDS THE LANE CHANGE AND RESETS THE LANE CHANGE
C-----FLAGS
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                        IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----END THE LANE CHANGE
      LEGAL = 4
      LNEXT = 0
      ISET = 5
      LATPOS = 0
      LCHGE = 1
                        IF ( NOF . EQ . 0 )          GO TO 1010
C-----RESET THE LANE CHANGE FLAGS FOR THIS VEHICLE
C     COLEASE,FIND,MCHGE,VEHD,NOF,LCHGE
      CALL  FIND     (MCHGE ,       6,NOF      ,        6)              COLEASE
                        IF ( MCHGE . EQ . 2 )        LCHGE = 3
 1010 CONTINUE
                        IF ( NOR . EQ . 0 )          RETURN
C-----RESET THE LANE CHANGE FLAG FOR THE NOR VEHICLE
C     COLEASE,FIND,MCHGE,VEHD,NOR,LCHGE
      CALL  FIND     (MCHGE ,       6,NOR      ,        6)              COLEASE
                        IF ( MCHGE . EQ . 3 )        MCHGE = 1
C     COLEASE,STORE,MCHGE,VEHD,NOR,LCHGE
      CALL  STORE    (MCHGE ,       6,NOR      ,        6)              COLEASE
      RETURN
      END                                                               ENDLCH
```

```
      SUBROUTINE LCHDFS                                                 COLEASE
C     TASK,LCHDES
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPOS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CARFQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / CARDEC,CARDIS,CRISLP,DECMAX,DENOM,JLCH,JSET,
     *                  LANSI,LOK,NOQ,OLDACC,RADICL,
     *                  RELDIS,RELSPD,SLPDEC,VSQT4,VT2,VCHKLS(6),
     *                  VSVEHU(5),VDELAY(14),VCKLAL(5),VGAPAC(28),
     *                  VCHGML(17),ZTEMPD(18)
      DIMENSION         MSG903(7),MSG904(7),MSG905(16)
      DATA      MSG903 / 4H LEG,4HAL N,4HOT C,4HHECK,4HED =,4H LCH,
     *                  4HDES  /
      DATA      MSG904 / 4H ILL,4HEGAL,4H TUR,4HN CO,4HDE =,4H LCH,
     *                  4HDES  /
      DATA      MSG905 / 4H TRY,4HING ,4HTO C,4HHANG,4HE LA,4HNES ,
     *                  4HWHEN,4H NO ,4HLANE,4H ALT,4HERNA,4HTIVE,
     *                  4H EXI,4HSTS ,4H= LC,4HHDES /
      DATA      N1,N2   / 4HLCHD,2HES /
C
C-----SUBROUTINE LCHDES DETERMINES IF A LANE CHANGE IS DESIRABLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                        IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----CHECK THE DESIRABILITY OF THE LANE CHANGE BASED ON LEGAL
      GO TO ( 1010,2010,1010,9030,9040 ) , LEGAL
 1010 CONTINUE
C-----THE TURN IS LEGAL FROM THE APPROACH BUT NOT FROM THIS LANE THUS
C-----SET WHICH SIDE THE VEHICLE SHOULD CHANGE TO
      ISIDE = LEGAL
C-----SET THE ENTRY NUMBER FOR THE LANE ENTITY OF THE LANE ON THE SIDE
C-----OF INTEREST TO CHECK
      LANSI = NLL
                        IF ( ISIDE . EQ . 3 )        LANSI = NLR
                        IF ( LANSI . EQ . 0 )        GO TO 9050
C-----CHECK THE LANE ON THE SIDE OF INTEREST TO SEE IF THE LANE IS
C-----AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE AND CLEAR TO THE
C-----INTERSECTION
      CALL  CHKLSI   ( LANSI,LOK )
C-----FIND THE NEAREST VEHICLE TO THE FRONT AND THE NEAREST VEHICLE TO
C-----THE REAR IN THE LANE ON THE SIDE OF INTEREST FOR THIS VEHICLE
```

```
      CALL  SVEHU    ( NOU )
C-----IF THE LANE ON THE SIDE OF INTEREST IS BLOCKED FOR THIS VEHICLE
C-----THEN CAR-FOLLOW THE NOF VEHICLE IN THAT LANE ELSE CHECK IF THERE
C-----IS AN ACCEPTABLE GAP TO LANE CHANGE INTO
                    IF ( LOK . NE . 0 )             GO TO 4010
      GO TO 2020
 2010 CONTINUE
C-----THE TURN IS LEGAL FROM THIS LANE BUT IF THE VEHICLE IS NOT
C-----DEDICATED TO AN INTERSECTION PATH THEN RETURN AND WAIT UNTIL THE
C-----VEHICLE IS DEDICATED TO AN INTERSECTION PATH
                    IF ( LNEXT . EQ . 0 )           RETURN
      JSET = ISET
      ISET = 6
C-----IF THERE ARE NO LANE ALTERNATIVES THEN RETURN AND DO NOT CHECK THE
C-----THE DESIRABILITY OF A LANE CHANGE ANY MORE
                    IF ( LALT . EQ . 1 )            RETURN
C    COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
      CALL  FIND     (JLCH  ,      4,LNEXT  ,     72)                   COLEASE
C-----IF THE VEHICLE IS THE FIRST VEHICLE IN THE LANE AND HIS
C-----INTERSECTION PATH DOES NOT CHANGE LANES WITHIN THE INTERSECTION
C-----THEN RETURN AND DO NOT CHECK THE DESIRABILITY OF A LANE CHANGE ANY
C-----MORE
      IF ( MFINL.EQ.LTRUE . AND . JLCH.EQ.0 )       RETURN
      ISET = JSET
C-----FIND THE LEGAL LANE FOR THE VEHICLE WITH THE MINIMUM EXPECTED
C-----DELAY
      CALL  DELAY
C-----IF THE VEHICLE SHOULD STAY IN THIS LANE THEN RETURN
                    IF ( ISIDE . EQ . 2 )           RETURN
      LANSI = NLL
                    IF ( ISIDE . EQ . 3 )           LANSI = NLR
                    IF ( LANSI . EQ . 0 )           GO TO 9050
 2020 CONTINUE
C-----CHECK IF THERE IS AN ACCEPTABLE GAP TO LANE CHANGE INTO AND IF NOT
C-----THEN DETERMINE THE APPROPRIATE DRIVER RESPONSE FOR LANE CHANGING
      CALL  GAPACC  ( LANSI )
C-----IF THERE IS AN ACCEPTABLE GAP THEN LOG THE VEHICLE OUT OF HIS
C-----PRESENT LANE AND INTO THE NEW LANE ELSE RESET THE LANE CHANGE FLAG
C-----AND RETURN
                    IF ( ISET . EQ . 1 )            GO TO 3010
      ISIDE = 2
      RETURN
 3010 CONTINUE
C-----THERE IS AN ACCEPTABLE GAP SO LOG THE VEHICLE OUT OF HIS PRESENT
C-----LANE AND INTO THE NEW LANE
      CALL  CHGMLN
      RETURN
 4010 CONTINUE
                    IF ( LOK . EQ . 2 )             GO TO 5010
                    IF ( NOSF . EQ . 0 )            GO TO 4020
C-----FIND THE ACC/DEC SLOPE TO CAR-FOLLOW THE NOF VEHICLE IN THE LANE
C-----ON THE SIDE OF INTEREST UNTIL THE LANE IS NO LONGER BLOCKED FOR
C-----THIS VEHICLE
      CRISLP = 4.0*DCHAR(IDRICL)
      VVSF = LEADSP/25.0
      RELSPD = VVSF - VELOLD
      RELDIS = AMAX1(PVSF-POSOLD,0.01)
      CARDIS = (1.7*VVSF + 4.0*RELSPD**2)/DCHAR(IDRICL)
                    IF ( RELDIS . GT . CARDIS )     RETURN
      CARDEC = CAREQA * ((VELOLD**CAREQM)/(RELDIS**CAREQL)) * RELSPD
      CARDEC = AMIN1(AMAX1(CARDEC,DMAX(IVEHCL)),-0.04/DT)
      SLPLCH = (CARDEC-ACCOLD)/DT
C-----BOUND THE ACC/DEC SLOPE FOR A LANE CHANGE
      SLPLCH = AMIN1(AMAX1(SLPLCH,-CRISLP),CRISLP)
      RETURN
 4020 CONTINUE
C-----FIND THE ACC/DEC SLOPE TO STOP AT THE END OF LANE ON THE SIDE OF
C-----INTEREST
      RELDIS = (PVSF - POSOLD)*0.9
      DENOM = 6.0*RELDIS
      VT2 = 2.0*VELOLD
      VSQT4 = VT2*VT2
      OLDACC = AMIN1(ACCOLD,0.0)
      RADICL = VSQT4 + DENOM*OLDACC
                    IF ( RADICL . LE . 0.0 )        RETURN
      DECMAX = -OLDACC - (VSQT4+VT2*SQRT(RADICL))/DENOM
             IF ( DECMAX . LE . DMAX(IDRICL) )      GO TO 5010
      SLPDEC = (OLDACC-DECMAX)*(OLDACC+DECMAX)/VT2
                    IF ( SLPDEC . GE . -0.3 )       RETURN
C-----BOUND THE ACC/DEC SLOPE FOR A LANE CHANGE
      SLPLCH = AMAX1(SLPDEC,-12.0)
      RETURN
C-----VEHICLE IS PAST THE END OF LANE ON SIDE OF INTEREST SO TAKE FORCED
C-----PATH FOR CURRENT LANE
 5010 CONTINUE
      LEGAL = 2
      ISET = 5
      CALL  PATHF    ( LTRUE,N1,N2 )
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9030 CONTINUE
      CALL  ABORTR   ( MSG903,27 )
      STOP  903
 9040 CONTINUE
      CALL  ABORTR   ( MSG904,27 )
      STOP  904
 9050 CONTINUE
      CALL  ABORTR   ( MSG905,64 )
      STOP  905
      END                                                               LCHDES
```

```
      SUBROUTINE CHKLSI ( LANSI,LOK )
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / VLCHDE(17),LB,LE,LGEOM1,LGEOM2,LGEOM3,LGEOM4,
     *                  VSVEHU(5),VDELAY(14),VCKLAL(5),VGAPAC(28),
     *                  VCHGML(17),ZTEMPD(18)
      DATA     N1,N2  / 4HCHKL,2HSI /
C
C-----SUBROUTINE CHKLSI CHECKS THE LANE ON THE SIDE OF INTEREST TO SEE
C-----IF THE LANE IS AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE
C-----AND CLEAR TO THE INTERSECTION (LOK=0=OK, LOK=1=NOT AVAILABLE YET,
C-----AND LOK=2=PAST END)
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----CHECK THE LANE ON THE SIDE OF INTEREST TO SEE IF THE LANE IS
C-----AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE
      LOK = 2
C   COLEASE,FIND,LGEOM3,LANE,LANSI,LGEOM(3)
      CALL  FIND    (LGEOM3,      3,LANSI ,     19)                 COLEASE
C   COLEASE,FIND,LGEOM4,LANE,LANSI,LGEOM(4)
      CALL  FIND    (LGEOM4,      3,LANSI ,     20)                 COLEASE
C-----IF THE LANE IS ONLY AVAILABLE AT THE FIRST THEN RETURN (DO NOT
C-----ALLOW A VEHICLE TO CHANGE LANES INTO A LANE THAT IS NOT
C-----AVAILABLE AT THE INTERSECTION)
                    IF ( LGEOM3 . EQ . LGEOM4 )  RETURN
C   COLEASE,FIND,LGEOM1,LANE,LANSI,LGEOM(1)
      CALL  FIND    (LGEOM1,      3,LANSI ,     17)                 COLEASE
C   COLEASE,FIND,LGEOM2,LANE,LANSI,LGEOM(2)
      CALL  FIND    (LGEOM2,      3,LANSI ,     18)                 COLEASE
C-----SET THE BEGINNING AND THE ENDING OF THE LANE FOR A CONTINUOUS LANE
      LB = LGEOM1
      LE = LGEOM4
C-----IF THE LANE IS CONTINUOUS THAN GO TO 1010 AND CHECK THE POSITION
C-----OF THE VEHICLE
                    IF ( LGEOM2 . EQ . LGEOM4 )  GO TO 1010
C-----SET THE BEGINNING AND THE ENDING OF THE LANE FOR A LANE AVAILABLE
C-----AT THE LAST (DO NOT ALLOW A LANE CHANGE INTO THE FIRST PART OF A
C-----LANE BLOCKED IN THE MIDDLE ONLY) AND CHECK THE POSITION OF THE
C-----VEHICLE
      LB = LGEOM3
      LE = LGEOM4
 1010 CONTINUE
C-----IF THE POSITION OF THE VEHICLE IS LT THE BEGINNING OF THE LANE OR
C-----GT THE ENDING OF THE LANE THEN RETURN WITH THE FLAG SET FOR
C-----BLOCKED LANE ELSE RETURN WITH THE FLAG SET FOR LANE NOT BLOCKED
                    IF ( POSNEW.GT.FLOAT(LE) )   RETURN
      LOK = 1
                    IF ( POSNEW.LT.FLOAT(LB) )   RETURN
      LOK = 0
      RETURN
      END                                                           CHKLSI
```

```
      SUBROUTINE SVEHU  (NOQ)                                       COLEASE
C   TASK,SVEHU,NOQ
      COMMON / LANE   / LWID      ,NLL      ,NLR      ,ISNA     ,   COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL    ,ILVL     ,   COLEASE
     *                  LCONTR    ,LTURN    ,LGEOM ( 4),NLDL    ,   COLEASE
     *                  LLDL  ( 5),IBLN     ,IDUMLA                 COLEASE
      COMMON / VEHD   / ISLP      ,IACC     ,IVEL     ,IPOS     ,   COLEASE
     *                  ISET      ,LCHGE    ,ISPDP    ,LEGAL    ,   COLEASE
     *                  IPRTM     ,ITIMV    ,IQDS     ,ISPDS    ,   COLEASE
     *                  ISDS      ,IDVS     ,ISTCON   ,IVMAXA   ,   COLEASE
     *                  IVMAXD    ,LATPOS   ,IDTS     ,LALT     ,   COLEASE
     *                  NORC      ,LOGFLG   ,MSTPF    ,MLAG     ,   COLEASE
     *                  MTCARS    ,MFINL    ,MSFLG    ,MPOBS    ,   COLEASE
     *                  MOASF     ,MSAOR    ,MPRO     ,MBLOCK   ,   COLEASE
     *                  MININT    ,IFVA     ,IACDS    ,ICDFS    ,   COLEASE
     *                  ISDEC     ,ISTMO    ,IACLDS   ,IRSTOP       COLEASE
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPIC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VLCHDE(17),VCHKLS(6),IPOSF,IPOSR,LANSI,LGEOM4,
     *                  MEGAL,VDELAY(14),VCKLAL(5),VGAPAC(28),
     *                  VCHGML(17),ZTEMPD(18)
      DATA     N1,N2  / 4HSVEH,2HU  /
C
C-----SUBROUTINE SVEHU FINDS THE NEAREST VEHICLE TO THE FRONT AND THE
C-----NEAREST VEHICLE TO THE REAR IN THE LANE ON THE SIDE OF INTEREST
C-----FOR THIS VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE ENTRY NUMBER FOR THE LANE ENTITY OF THE LANE ON THE SIDE
C-----OF INTEREST BASED ON ISIDE
      LANSI = NLL
                    IF ( ISIDE . EQ . 3 )         LANSI = NLR
C-----INITIALIZE SOME PARAMETERS FOR SVEHU
      NOQ = 0
      NOSF = 0
      NOSR = 0
      IPOSF = LGEOM(4)*25.0 + 0.5
      IPOSR = 0
      LEADSP = IVEL
      LAGSPD = IVEL
                    IF ( LANSI . GT . 0 )         GO TO 1010
C-----THERE IS NO LANE ALTERNATIVE ON THE SIDE OF INTEREST THUS RETURN
      ISIDE = 2
      ISET = 5
      GO TO 2010
 1010 CONTINUE
C-----SET THE POSITION OF THE NEAREST VEHICLE TO THE FRONT TO THE END OF
C-----THE LANE ON THE SIDE OF INTEREST
C   COLEASE,FIND,LGEOM4,LANE,LANSI,LGEOM(4)
      CALL  FIND    (LGEOM4,      3,LANSI ,     20)                 COLEASE
      IPOSF = LGEOM4*25.0 + 0.5
C-----SET NOSF TO THE FIRST VEHICLE IN THE LANE ON THE SIDE OF INTEREST
C   COLEASE,FIND,NOSF,LANE,LANSI,IFVL
      CALL  FIND    (NOSF  ,      3,LANSI ,     13)                 COLEASE
                    IF ( NOSF . EQ . 0 )         GO TO 2010
C-----FIND THE POSITION AND SPEED OF THE FIRST VEHICLE IN THE LANE ON
C-----THE SIDE OF INTEREST
C   COLEASE,FIND,IPOSF,VEHD,NOSF,IPOS
      CALL  FIND    (IPOSF ,      6,NOSF  ,      4)                 COLEASE
C   COLEASE,FIND,LEADSP,VEHD,NOSF,IVEL
      CALL  FIND    (LEADSP,      6,NOSF  ,      3)                 COLEASE
                    IF ( ISIDE . NE . 1 )         GO TO 1020
C-----THE FIRST VEHICLE IN THE LANE ON THE SIDE OF INTEREST IS TO THE
C-----LEFT AND HAS BEEN UPDATED THIS DT THUS UN-UPDATE HIM
```

```
      IPOSF = IPOSF - LEADSP*DT + 0.5
 1020 CONTINUE
                    IF ( LEADSP . GT . 0 )        GO TO 1030
C   COLEASE,FIND,MEGAL,VEHD,NOSF,LEGAL
      CALL  FIND    (MEGAL ,      6,NOSF  ,      8)                 COLEASE
                    IF ( MEGAL . EQ . 2 )        GO TO 1030
                    IF ( MEGAL . GT . 3 )        GO TO 1030
C-----THE FIRST VEHICLE IN THE LANE ON THE SIDE OF INTEREST IS STOPPED
C-----AND HE MUST CHANGE LANES THUS SET NOQ TO BE THE NUMBER OF 20 FOOT
C-----VEHICLES THAT WOULD OCCUPY THE DISTANCE FROM THE FIRST VEHICLE IN
C-----THE LANE ON THE SIDE OF INTEREST TO THE END OF THAT LANE
      NOQ = (LGEOM4-IPOSF)/20
 1030 CONTINUE
C-----IF THE POSITION OF THE FIRST VEHICLE IN THE LANE ON THE SIDE OF
C-----INTEREST IS GT THE POSITION OF THIS VEHICLE THEN GO TO 1040 AND
C-----CHECK THE NEXT VEHICLE IN THE LANE ON THE SIDE OF INTEREST ELSE
C-----SET THE NEAREST VEHICLE TO THE FRONT TO NO VEHICLE AND SET THE
C-----NEAREST VEHICLE TO THE REAR TO THE FIRST VEHICLE IN THE LANE ON
C-----THE SIDE OF INTEREST
                    IF ( IPOSF . GT . IPOS )      GO TO 1040
      NOSR = NOSF
      NOSF = 0
      IPOSR = IPOSF
      IPOSF = LGEOM4*25.0 + 0.5
      LAGSPD = LEADSP
      LEADSP = IVEL
      NOQ = 0
      GO TO 2010
 1040 CONTINUE
C-----INCREMENT THE NUMBER OF VEHICLES IN THE LANE ON THE SIDE OF
C-----INTEREST AHEAD OF THIS VEHICLE
      NOQ = NOQ + 1
C-----SET THE NEAREST VEHICLE TO THE REAR IN THE LANE ON THE SIDE OF
C-----INTEREST TO THE NOR FOR THE NOSF VEHICLE
C   COLEASE,FIND,NOSR,VEHF,NOSF,NOR
      CALL  FIND    (NOSR  ,      7,NOSF  ,      5)                 COLEASE
C-----IF THERE IS NO VEHICLE BEHIND THE NOSF VEHICLE THEN GO TO 2010 AND
C-----SET THE POSITIONS ELSE FIND THE POSITION AND SPEED OF THE NOSR
C-----VEHICLE IN THE LANE ON THE SIDE OF INTEREST
                    IF ( NOSR . EQ . 0 )         GO TO 2010
C   COLEASE,FIND,IPOSR,VEHD,NOSR,IPOS
      CALL  FIND    (IPOSR ,      6,NOSR  ,      4)                 COLEASE
C   COLEASE,FIND,LAGSPD,VEHD,NOSR,IVEL
      CALL  FIND    (LAGSPD,      6,NOSR  ,      3)                 COLEASE
                    IF ( ISIDE . NE . 1 )        GO TO 1050
C-----THE NOSR VEHICLE IN THE LANE ON THE SIDE OF INTEREST IS TO THE
C-----LEFT AND HAS BEEN UPDATED THIS DT SO UN-UPDATE HIM
      IPOSR = IPOSR - LAGSPD*DT + 0.5
 1050 CONTINUE
                    IF ( LAGSPD . GT . 0 )        GO TO 1060
C   COLEASE,FIND,MEGAL,VEHD,NOSR,LEGAL
      CALL  FIND    (MEGAL ,      6,NOSR  ,      8)                 COLEASE
                    IF ( MEGAL . EQ . 2 )        GO TO 1060
                    IF ( MEGAL . GT . 3 )        GO TO 1060
C-----THE NOSR VEHICLE IN THE LANE ON THE SIDE OF INTEREST IS STOPPED
C-----AND HE MUST CHANGE LANES THUS SET NOQ TO BE THE NUMBER OF 20 FOOT
C-----VEHICLES THAT WOULD OCCUPY THE DISTANCE FROM THE NOSR VEHICLE IN
C-----THE LANE ON THE SIDE OF INTEREST TO THE END OF THAT LANE
      NOQ = (LGEOM4-IPOSR)/20
 1060 CONTINUE
C-----IF THE POSITION OF THE NOSR VEHICLE IN THE LANE ON THE SIDE OF
C-----INTEREST IS LE THE POSITION OF THIS VEHICLE THEN GO TO 2010 AND
C-----SET THE POSITIONS ELSE SET THE NEW NOSF VEHICLE TO THE NOSR
C-----VEHICLE AND SET THE NEW NOSR VEHICLE TO NO VEHICLE AND CHECK AGAIN
                    IF ( IPOSR . LE . IPOS )      GO TO 2010
      NOSF = NOSR
      NOSR = 0
      IPOSF = IPOSR
      IPOSR = 0
      LEADSP = LAGSPD
      LAGSPD = IVEL
      GO TO 1040
 2010 CONTINUE
C-----SET THE POSITIONS OF THE NOSF AND THE NOSR VEHICLE AND RETURN
      PVSF = IPOSF/25.0
      PVSR = IPOSR/25.0
      RETURN
      END                                                           SVEHU
```

```
      SUBROUTINE DELAY                                                  COLEASE
C     TASK,DELAY
      COMMON / LANE   / LWID      ,NLL      ,NLR      ,ISNA     ,       COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL    ,ILVL     ,       COLEASE
     *                  LCONTR    ,LTURN    ,LGEOM ( 4),NLDL    ,       COLEASE
     *                  LLDL  ( 5),IBLN     ,IDUMLA                     COLEASE
      COMMON / VEHD   / ISLP      ,IACC     ,IVEL     ,IPOS     ,       COLEASE
     *                  ISET      ,LCHGE    ,ISPDP    ,LEGAL    ,       COLEASE
     *                  IPRTM     ,ITIMV    ,IQDS     ,ISPDS    ,       COLEASE
     *                  ISDS      ,IDVS     ,ISTCON   ,IVMAXA   ,       COLEASE
     *                  IVMAXD    ,LATPOS   ,IDTS     ,LALT     ,       COLEASE
     *                  NORC      ,LOGFLG   ,MSTPF    ,MLAG     ,       COLEASE
     *                  MTCARS    ,MFINL    ,MSFLG    ,MPOBS    ,       COLEASE
     *                  MOASF     ,MSAOR    ,MPRO     ,MBLOCK   ,       COLEASE
     *                  MININT    ,IFVA     ,IACDS    ,ICDFS    ,       COLEASE
     *                  ISDEC     ,ISTMO    ,IACLDS   ,IRSTOP           COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL   ,ISPD     ,NOF      ,       COLEASE
     *                  NOR       ,LNEXT    ,LPRES    ,ITURN    ,       COLEASE
     *                  IBAPS     ,IPRTLO   ,IEXTIM   ,NOBAPD           COLEASE
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / VLCHDE(17),VCHKLS(6),VSVEHU(5),JLCH,JTURN,LAGR,
     *                  LANSI,LEADR,LOK,NOQ,NORF,NORR,PVRF,PVRR,QUEL,
     *                  QUER,QUES,VCKLAL(5),VGAPAC(28),VCHGML(17),
     *                  ZTEMPD(18)
      DIMENSION         IPENTC(3,3)
C-----DATA      IPENTC / LL SL RL   LS SS RS   LR SR RR /  ME=NOF
      DATA      IPENTC /  1, 4, 4,   0, 0, 0,   2, 2, 1 /
      DATA      N1,N2  / 4HDELA,2HY  /
C
C-----SUBROUTINE DELAY FINDS THE LEGAL LANE FOR THE VEHICLE WITH THE
C-----MINIMUM EXPECTED DELAY
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                  IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      JTURN = 2
                  IF ( NOF . EQ . 0 )          GO TO 1010
C     COLEASE,FIND,JTURN,VEHF,NOF,ITURN
      CALL  FIND     (JTURN ,      7,NOF    ,       8)                  COLEASE
                  IF ( JTURN . EQ . 0 )        JTURN = 2
 1010 CONTINUE
C-----FIND THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE
C-----STRAIGHT AHEAD BASED ON THE TURN CODE OF THIS VEHICLE AND THE TURN
C-----CODE FOR THE NOF VEHICLE
      QUES = IVN-1 + IPENTC(ITURN,JTURN)*DCHAR(IDRICL)
C     COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
      CALL  FIND     (JLCH  ,      4,LNEXT  ,      72)                  COLEASE
C-----IF THE VEHICLES INTERSECTION PATH CHANGES LANES WITHIN THE
C-----INTERSECTION THEN INCREASE THE EQUIVALENT NUMBER OF VEHICLES IN
C-----THE QUEUE IN THE LANE STRAIGHT AHEAD
                  IF ( JLCH . NE . 0 )         QUES = QUES + 10.0
C-----INITIALIZE THE VALUES FOR THE EQUIVALENT NUMBER OF VEHICLES IN THE
C-----LANE TO THE LEFT AND THE LANE TO THE RIGHT
      QUER =  1000.0
      QUEL =  1000.0
 1020 CONTINUE
C-----PROCESS BY THE LANE ALTERNATIVE
      GO TO ( 2010,4010,5010,4010,3010,6020 ) , LALT
 2010 CONTINUE
C-----THERE ARE NO LANE ALTERNATIVES THUS RETURN AND DO NOT CHECK THE
C-----DESIRABILITY OF A LANE CHANGE ANY MORE
      ISET = 6
      GO TO 6020
 3010 CONTINUE
C-----CHECK THE LANE ALTERNATIVES FOR THIS LANE
      CALL  CKLALT
      GO TO 1020
 4010 CONTINUE
C-----FIND THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE TO
C-----THE RIGHT
      ISIDE = 3
      JTURN = 2
      LANSI = NLR
C-----CHECK THE LANE ON THE SIDE OF INTEREST TO SEE IF THE LANE IS
C-----AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE AND CLEAR TO THE
C-----INTERSECTION
      CALL  CHKLSI   ( LANSI,LOK )
C-----IF THE LANE IS NOT AVAILABLE FOR THIS VEHICLE THEN GO TO 5010 AND
C-----CHECK THE LANE ON THE LEFT
                  IF ( LOK . NE . 0 )          GO TO 5010
C-----FIND THE NEAREST VEHICLE TO THE FRONT AND THE NEAREST VEHICLE TO
C-----THE REAR IN THE LANE ON THE SIDE OF INTEREST FOR THIS VEHICLE
      CALL  SVEHU    ( NOQ )
C-----SAVE THE VEHICLE PARAMETERS FOR THE LANE TO THE RIGHT
      NORF = NOSF
      NORR = NOSR
      PVRF = PVSF
      PVRR = PVSR
      LEADR = LEADSP
      LAGR = LAGSPD
                  IF ( NOSF . EQ . 0 )         GO TO 4020
C-----FIND THE LEAD VEHICLES TURN CODE
C     COLEASE,FIND,JTURN,VEHF,NOSF,ITURN
      CALL  FIND     (JTURN ,      7,NOSF   ,       8)                  COLEASE
C-----IF THE LEAD VEHICLES TURN CODE EQ 0 THEN SET FOR STRAIGHT
                  IF ( JTURN . EQ . 0 )        JTURN = 2
 4020 CONTINUE
C-----COMPUTE THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE
C-----TO THE RIGHT BASED ON THE TURN CODE OF THE VEHICLE AND THE TURN
C-----CODE OF THE LEAD VEHICLE ON THE RIGHT
      QUER = NOQ+1 + IPENTC(ITURN,JTURN)*DCHAR(IDRICL)
 5010 CONTINUE
C-----IF THE LANE TO THE LEFT IS NOT AN ALTERNATIVE FOR THIS LANE THEN
C-----GO TO 6010 AND DETERMINE WHICH LANE HAS THE MINIMUM EXPECTED DELAY
                  IF ( LALT . EQ . 2 )         GO TO 6010
      ISIDE = 1
      JTURN = 2
      LANSI = NLL
C-----CHECK THE LANE ON THE SIDE OF INTEREST TO SEE IF THE LANE IS
C-----AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE AND CLEAR TO THE
C-----INTERSECTION
      CALL  CHKLSI   ( LANSI,LOK )
C-----IF THE LANE TO THE LEFT IS NOT AVAILABLE FOR THE VEHICLE THEN GO
C-----TO 6010 AND DETERMINE WHICH LANE HAS THE MINIMUM EXPECTED DELAY
                  IF ( LOK . NE . 0 )          GO TO 6010
C-----FIND THE NEAREST VEHICLE TO THE FRONT AND THE NEAREST VEHICLE TO
C-----THE REAR IN THE LANE ON THE SIDE OF INTEREST FOR THIS VEHICLE
      CALL  SVEHU    ( NOQ )
                  IF ( NOSF . EQ . 0 )         GO TO 5020
C-----FIND THE LEAD VEHICLES TURN CODE
C     COLEASE,FIND,JTURN,VEHF,NOSF,ITURN
      CALL  FIND     (JTURN ,      7,NOSF   ,       8)                  COLEASE
C-----IF THE LEAD VEHICLES TURN CODE EQ 0 THEN SET FOR STRAIGHT
                  IF ( JTURN . EQ . 0 )        JTURN = 2
 5020 CONTINUE
      QUEL = NOQ+1 + IPENTC(ITURN,JTURN)*DCHAR(IDRICL)
C-----COMPUTE THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE
C-----TO THE LEFT BASED ON THE TURN CODE OF THE VEHICLE AND THE TURN
C-----CODE OF THE LEAD VEHICLE ON THE LEFT
 6010 CONTINUE
C-----IF THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THIS LANE IS
C-----LE THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE TO
C-----THE LEFT AND IN THE LANE TO THE RIGHT THEN GO TO 6020 AND SET NO
C-----LANE CHANGE DESIRABLE
      IF ( QUES.LE.QUER.AND.QUES.LE.QUEL )       GO TO 6020
```

```
C-----LESS DELAY CAN BE EXPECTED IF THIS VEHICLE WOULD CHANGE LANES THUS
C-----IF THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN THE LANE TO
C-----THE LEFT IS LE THE EQUIVALENT NUMBER OF VEHICLES IN THE QUEUE IN
C-----THE LANE TO THE RIGHT THEN RETURN WITH THE POSITION AND INDEX OF
C-----THE LEAD AND LAG VEHICLES IN THE LEFT LANE SET AND TRY TO CHANGE
C-----LANES ELSE SET THE POSITION AND THE INDEX OF THE LEAD AND LAG
C-----VEHICLES FOR THE RIGHT LANE AND TRY TO CHANGE LANES
                      IF ( QUEL . LE . QUER )       RETURN
      ISIDE = 3
      NOSF = NORF
      NOSR = NORR
      PVSF = PVRF
      PVSR = PVRR
      LEADSP = LEADR
      LAGSPD = LAGR
      RETURN
 6020 CONTINUE
C-----SET NO LANE CHANGE DESIRABLE FLAG AND RETURN
      ISIDE = 2
      NOSF = 0
      NOSR = 0
      RETURN
      END                                                               DELAY
```

```
      SUBROUTINE CKLALT                                                 COLEASE
C     TASK,CKLALT
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / VLCHDE(17),VCHKLS(6),VSVEHU(5),VDELAY(14),I,
     *                  IPATH,JLCH,MOBAP,MPINT,VGAPAC(28),VCHGML(17),
     *                  ZTEMPD(18)
      DATA     N1,N2  / 4HCKLA,2HLT /
C
C-----SUBROUTINE CKLALT CHECKS THE LANE ALTERNATIVES FOR THIS LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE THE LANE ALTERNATIVES FOR NO LANE ALTERNATIVE
      LALT = 1
C-----IF THERE IS NO LANE TO THE RIGHT THEN GO TO 2010 AND CHECK THE
C-----LANE TO THE LEFT
                      IF ( NLR . EQ . 0 )            GO TO 2010
C     COLEASE,FIND,MPINT,LANE,NLR,NPINT
      CALL  FIND     (MPINT ,      3,NLR    ,       5)                  COLEASE
C-----IF THERE ARE NO PATHS INTO THE INTERSECTION FROM THE LANE TO THE
C-----RIGHT THEN GO TO 2010 AND CHECK THE LANE TO THE LEFT
                      IF ( MPINT . EQ . 0 )          GO TO 2010
C-----CHECK EACH INTERSECTION PATH FROM THE LANE TO THE RIGHT TO SEE IF
C-----IT GOES TO THE VEHICLES DESIRED OUTBOUND APPROACH
      DO 1010  I = 1 , MPINT
C     COLEASE,FIND,IPATH,LANE,NLR,LINTP(I)
      CALL  FIND     (IPATH ,      3,NLR    ,       5+I      )          COLEASE
C     COLEASE,FIND,JLCH,PATH,IPATH,ILCH
      CALL  FIND     (JLCH  ,      4,IPATH  ,      72)                  COLEASE
C-----IF THE INTERSECTION PATH BEING CHECKED CHANGES LANES WITHIN THE
C-----INTERSECTION THEN GO TO 1010 AND SKIP TO THE NEXT INTERSECTION
C-----PATH
                      IF ( JLCH . NE . 0 )           GO TO 1010
C     COLEASE,FIND,MOBAP,PATH,IPATH,LOBAP
      CALL  FIND     (MOBAP ,      4,IPATH  ,      71)                  COLEASE
C-----IF THE LINKING OUTBOUND APPROACH FOR THE INTERSECTION PATH IS EQ
C-----TO THE DESIRED OUTBOUND APPROACH FOR THIS VEHICLE THEN GO TO 1020
C-----AND SET THE LANE TO THE RIGHT AS A LANE ALTERNATIVE
                      IF ( MOBAP . EQ . NOBAPD )    GO TO 1020
 1010 CONTINUE
C-----NONE OF THE INTERSECTION PATHS FROM THE LANE TO THE RIGHT GOES TO
C-----THE VEHICLES DESIRED OUTBOUND APPROACH THUS GO TO 2010 AND CHECK
C-----THE LANE TO THE LEFT
      GO TO 2010
 1020 CONTINUE
C-----SET THE LANE TO THE RIGHT AS A LANE ALTERNATIVE
      LALT = LALT + 1
 2010 CONTINUE
C-----IF THERE IS NO LANE TO THE LEFT THEN RETURN
                      IF ( NLL . EQ . 0 )            RETURN
C     COLEASE,FIND,MPINT,LANE,NLL,NPINT
      CALL  FIND     (MPINT ,      3,NLL    ,       5)                  COLEASE
```

```
C-----IF THERE ARE NO PATHS INTO THE INTERSECTION FROM THE LANE TO THE
C-----LEFT THEN RETURN
                    IF ( MPINT . EQ . 0 )         RETURN
C-----CHECK EACH INTERSECTION PATH FROM THE LANE TO THE LEFT TO SEE IF
C-----IT GOES TO THE VEHICLES DESIRED OUTBOUND APPROACH
      DO 2020  I = 1 , MPINT
C     COLEASE,FIND,IPATH,LANE,NLL,LINTP(I)
      CALL  FIND    (IPATH ,      3,NLL    ,      5+I      )                    COLEASE
C     COLEASE,FIND,JLCH,PATH,IPATH,ILCH
      CALL  FIND    (JLCH  ,      4,IPATH  ,      72)                           COLEASE
C-----IF THE INTERSECTION PATH BEING CHECKED CHANGES LANES WITHIN THE
C-----INTERSECTION THEN GO TO 2020 AND SKIP TO THE NEXT INTERSECTION
C-----PATH
                    IF ( JLCH . NE . 0 )          GO TO 2020
C     COLEASE,FIND,MOBAP,PATH,IPATH,LOBAP
      CALL  FIND    (MOBAP ,      4,IPATH ,       71)                           COLEASE
C-----IF THE LINKING OUTBOUND APPROACH FOR THE INTERSECTION PATH IS EQ
C-----TO THE DESIRED OUTBOUND APPROACH FOR THIS VEHICLE THEN GO TO 2030
C-----AND SET THE LANE TO THE LEFT AS A LANE ALTERNATIVE
                    IF ( MOBAP . EQ . NOBAPD )   GO TO 2030
 2020 CONTINUE
C-----NONE OF THE INTERSECTION PATHS FROM THE LANE TO THE LEFT GOES TO
C-----THE VEHICLES DESIRED OUTBOUND APPROACH THUS RETURN
      RETURN
 2030 CONTINUE
C-----SET THE LANE TO THE LEFT AS A LANE ALTERNATIVE
      LALT = LALT + 2
      RETURN
      END                                                                       CKLALT
```

```
      SUBROUTINE GAPACC ( LANS1 )                                               COLEASE
C   TASK,GAPACC,LANSI
      COMMON / LOGICV / LTRUE,LFALSE                                            COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,            COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,            COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,            COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,            COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,            COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,            COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,            COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,            COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,            COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP                 COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,            COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,            COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD                 COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VLCHDE(17),VCHKLS(6),VSVEHU(5),VDELAY(14),
     *                  VCKLAL(5),ACCVEH,ALAGAP,ALEGAP,CRISLP,DECMAX,
     *                  DENOM,FACT,GAPLA,GAPLE,JACC,JBLN,JSET,JSISET,
     *                  JVEHCL,LEGAP,MCONTR,OLDACC,RADICL,RELDIS,RESPLA,
     *                  RESPLE,SLOPE,SLPDEC,T,T1,VSQT4,VT2,X,
     *                  VCHGML(17),ZTEMPD(18)
      DATA      GAPMIN / 8.0 /
      DATA      N1,N2  / 4HGAPA,2HCC /
C4701 FORMAT(52H  RESPLE  ALEGAP   GAPLE  RESPLA  ALAGAP   GAPLA    ,
C4   *        52HISET    NOSF    PVSF    VVSF    AVSF    NOSR    PVSR,
C4   *        16H    VVSR    AVSR,/,6F8.2,2I8,3F8.2,I8,3F8.2)
C4702 FORMAT(4H T =F7.2,4H X =F7.2,8H GAPLE =F7.2)
C4703 FORMAT(4H T =F7.2,4H X =F7.2,8H GAPLA =F7.2)
C4704 FORMAT(4H T =F7.2,4H X =F7.2,8H GAPLE =F7.2,
C4   *        21H FOR ACCEL AND ISET=3)
C
C-----SUBROUTINE GAPACC CHECKS IF THERE IS AN ACCEPTABLE GAP TO LANE
C-----CHANGE INTO AND IF NOT THEN DETERMINE THE APPROPRIATE DRIVER
C-----RESPONSE FOR LANE CHANGING
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR CHECKING FOR A GAP
      FACT = FACTOR*DCHAR(IDRICL)*VCHAR(IVEHCL)
      CRISLP = 4.0*DCHAR(IDRICL)
      VVSF = LEADSP/25.0
      VVSR = LAGSPD/25.0
      AVSF = 0.0
      AVSR = 0.0
                    IF ( NOSF . NE . 0 )         GO TO 1020
C-----IF THERE IS A LEAD VEHICLE ON LANE ON THE SIDE OF INTEREST THEN GO
C-----TO 1020 AND FIND HIS CURRENT ACC/POS
                    IF ( LEGAL . EQ . 2 )        GO TO 1030
C-----IF TURN IS LEGAL FROM CURRENT LANE GO TO 1030
```

```
                  IF ( IA . EQ . NOBAPD )       GO TO 1030
C-----IF VEHICLE ON OUTBOUND APPROACH GO TO 1030
C   COLEASE,FIND,MCONTR,LANE,LANSI,LCONTR
      CALL  FIND    (MCONTR,      3,LANSI ,     15)                  COLEASE
C-----IF INTERSECTION IS CONTROLLED BUT LANE IS UNCONTROLLED GO TO 1030
              IF ( ICONTR.NE.1.AND.MCONTR.EQ.2 )   GO TO 1030
C-----IF LANE IS NOT SIGNAL CONTROLLED GO TO 1010
                  IF ( MCONTR . LT . 5 )        GO TO 1010
C   COLEASE,FIND,JBLN,LANE,LANSI,IBLN
      CALL  FIND    (JBLN ,       3,LANSI ,     27)                  COLEASE
      JSISET = ISISET(ICAMPC,JBLN)
C-----IF LANE SIGNAL CONTROL IS GREEN GO TO 1030
                  IF ( JSISET . EQ . 01 )       GO TO 1030
                  IF ( JSISET . EQ . 04 )       GO TO 1030
C-----IF CHANGING LEFT AND SIGNAL IS PROTECTED LEFT GO TO 1030
              IF ( ISIDE.EQ.1.AND.JSISET.GE.23 )   GO TO 1030
C-----OTHERWISE GO TO 1010
                  IF ( ISIDE . EQ . 1 )         GO TO 1010
C-----VEHICLE CHANGING RIGHT AND CHECKS FOR RIGHT TURN SIGNAL GREEN
                  IF ( JSISET . EQ . 07 )       GO TO 1030
                  IF ( JSISET . EQ . 09 )       GO TO 1030
                  IF ( JSISET . EQ . 13 )       GO TO 1030
                  IF ( JSISET . EQ . 15 )       GO TO 1030
                  IF ( JSISET . EQ . 17 )       GO TO 1030
                  IF ( JSISET . EQ . 18 )       GO TO 1030
                  IF ( JSISET . EQ . 23 )       GO TO 1030
 1010 CONTINUE
C-----SET UP MINIMUM ACCEPTABLE LEAD VEHICLE PARAMETERS FOR LANE CHANGE
      LEADSP = 125
      VVSF = 5.0
      GO TO 1030
 1020 CONTINUE
C-----FIND THE LEAD VEHICLES ACC/DEC
C   COLEASE,FIND,JACC,VEHD,NOSF,IACC
      CALL  FIND    (JACC  ,      6,NOSF  ,      2)                  COLEASE
      AVSF = JACC/312.5 - 32.0
C-----FIND THE LEAD VEHICLES REAR BUMPER POSITION
C   COLEASE,FIND,JVEHCL,VEHF,NOSF,IVEHCL
      CALL  FIND    (JVEHCL,      7,NOSF  ,      2)                  COLEASE
      PVSF = PVSF - LENV(JVEHCL) - 4.0
                  IF ( ISIDE . NE . 1 )         GO TO 1030
C-----THE LEAD VEHICLE IS TO THE LEFT AND HAS BEEN UPDATED THIS DT THUS
C-----UN-UPDATE THE VELOCITY
      VVSF = AMAX1((VVSF-AVSF*DT),0.0)
 1030 CONTINUE
                  IF ( NOSR . EQ . 0 )          GO TO 1040
C-----FIND THE LAG VEHICLES ACC/DEC
C   COLEASE,FIND,JACC,VEHD,NOSR,IACC
      CALL  FIND    (JACC  ,      6,NOSR  ,      2)                  COLEASE
      AVSR = JACC/312.5 - 32.0
                  IF ( ISIDE . NE . 1 )         GO TO 1040
C-----THE LAG VEHICLE IS TO THE LEFT AND HAS BEEN UPDATED THIS DT THUS
C-----UN-UPDATE THE VELOCITY
      VVSR = AMAX1((VVSR-AVSR*DT),0.0)
 1040 CONTINUE
                  IF ( VVSR . LT . 5.0 )        GO TO 4020
C-----FIND THE ACCEPTABLE LEAD GAP AND THE ACTUAL LEAD GAP
      RESPLE = VELOLD - VVSF
      ALEGAP = (2.0+0.7*VELOLD+(ABS(RESPLE)*RESPLE*0.05))/FACT
      ALEGAP = AMAX1(ALEGAP,GAPMIN/DCHAR(IDRICL))
      GAPLE = PVSF - POSOLD
      LEGAP = LFALSE
C-----IF THE ACTUAL LEAD GAP IS GE THE ACCEPTABLE LEAD GAP THEN SET THE
C-----LEAD GAP OK FLAG
                  IF ( GAPLE . GE . ALEGAP )    LEGAP = LTRUE
C-----IF THE LEAD GAP IS NOT OK AND THE LEAD VEHICLE IS ALMOST STOPPED
C-----THEN GO TO 4020 AND REJECT THE GAP
      IF ( LEGAP.EQ.LFALSE . AND . VVSF.LT.5.0 ) GO TO 4020
C-----FIND THE ACCEPTABLE LAG GAP AND THE ACTUAL LAG GAP
      RESPLA = VVSR - VELOLD
      ALAGAP = (4.0+1.4*VELOLD+(ABS(RESPLA)*RESPLA*0.10))/FACT
```

```
      ALAGAP = AMAX1(ALAGAP,GAPMIN/DCHAR(IDRICL))
      GAPLA = POSOLD - LENV(IVEHCL) - 4.0 - PVSR
C5                IF ( IPRTLO . EQ . 0 )        GO TO 101
C4                IF ( TIME . LT . TPRINT )     GO TO 101
C4    PRINT 701 , RESPLE,ALEGAP,GAPLE,RESPLA,ALAGAP,GAPLA,ISET,NOSF,
C4   *            PVSF,VVSF,AVSF,NOSR,PVSR,VVSR,AVSR
C4101 CONTINUE
C-----IF THE ACTUAL LAG GAP IS LT THE ACCEPTABLE LAG GAP THEN GO TO 4010
C-----AND CHECK THE LEAD GAP
                  IF ( GAPLA . LT . ALAGAP )    GO TO 4010
C-----IF THE LEAD GAP IS NOT OK WHEN THE LAG GAP IS OK THEN GO TO 3010
C-----AND REJECT THE GAP
                  IF ( LEGAP . EQ . LFALSE )    GO TO 3010
C-----BOTH THE LEAD GAP AND THE LAG GAP ARE OK THUS CHECK TO SEE THAT
C-----THERE WILL NOT BE A COLLISION IF THIS VEHICLE CHANGES LANES
                  IF ( RESPLE . LE . 0.0 )      GO TO 2010
C-----FIND THE RELATIVE DISTANCE REQUIRED FOR THIS VEHICLE TO DECELERATE
C-----TO THE LEAD VEHICLES SPEED
                  IF ( AVSF . EQ . 0.0 )        AVSF = 1.0E-20
      SLOPE = -0.75*CRISLP
      T = (-ACCOLD-SQRT(ACCOLD**2-2.0*SLOPE*RESPLE))/SLOPE
      T1 = -VVSF/AVSF
                  IF ( T1 . LT . 0.0 )          T1 = T
      T1 = AMIN1(T1,T)
      X = VELOLD*T+0.5*ACCOLD*T**2+SLOPE*T**3/6.0-VVSF*T1-0.5*AVSF*T1**2
C5                IF ( IPRTLO . EQ . 0 )        GO TO 102
C4                IF ( TIME . LT . TPRINT )     GO TO 102
C4    PRINT 702 , T,X,GAPLE
C4102 CONTINUE
C-----IF THE ACTUAL LEAD GAP IS LT THE RELATIVE DISTANCE REQUIRED FOR
C-----THIS VEHICLE TO DECELERATE TO THE LEAD VEHICLES SPEED THEN GO TO
C-----3010 AND REJECT THE GAP
                  IF ( GAPLE . LT . X )         GO TO 3010
 2010 CONTINUE
                  IF ( RESPLA . LE . 0.0 )      GO TO 2020
C-----FIND THE RELATIVE DISTANCE REQUIRED FOR THE LAG VEHICLE TO
C-----DECELERATE TO THIS VEHICLES SPEED
                  IF ( ACCOLD . EQ . 0.0 )      ACCOLD = 1.0E-20
      SLOPE = -0.75*CRISLP
      T = (-AVSR-SQRT(AVSR**2-2.0*SLOPE*RESPLA))/SLOPE
      T1 = -VELOLD/ACCOLD
                  IF ( T1 . LT . 0.0 )          T1 = T
      T1 = AMIN1(T1,T)
      X = VVSR*T+0.5*AVSR*T**2+SLOPE*T**3/6.0-VELOLD*T1-0.5*ACCOLD*T1**2
C5                IF ( IPRTLO . EQ . 0 )        GO TO 103
C4                IF ( TIME . LT . TPRINT )     GO TO 103
C4    PRINT 703 , T,X,GAPLA
C4103 CONTINUE
C-----IF THE ACTUAL LAG GAP IS LT THE RELATIVE DISTANCE REQUIRED FOR THE
C-----LAG VEHICLE TO DECELERATE TO THIS VEHICLES SPEED THEN GO TO 5010
C-----AND CHECK TO SEE IF THIS VEHICLE CAN ACCELERATE FOR THE GAP
                  IF ( GAPLA . LT . X )         GO TO 5010
 2020 CONTINUE
C-----EVERYTHING SEEMS TO BE OK SO INITIATE THE LANE CHANGE
      ISET = 1
      RETURN
 3010 CONTINUE
C-----THE LAG GAP IS OK BUT THE LEAD GAP IS NOT OK THUS IF THE VEHICLE
C-----HAS BEEN ACCELERATING FOR THE GAP THEN GO TO 2020 AND INITIATE THE
C-----LANE CHANGE
                  IF ( ISET . EQ . 3 )          GO TO 2020
 3020 CONTINUE
C-----CALCULATE THE LANE CHANGE ACC/DEC SLOPE TO REDUCE THE VEHICLES
C-----VELOCITY TO 85 PERCENT OF THE LEAD VEHICLES SPEED IN ONE DT
      SLPLCH = (0.85*VVSF-(VELOLD+ACCOLD*DT))/(0.5*DTSQ)
C-----BOUND THE LANE CHANGE ACC/DEC SLOPE
      SLPLCH = AMIN1(AMAX1(SLPLCH,-CRISLP),CRISLP)
C-----IF THE LANE CHANGE IS FORCED THEN GO TO 4030 AND STOP IN HALF THE
C-----REMAINING DISTANCE TO THE END OF THE LANE
                  IF ( LEGAL . EQ . 1 )         GO TO 4030
                  IF ( LEGAL . EQ . 3 )         GO TO 4030
```

```
C-----REJECT THE GAP AND CAR-FOLLOW THE LEAD VEHICLE
      ISET = 2
      RETURN
 4010 CONTINUE
C-----THE LAG GAP IS NOT OK THUS IF THE LEAD GAP IS OK THEN GO TO 5010
C-----AND CHECK TO SEE IF THIS VEHICLE CAN ACCELERATE FOR THE GAP
                    IF ( LEGAP . EQ . LTRUE )    GO TO 5010
 4020 CONTINUE
C-----NEITHER THE LEAD GAP NOR THE LAG GAP IS OK THUS IF THE VEHICLE HAS
C-----BEEN ACCELERATING FOR THE GAP THEN INITIATE THE LANE CHANGE
                    IF ( ISET . EQ . 3 )         GO TO 2020
C-----IF THE LANE CHANGE IS FORCED THEN GO TO 3020 AND CAR-FOLLOW THE
C-----LEAD VEHICLE
                    IF ( LEGAL . EQ . 1 )        GO TO 3020
                    IF ( LEGAL . EQ . 3 )        GO TO 3020
C-----REJECT THE GAP AND CONTINUE NORMALLY
      ISET = 5
      RETURN
 4030 CONTINUE
C-----REJECT THE GAP AND CALCULATE THE LANE CHANGE ACC/DEC SLOPE
C-----REQUIRED TO STOP THE VEHICLE IN HALF THE REMAINING DISTANCE TO THE
C-----END OF THE LANE
      ISET = 4
      RELDIS = (ENDLN-POSOLD)/2.0
      DENOM = 6.0*RELDIS
      VT2 = 2.0*VELOLD
      VSQT4 = VT2*VT2
      OLDACC = AMIN1(ACCOLD,0.0)
      RADICL = VSQT4 + DENOM*OLDACC
                    IF ( RADICL . LE . 0.0 )     RETURN
      DECMAX = -OLDACC - (VSQT4+VT2*SQRT(RADICL))/DENOM
      SLPDEC = (OLDACC-DECMAX)*(OLDACC+DECMAX)/VT2
                    IF ( SLPOLD . GE . 0.0 )     GO TO 4050
C-----IF THE VEHICLES ACC/DEC IS LE -9 THEN SET THE LANE CHANGE ACC/DEC
C-----SLOPE TO 60 PERCENT OF THE OLD ACC/DEC SLOPE
                    IF ( ACCOLD . LE . -9.0 )    SLPDEC = 0.6*SLPOLD
 4050 CONTINUE
C-----BOUND THE LANE CHANGE ACC/DEC SLOPE.
      SLPLCH = AMIN1(AMAX1(SLPDEC,-12.0),SLPLCH,-0.01)
      RETURN
 5010 CONTINUE
C-----THE LEAD GAP IS OK BUT THE LAG GAP IS NOT OK THUS CHECK IF THE
C-----VEHICLE CAN ACCELERATE TO CHANGE AHEAD OF THE LAG VEHICLE
C-----IF THE ACTUAL LAG GAP IS LT 0 THEN DO NOT ACCELERATE FOR THE GAP
                    IF ( GAPLA . LT . 0.0 )      GO TO 4020
C-----IF THE VEHICLE IS STOPPING THEN DO NOT ACCELERATE FOR THE GAP
                    IF ( ICDFS . NE . LFALSE )   GO TO 4020
C-----IF THE LAG VEHICLE SPEED IS GT 6 FPS MORE THAN THIS VEHICLES SPEED
C-----THEN DO NOT ACCELERATE FOR THE GAP
                    IF ( RESPLA . GT . 6.0 )     GO TO 4020
C-----IF THIS VEHICLES ACC/DEC IS LT -CRISLP THEN DO NOT ACCELERATE FOR
C-----THE GAP
                    IF ( ACCOLD . LT . -CRISLP ) GO TO 4020
                    IF ( NOSR . EQ . 0 )         GO TO 5020
C   COLEASE,FIND,JSET,VEHD,NOSR,ISET
      CALL  FIND    (JSET  ,      6,NOSR  ,      5)                      COLEASE
C-----IF THE LAG VEHICLE IS ACCELERATING FOR A GAP THEN DO NOT
C-----ACCELERATE FOR THE GAP
                    IF ( JSET . EQ . 3 )         GO TO 4020
 5020 CONTINUE
C-----IF THE ACTUAL LEAD GAP PLUS THE ACTUAL LAG GAP IS LT 1.2 TIMES THE
C-----ACCEPTABLE LEAD GAP PLUS THE ACCEPTABLE LAG GAP THEN DO NOT
C-----ACCELERATE FOR THE GAP
      IF ( GAPLE+GAPLA.LT.1.2*(ALEGAP+ALAGAP) )  GO TO 4020
C-----IF THE DISTANCE TO THE PREVIOUS VEHICLE IN THIS LANE IS LT THE
C-----DISTANCE THAT MUST BE MADE UP IN THE LAG GAP THEN DO NOT
C-----ACCELERATE FOR THE GAP
            IF ( RELPOS.LT.ALEGAP+ALAGAP-GAPLA ) GO TO 4020
                    IF ( RESPLE . LE . 6.0 )     GO TO 5030
C-----CALCULATE THE RELATIVE DISTANCE REQUIRED FOR THIS VEHICLE TO
C-----DECELERATE TO THE LEAD VEHICLE SPEED
                    IF ( AVSF . EQ . 0.0 )       AVSF = 1.0E-20
      SLOPE = -0.75*CRISLP
      T = (-ACCOLD-SQRT(ACCOLD**2-2.0*SLOPE*RESPLE))/SLOPE
      T1 = -VVSF/AVSF
                    IF ( T1 . LT . 0.0 )         T1 = T
      T1 = AMIN1(T1,T)
      X = VELOLD*T+0.5*ACCOLD*T**2+SLOPE*T**3/6.0-VVSF*T1-0.5*AVSF*T1**2
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 104
C4                  IF ( TIME . LT . TPRINT )    GO TO 104
C4    PRINT 704 , T,X,GAPLE
C4104 CONTINUE
C-----IF THE ACTUAL LEAD GAP IS LT THE RELATIVE DISTANCE REQUIRED FOR
C-----THIS VEHICLE TO DECELERATE TO THE LEAD VEHICLES SPEED THEN DO NOT
C-----ACCELERATE FOR THE GAP
                    IF ( GAPLE . LT . X )        GO TO 4020
 5030 CONTINUE
C-----CALCULATE THE LANE CHANGE ACC/DEC SLOPE REQUIRED TO ACCELERATE THE
C-----THE VEHICLE AT 75 PERCENT OF THE MAXIMUM ACCELERATION FOR THE
C-----VEHICLE AT THE CURRENT VELOCITY
      ISET = 3
      ACCVEH = 0.75*DCHAR(IDRICL)*AMAX(IVEHCL)*(1.0-VELOLD/VMAX(IVEHCL))
      SLPLCH = AMIN1((ACCVEH-ACCOLD)/DT,CRISLP)
                    IF ( NOSR . EQ . 0 )         RETURN
C-----FLAG THE NOSR VEHICLE TO DECELERATE TO FOLLOW A LANE CHANGING
C-----VEHICLE
C   COLEASE,STORE,LTRUE,VEHD,NOSR,MLAG
      CALL  STORE   (LTRUE ,      6,NOSR  ,     24)                      COLEASE
      RETURN
      END                                                                GAPACC
```

```
      SUBROUTINE CHGMLN                                                 COLEASE
C     TASK,CHGMLN
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLOL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VLCHDE(17),VCHKLS(6),VSVEHU(5),VDELAY(14),
     *                  VCKLAL(5),VGAPAC(28),DECMAX,I,JBLN,JGO,JLN,JSET,
     *                  JVEL,LGEOM2,LGEOM4,LTF,MCONTR,MEGAL,MWID,NOASF,
     *                  NVILL,POSLAT,XCRIT,ZTEMPD(18)
      DIMENSION         IENT6(1),IENT7(1)
      EQUIVALENCE       (IFVA,IENT6(1)),(MINFLZ,IENT7(1))
      DATA    F3      / 1.33333333333333 /
      DATA    N1,N2   / 4HCHGM,2HLN /
C
C-----SUBROUTINE CHGMLN LOGS THE VEHICLE OUT OF HIS PRESENT LANE AND
C-----INTO THE NEW LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE LANE CHANGE FLAG
      LCHGE = 2
C-----RESET SOME OF THE VEHICLES PARAMETERS
      PVPOS = PVSF
      PVVEL = VVSF
      PVACC = AVSF
      IREPFX = LTRUE
      MSFLG = LFALSE
      LALT = 5
      MBLOCK = LFALSE
      IPRTM = 0
      JPRTM = 0
                    IF ( NOSF . EQ . 0 )          GO TO 1020
```

```
C-----THERE IS A LEAD VEHICLE SO UPDATE THE PREVIOUS VEHICLE PARAMETERS
      PVPOS = PVPOS + PVVEL*DT
      PVVEL = AMAX1((PVVEL+PVACC*DT),0.0)
C-----RESET ALL THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES TO LFALSE
      DO 1010  I = 1 , 7
      IENT6(I) = LFALSE
 1010 CONTINUE
                    IF ( PVVEL . LE . 0.01 )      GO TO 1030
C-----THE LEAD VEHICLE IS MOVING SO SET THE VEHICLE TO CAR-FOLLOW HIM
      IFVA = LTRUE
      DESVEL = AMIN1(DESVEL,0.95*PVVEL)
      GO TO 1040
 1020 CONTINUE
C-----THERE IS NO LEAD VEHICLE AND IF THIS VEHICLE IS NOT CONTINUING A
C-----DECELERATION FOR A STOP THEN GO TO 1040 AND CONTINUE
                    IF ( ICDFS . EQ . LFALSE )    GO TO 1040
      ICDFS = LFALSE
 1030 CONTINUE
C-----SET THIS VEHICLE TO CHECK CRITICAL STOPPING DISTANCE FOR A
C-----DECELERATION FOR A STOP
      ISDEC = LTRUE
 1040 CONTINUE
      RELPOS = PVPOS - POSNEW
      RELVEL = PVVEL - VELNEW
C-----DECREMENT THE NUMBER OF VEHICLES IN THE PRESENT LANE
      NVILL = NVIL(ILN) - 1
C     COLEASE,STORE,NVILL,APPRO,IA,NVIL(ILN)
      CALL  STORE   (NVILL ,      1,IA     ,      7+ILN    )            COLEASE
      NVIL(ILN) = NVILL
C-----LOG THE VEHICLE OUT OF THE PRESENT LANE
      LTF = LFALSE
                    IF ( NOF . NE . 0 )           GO TO 2010
C-----SET THE FIRST VEHICLE IN THE PRESENT LANE TO THIS VEHICLES OLD NOR
C-----(OLD NOF EQ 0)
      LTF = LTRUE
C     COLEASE,STORE,NOR,LANE,LPRES,IFVL
      CALL  STORE   (NOR   ,      3,LPRES  ,     13)                    COLEASE
      IFVL = NOR
      GO TO 2020
 2010 CONTINUE
C-----SET THE NOR FOR THE OLD NOF VEHICLE TO THIS VEHICLES OLD NOR
C-----(OLD NOF NE 0)
C     COLEASE,STORE,NOR,VEHF,NOF,NOR
      CALL  STORE   (NOR   ,      7,NOF    ,      5)                    COLEASE
 2020 CONTINUE
                    IF ( NOR . NE . 0 )           GO TO 2030
C-----SET THE LAST VEHICLE IN THE PRESENT LANE TO THIS VEHICLES OLD NOF
C-----(OLD NOR EQ 0)
C     COLEASE,STORE,NOF,LANE,LPRES,ILVL
      CALL  STORE   (NOF   ,      3,LPRES  ,     14)                    COLEASE
      ILVL = NOF
      GO TO 2040
 2030 CONTINUE
C-----SET MFINL AND MOASF TO LTF, RESET IACC TO SLIGHTLY DECELERATING
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE NOF FOR NOF FOR THE OLD NOR VEHICLE
C-----(OLD NOR NE 0)
      CALL  FLGNOR ( LTF,NOF )
                    IF ( NOF . EQ . 0 )           GO TO 2040
C-----SET THE CORRECT VALUE FOR MOASF FOR THE OLD NOR VEHICLE
C-----(OLD NOR NE 0 AND OLD NOF NE 0)
C     COLEASE,FIND,JVEL,VEHD,NOF,IVEL
      CALL  FIND    (JVEL  ,      6,NOF    ,      3)                    COLEASE
      NOASF = LFALSE
                    IF ( JVEL . LE . 0 )          NOASF = LTRUE
C     COLEASE,STORE,NOASF,VEHD,NOR,MOASF
      CALL  STORE   (NOASF ,      6,NOR    ,     29)                    COLEASE
 2040 CONTINUE
C-----LOG THE VEHICLE INTO THE NEW LANE
C-----SET THE VEHICLES NEW NOF AND NOR FOR THE NEW LANE
      NOF = NOSF
```

```
      IVPV = NOF
      NOR = NOSR
C-----UPDATE THE VEHICLES LANE INDEXES
      JLN = ILN + (ISIDE-2)
      LPRES = LPRES + (ISIDE-2)
C-----INCREMENT THE NUMBER OF VEHICLES IN THE NEW LANE
      NVILL = NVIL(JLN) + 1
C   COLEASE,STORE,NVILL,APPRO,IA,NVIL(JLN)
      CALL  STORE   (NVILL ,      1,IA     ,      7+JLN    )                  COLEASE
      NVIL(JLN) = NVILL
C-----IF THE VEHICLE IS CHANGING LANES TO THE RIGHT THEN SET THE FLAG
C-----FOR ALREADY PROCESSED IN THIS DT
                        IF ( ISIDE . EQ . 3 )        LALT = 6
                        IF ( NOF . NE . 0 )          GO TO 3010
C-----SET THIS VEHICLE AS THE NEW FIRST VEHICLE IN THE NEW LANE
C-----(NEW NOF EQ 0)
      MFINL = LTRUE
      MOASF = LTRUE
C   COLEASE,STORE,IV,LANE,LPRES,IFVL
      CALL  STORE   (IV     ,      3,LPRES ,      13)                         COLEASE
                        IF ( NOR . EQ . 0 )          GO TO 3020
C-----CHECK IF THE NEW NOR VEHICLES LANE CHANGING FLAG CAN BE TURNED
C-----BACK ON (NEW NOF EQ 0 AND NEW NOR NE 0)
C   COLEASE,FIND,JSET,VEHD,NOR,ISET
      CALL  FIND    (JSET  ,      6,NOR    ,       5)                         COLEASE
                        IF ( JSET . NE . 6 )         GO TO 3020
C   COLEASE,FIND,MEGAL,VEHD,NOR,LEGAL
      CALL  FIND    (MEGAL ,      6,NOR    ,       8)                         COLEASE
                        IF ( MEGAL . EQ . 4 )        GO TO 3020
C-----TURN THE NEW NOR VEHICLES LANE CHANGING FLAG BACK ON
C   COLEASE,STORE,5,VEHD,NOR,ISET
      CALL  STORE   (5      ,      6,NOR    ,       5)                         COLEASE
      GO TO 3020
 3010 CONTINUE
C-----SET THIS VEHICLE AS THE NEW NOR FOR THE NEW NOF VEHICLE AND FIND
C-----THE NEW VALUE FOR MOASF FOR THIS VEHICLE (NEW NOF NE 0)
      MFINL = LFALSE
C   COLEASE,STORE,IV,VEHF,NOF,NOR
      CALL  STORE   (IV     ,      7,NOF    ,       5)                         COLEASE
C   COLEASE,FIND,JVEL,VEHD,NOF,IVEL
      CALL  FIND    (JVEL  ,      6,NOF    ,       3)                         COLEASE
      MOASF = LFALSE
                        IF ( JVEL . LE . 0 )         MOASF = LTRUE
 3020 CONTINUE
                        IF ( NOR . NE . 0 )          GO TO 3030
C-----SET THIS VEHICLE AS THE NEW LAST VEHICLE IN THE NEW LANE
C-----(NEW NOR EQ 0)
C   COLEASE,STORE,IV,LANE,LPRES,ILVL
      CALL  STORE   (IV     ,      3,LPRES ,      14)                         COLEASE
      GO TO 3040
 3030 CONTINUE
C-----SET MFINL AND MOASF TO LFALSE, RESET IACC TO SLIGHTLY DECELERATING
C-----IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT DECELERATING, SET MSFLG
C-----TO LFALSE, AND FINALLY STORE IV FOR NOF FOR THE NEW NOR VEHICLE
C-----(NEW NOR NE 0)
      CALL  FLGNOR  ( LFALSE,IV )
C-----FLAG THE NEW NOR VEHICLE THAT HE IS FOLLOWING A LANE CHANGING
C-----VEHICLE
C   COLEASE,STORE,3,VEHD,NOR,LCHGE
      CALL  STORE   (3      ,      6,NOR    ,       6)                         COLEASE
 3040 CONTINUE
C   COLEASE,FIND,MWID,LANE,LPRES,LWID
      CALL  FIND    (MWID  ,      3,LPRES ,       1)                         COLEASE
C-----SET THE TOTAL LATERAL DISTANCE FOR THE LANE CHANGE (BIASED BY 2)
      LEGAL = MWID + LWID
C-----SET THE CURRENT LATERAL POSITION FOR THE LANE CHANGE TO THE TOTAL
C-----LATERAL DISTANCE FOR THE LANE CHANGE
C-----(A POSITIVE VALUE FOR POSLAT MEANS THE VEHICLE IS CHANGING LEFT)
C-----(A NEGATIVE VALUE FOR POSLAT MEANS THE VEHICLE IS CHANGING RIGHT)
      POSLAT = LEGAL/2.0
                        IF ( ISIDE . EQ . 3 )        POSLAT = -POSLAT
```

```
C-----BIAS THE CURRENT LATERAL POSITION FOR THE LANE CHANGE
      LATPOS = 8.0*(POSLAT+15.0) + 0.5
                        IF ( IA . EQ . NOBAPD )      RETURN
                        IF ( LNEXT . EQ . 0 )        GO TO 3050
C-----UNSET THE INTERSECTION CONFLICTS FOR THE INTERSECTION PATH FOR THE
C-----VEHICLE
      CALL  UNSETC
 3050 CONTINUE
C-----FIND AN INTERSECTION PATH FOR THIS VEHICLE BASED ON THE CURRENT
C-----APPROACH, THE NEW LANE, AND THE DESIRED OUTBOUND APPROACH
      CALL  PATHF   ( LFALSE,N1,N2 )
C-----THIS VEHICLE SHOULD CHECK TO SEE IF IT SHOULD BE WITHIN THE
C-----INFLUENCE ZONE OF THE INTERSECTION CONTROL THUS IF THE VEHICLE HAS
C-----NOT DEDICATED HIMSELF TO AN INTERSECTION PATH THEN RETURN AND WAIT
C-----UNTIL THE VEHICLE IS DEDICATED TO AN INTERSECTION PATH
                        IF ( LNEXT . EQ . 0 )        GO TO 4010
C   COLEASE,FIND,LGEOM2,LANE,LPRES,LGEOM(2)
      CALL  FIND    (LGEOM2,      3,LPRES ,      18)                          COLEASE
C   COLEASE,FIND,LGEOM4,LANE,LPRES,LGEOM(4)
      CALL  FIND    (LGEOM4,      3,LPRES ,      20)                          COLEASE
      ENDLN = LGEOM4
                        IF ( MBLOCK . EQ . LTRUE )   ENDLN = LGEOM2
      RELEND = ENDLN - POSOLD
                        IF ( MINFLZ . EQ . LTRUE )   GO TO 3060
C-----CALCULATE THE THRESHOLD DISTANCE FROM THE END OF THE LANE THAT THE
C-----VEHICLE SHOULD BECOME WITHIN THE INFLUENCE ZONE OF THE
C-----INTERSECTION CONTROL (LET 4+PIJR SECONDS AT THE CURRENT VELOCITY
C-----PLUS THE STOPPING DISTANCE BE THE THRESHOLD DISTANCE)
      DECMAX = DUTOL*(-6.0-VELNEW/44.0)*DCHAR(IDRICL)
      DECMAX = AMAX1(DECMAX,DMAX(IVEHCL))
      XCRIT = VELNEW*(4.0+PIJR(IDRICL)+F3*VELNEW/DECMAX)
C-----LET 400 FEET BE THE MINIMUM THRESHOLD DISTANCE
      XCRIT = AMAX1(XCRIT,400.0)
C-----IF THE DISTANCE FROM THE END OF THE END OF THE LANE IS GT THE
C-----THRESHOLD DISTANCE THEN RETURN AND WAIT UNTIL THE VEHICLE IS
C-----CLOSER
                        IF ( RELEND . GT . XCRIT )   GO TO 4010
 3060 CONTINUE
C-----THE VEHICLE WAS WITHIN THE INFLUENCE ZONE OF THE INTERSECTION
C-----CONTROL SO SET THE PARAMETERS NECESSARY TO CALL INFLZN FOR THE
C-----NEW LANE
      MCONTR = LCONTR
      JBLN = IBLN
      JGO = IGO
C   COLEASE,FIND,LCONTR,LANE,LPRES,LCONTR
      CALL  FIND    (LCONTR,      3,LPRES ,      15)                          COLEASE
C   COLEASE,FIND,IBLN,LANE,LPRES,IBLN
      CALL  FIND    (IBLN  ,      3,LPRES ,      27)                          COLEASE
      IGO = 1
      DO 3070  I = 1 , 10
      IENT7(I) = LFALSE
 3070 CONTINUE
C-----INITIALIZE THE VEHICLES INTERSECTION CONTROL LOGICAL ATTRIBUTES
C-----BASED ON THE TYPE OF TRAFFIC CONTROL FOR THE NEW LANE
      CALL  INFLZN
C-----RESET PARAMETERS FOR THE PRESENT LANE
      LCONTR = MCONTR
      IBLN = JBLN
      IGO = JGO
 4010 CONTINUE
C-----SET THE INTERSECTION CONTROL LOGIC TIMER SO THIS VEHICLE WILL BE
C-----PROCESSED NEXT DT
      LOGTMP = 2
      LOGFLG = 2
      RETURN
      END                                                                     CHGMLN
```

```
      SUBROUTINE ACDCP                                                  COLEASE
C     TASK,ACDCP
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / K,RADICL,T,VCARFO(20),VACCEL(12),VCRIDI(14),
     *                  VADLVA(6),VHOLDS(2),ZTEMPD(53)
      DIMENSION         MSG906(11),MSG907(11)
      DATA      MSG906 / 4H NO ,4HVEHD,4H DEP,4HENDE,4HNT A,4HTTRI,
     *                  4HBUTE,4H TRU,4HE = ,4HACDC,4HP    /
      DATA      MSG907 / 4H STO,4HPPED,4H VEH,4HICLE,4HS NO,4HT PR,
     *                  4HOGRA,4HMMED,4H YET,4H = A,4HCDCP /
      DATA      N1,N2  / 4HACDC,2HP  /
C3701 FORMAT(3HDM=F7.3)
C
C-----SUBROUTINE ACDCP CHECKS THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES,
C-----CALLS THE APPROPRIATE ACC/DEC ROUTINES, AND COMPUTES THE VEHICLES
C-----NEW POS/VEL/ACC
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----IF THE VEHICLE IS IN PIJR TIME THEN GO TO 7090 HOLD THE VEHICLES
C-----SPEED
                    IF ( JPRTM . GT . 0 )         GO TO 7090
      MSFLG = LFALSE
C-----IF THIS VEHICLE IS THE FIRST VEHICLE IN THIS LANE WHO DECIDED TO
C-----STOP AT THE STOP LINE FOR AN AMBER SIGNAL THEN GO TO 4020 AND
C-----CHECK CRITICAL STOPPING DISTANCE FOR A DECELERATION TO A STOP
                    IF ( IGO . EQ . 2 )           GO TO 4020
                    IF ( ICDFS . EQ . LFALSE )    GO TO 1010
C-----CONTINUE DECELERATION FOR A STOP
      MSFLG = LTRUE
C-----IF THE PREVIOUS VEHICLE IS NO LONGER STOPPED THEN SET THE FLAG TO
C-----DISCONTINUE DECELERATION FOR A STOP
                    IF ( PVVEL . GT . 0.0 )       MSFLG = LFALSE
C3    IPFLAG = 10HSTOPPING
      GO TO 6010
 1010 CONTINUE
                    IF ( IFVA . EQ . LFALSE )     GO TO 2010
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO FOLLOW THE VEHICLE AHEAD
      CALL  CARFOL
      GO TO 6010
 2010 CONTINUE
                    IF ( IACLDS . EQ . LFALSE )  GO TO 3010
C-----ACCELERATE ACCORDING TO THE LEAD VEHICLES SPEED
      DESVEL = PVVEL
      GO TO 3020
 3010 CONTINUE
                    IF ( IACDS . EQ . LFALSE )   GO TO 4010
 3020 CONTINUE
C-----ACCELERATE ACCORDING TO THE DESIRED SPEED FOR THIS VEHICLE
      CALL  ACCEL
      GO TO 6010
 4010 CONTINUE
C-----IF THE REMAIN STOPPED FLAG IS SET THEN GO TO 7080 AND REMAIN
C-----STOPPED
                    IF ( IRSTOP . NE . LFALSE )  GO TO 7080
                    IF ( ISDEC . EQ . LFALSE )   GO TO 5010
 4020 CONTINUE
                    IF ( VELOLD . LE . 0.0 )      GO TO 7080
C-----CHECK CRITICAL STOPPING DISTANCE FOR A DECELERATION TO A STOP AND
C-----IF VIOLATED THEN INITIATE A DECELERATION TO A STOP
      CALL  CRIDIS  ( K )
C-----IF THE VEHICLE DID NOT VIOLATE THE CRITICAL STOPPING DISTANCE FOR
C-----A DECELERATION TO A STOP THIS DT OR WITHIN PIJR TIME THEN GO TO
C-----3020 AND ACCELERATE ACCORDING TO THE DESIRED SPEED FOR THIS
C-----VEHICLE
                    IF ( K . EQ . 2 )             GO TO 3020
      GO TO 7010
 5010 CONTINUE
                    IF ( ISTMO . EQ . LFALSE )   GO TO 9060
C-----CHECK IF STOPPED BUS OR PARKED VEHICLE SHOULD START TO MOVE
      GO TO 9070
 6010 CONTINUE
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER DT SECONDS
C-----(POS/VEL/ACC IS ALSO COMPUTED IN CRIDIS IF K NE 2 BUT GOES TO 7010
C-----AFTERWARDS AND DOES NOT COME THROUGH THIS CODE)
      CALL  NEWVEL  ( DT,DTSQ,DTCU )
C-----IF THIS VEHICLE WAS PREVIOUSLY STOPPED AND THE NEW VELOCITY IS EQ
C-----ZERO THEN GO TO 7080 AND REMAIN STOPPED
      IF ( MSTPF.EQ.LTRUE.AND.VELNEW.EQ.0.0 )    GO TO 7080
 7010 CONTINUE
      MSTPF = LFALSE
C-----IF THIS VEHICLES VELOCITY IS GT 0 THEN RETURN
                    IF ( VELNEW . GT . 0.0 )      RETURN
C-----THE VEHICLE STOPPED THIS DT
      LOGTMP = 2
C-----CALCULATE THE TIME REQUIRED TO BRING THE VEHICLE TO A STOP WITHIN
C-----THIS DT
      VELOLD = AMAX1(VELOLD,0.01)
                    IF ( SLPNEW . EQ . 0.0 )      GO TO 7020
      RADICL = ACCOLD**2 - 2.0*SLPNEW*VELOLD
                    IF ( RADICL . LT . 0.0 )      GO TO 7020
      T = (-ACCOLD-SQRT(RADICL))/SLPNEW
      GO TO 7030
 7020 CONTINUE
                    IF ( ACCOLD . GE . 0.0 )      GO TO 7040
      T = VELOLD/(-ACCOLD)
 7030 CONTINUE
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER T SECONDS
C-----(THE VELOCITY SHOULD BE 0)
      CALL  NEWVEL  ( T,T**2,T**3 )
C3          ENCODE  ( 10,701,JPFLAG )                 ACCNEW
C-----UPDATE THE VEHICLES MAXIMUM DECELERATION RATE
      IVMAXD = MAX0(IVMAXD,IFIX(-ACCNEW*10.0+0.5))
 7040 CONTINUE
                    IF ( MININT . EQ . LTRUE )   GO TO 7080
```

```
                    IF ( RELPOS . GT . 10.0 )     GO TO 7080
      MATSTL = LFALSE
C-----IF THIS VEHICLE IS THE FIRST VEHICLE IN THE LANE AND THE LANE IS
C-----NOT BLOCKED THEN THE VEHICLE IS STOPPED AT THE STOP LINE
      IF ( MFINL.EQ.LTRUE.AND.MBLOCK.EQ.LFALSE ) MATSTL = LTRUE
      IREPIL = LTRUE
                    IF ( MATSTL . EQ . LFALSE )  GO TO 7080
C-----THE VEHICLE IS STOPPED AT THE STOP LINE ON AN INBOUND APPROACH SO
C-----ADD THE STOPPED VEHICLE TO THE LIST OF VEHICLES AT THE
C-----INTERSECTION
      CALL  ADLVAI
C-----CHECK IF LEFT-TURN-ON-RED OR RIGHT-TURN-ON-RED MAY BE MADE BASED
C-----ON THE LANE CONTROL FOR THIS LANE
                    IF ( LCONTR - 6 )              7080 , 7050 , 7060
 7050 CONTINUE
C-----LEFT-TURN-ON-RED PERMITTED FOR THIS LANE AND IF THIS VEHICLE IS
C-----NOT GOING TO TURN LEFT THEN GO TO 7080 ELSE SET LEFT-TURN-ON-RED
C-----FLAG
                    IF ( ITURN . NE . 1 )          GO TO 7080
      GO TO 7070
 7060 CONTINUE
C-----RIGHT-TURN-ON-RED PERMITTED FOR THIS LANE AND IF THIS VEHICLE IS
C-----NOT GOING TO TURN RIGHT THEN GO TO 7080 ELSE SET RIGHT-TURN-ON-RED
C-----FLAG
                    IF ( ITURN . NE . 3 )          GO TO 7080
 7070 CONTINUE
C-----SET THE LEFT-TURN-ON-RED OR RIGHT-TURN-ON-RED FLAG
      MLRTOR = LTRUE
      MTCARS = LFALSE
      LOGTMP = 2 + IPIJR(IDRICL)
C3    KPFLAG = 10HI MAY RTOR
 7080 CONTINUE
C-----THE VEHICLE IS STOPPED
C-----SET THE VEHICLES ACC/DEC LOGIC TIMER
      IPRTM = IPIJR(IDRICL)
C-----IF THE VEHICLE WAS TRYING NOT TO STOP THEN RESET THE VEHICLES
C-----ACC/DEC LOGIC TIMER TO ZERO
                    IF ( SLPNEW . GT . 0.0 )      IPRTM = 0
C-----RESET SOME OF THE VEHICLES PARAMETERS
      SLPNEW = 0.0
      ACCNEW = 0.0
      VELNEW = 0.0
      MSTPF = LTRUE
      MSFLG = LFALSE
      MSAOR = LFALSE
C3    IPFLAG = 10HMOVE UP
C-----IF THE VEHICLE IS STOPPED MORE THAN 10 FEET FROM THE PREVIOUS
C-----VEHICLE THEN MOVE UP ELSE REMAIN STOPPED
                    IF ( RELPOS . GT . 10.0 )     RETURN
      MSAOR = LTRUE
C3    IPFLAG = 10HSTOPPED
      IPRTM = 0
      RETURN
 7090 CONTINUE
C-----HOLD THE VEHICLES SPEED AT ITS CURRENT VALUE
      CALL  HOLDSP  ( JPRTM )
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9060 CONTINUE
      CALL  ABORTR  ( MSG906,41 )
      STOP  906
 9070 CONTINUE
      CALL  ABORTR  ( MSG907,44 )
      STOP  907
      END
```

ACDCP

```
      SUBROUTINE CARFOL                                                 COLEASE
C     TASK,CARFOL
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACDCP(3),A,ACC,ACCMAX,B,C,CARDEC,CARDIS,CRISLP,
     *                  DECVEH,DIST,FACT,LATNOW,LAT2GO,RADICL,SLOPE,
     *                  SLOPEU,SPD,T,T1,VT1,VACCEL(12),VCRIDI(14),
     *                  VADLVA(6),VHOLDS(2),ZTEMPD(53)
      DATA      N1,N2  / 4HCARF,2HOL /
C3701 FORMAT(3HRV=F7.2)
C3702 FORMAT(3HRP=F7.2)
C3703 FORMAT(3HCD=F7.2)
C
C-----SUBROUTINE CARFOL CALCULATES THE ACC/DEC SLOPE REQUIRED TO FOLLOW
C-----THE VEHICLE AHEAD
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR CARFOL
      DECVEH = DMAX(IVEHCL)
      CRISLP = 4.0*DCHAR(IDRICL)
C3    JPFLAG = 10HFOLLOWING
                    IF ( MLAG . EQ . LFALSE )    GO TO 1010
C-----A VEHICLE IS TRYING TO CHANGE LANES AHEAD OF THIS VEHICLE THUS SET
C-----THE LANE CHANGE ACC/DEC SLOPE TO 75 PERCENT OF THE DRIVERS
C-----CRITICAL SLOPE
      SLPLCH = -0.75*CRISLP
C-----IF THE DRIVERS ACC/DEC IS ALREADY LT HALF THE DRIVERS CRITICAL
C-----SLOPE THEN USE ONLY HALF OF THE LANE CHANGE ACC/DEC SLOPE
           IF ( ACCOLD . LT . -0.5*CRISLP )     SLPLCH = 0.5*SLPLCH
C3    JPFLAG = 10HFOLLOW LCG
      MLAG = LFALSE
 1010 CONTINUE
      PVVEL = AMAX1(PVVEL,0.01)
C-----IF THE PREVIOUS VEHICLE IS GOING FASTER THAN THIS VEHICLE THEN
C-----GO TO 4010 AND CHECK FURTHER
                    IF ( RELVEL . GE . 0.0 )     GO TO 4010
C-----IF THIS VEHICLE OR THE PREVIOUS VEHICLE IS CHANGING LANES THEN GO
C-----TO 3010 AND FACTOR THE RELATIVE POSITION OF THE VEHICLES
                    IF ( LCHGE . GT . 1 )        GO TO 3010
 2010 CONTINUE
C-----FIND THE CONSERVATIVE CAR FOLLOWING DISTANCE
      CARDIS = (1.7*PVVEL + 4.0*RELVEL**2)/DCHAR(IDRICL)
C-----IF THE VEHICLE IS FURTHER THAN CARDIS FROM THE PREVIOUS VEHICLE
```

```
C-----THEN GO TO 7010 AND CHECK FURTHER
                  IF ( RELPOS . GT . CARDIS )  GO TO 7010
      RELPOS = AMAX1(RELPOS,0.01)
C-----CALCULATE THE REQUIRED ACC/DEC USING THE NON-INTEGER GENERALIZED
C-----CAR FOLLOWING EQUATION
      CARDEC = CAREQA * ((VELOLD**CAREQM)/(RELPOS**CAREQL)) * RELVEL
C-----BOUND THE REQUIRED ACC/DEC
      CARDEC = AMIN1(AMAX1(CARDEC,DECVEH),-0.04/DT)
C-----CALCULATE THE REQUIRED ACC/DEC SLOPE TO BRING THE VEHICLES ACC/DEC
C-----TO CARDEC IN ONE DT
      SLPNEW = (CARDEC-ACCOLD)/DT
      FACT = -1.0
C-----IF THE VEHICLES ACC/DEC IS GT 0 OR IF THE RELATIVE POSITION IS LE
C-----40 FEET THEN ALLOW A NEGATIVE ACC/DEC SLOPE OF -1.3*CRISLP
                  IF ( ACCOLD . GT . 0.0 )      FACT = -1.3
      IF ( RELPOS . LE . 40.0/DCHAR(IDRICL) )    FACT = -1.3
C-----BOUND THE ACC/DEC SLOPE FOR CAR FOLLOWING
      SLPNEW = AMIN1(AMAX1(SLPNEW,FACT*CRISLP),CRISLP)
C3          ENCODE  ( 10,701,IPFLAG )              RELVEL
C3          ENCODE  ( 10,702,JPFLAG )              RELPOS
      GO TO 7030
 3010 CONTINUE
C-----THIS VEHICLE OR THE PREVIOUS VEHICLE IS CHANGING LANES THUS FACTOR
C-----THE RELATIVE POSITION
      LATNOW = LATPOS
      LAT2GO = LEGAL
C-----IF THIS VEHICLE IS CHANGING LANES THEN GO TO 3020
                  IF ( LCHGE . EQ . 2 )         GO TO 3020
C-----THE PREVIOUS VEHICLE IS CHANGING LANES
C   COLEASE,FIND,LATNOW,VEHD,NOF,LATPOS
      CALL  FIND    (LATNOW,      6,NOF    ,     18)                COLEASE
C   COLEASE,FIND,LAT2GO,VEHD,NOF,LEGAL
      CALL  FIND    (LAT2GO,      6,NOF    ,      8)                COLEASE
 3020 CONTINUE
C-----FACTOR THE RELATIVE POSITION OF THE VEHICLE BASED ON THE PERCENT
C-----OF THE LANE CHANGE COMPLETED AND CHECK AGAIN
C-----(FACTOR = 1.5 AT THE BEGINNING OF THE LANE CHANGE)
C-----(FACTOR = 1.0 AT THE END OF THE LANE CHANGE)
      RELPOS = RELPOS*(1.0+0.5*ABS(LATNOW/8.0-15.0)/(LAT2GO/2.0))
      GO TO 2010
 4010 CONTINUE
C-----THE PREVIOUS VEHICLE IS GOING FASTER THEN THIS VEHICLE SO RESET
C-----THE CAR FOLLOWING DISTANCE
      CARDIS = 1.7*PVVEL/DCHAR(IDRICL)
C-----IF THE RELATIVE POSITION OF THE VEHICLE IS LT 1.2 TIMES THE CAR
C-----FOLLOWING DISTANCE THEN GO TO 5010 AND CHECK FURTHER
            IF ( RELPOS . LT . 1.2*CARDIS )      GO TO 5010
 4020 CONTINUE
                  IF ( FLOAT(ISPD).LE.PVVEL )  GO TO 4030
C-----THE VEHICLES DESIRED SPEED IS GT THE PREVIOUS VEHICLES SPEED SO
C-----FACTOR THE VEHICLES DESIRED SPEED FOR ACCELERATION
C-----(FACT = 0 AND DESVEL = PVVEL  WHEN RELPOS = CARDIS)
C-----(FACT = 1 AND DESVEL = DESVEL WHEN RELPOS = 5*CARDIS)
      FACT = AMIN1(AMAX1((RELPOS-CARDIS)/(4.0*CARDIS),-0.04),1.0)
      DESVEL = PVVEL + (DESVEL-PVVEL)*FACT
 4030 CONTINUE
C-----ACCELERATE ACCORDING TO THE DESIRED SPEED FOR THIS VEHICLE
      CALL  ACCEL
      RETURN
 5010 CONTINUE
C-----THE VEHICLES RELATIVE POSITION IS LT 1.2*CARDIS SO RESET CARDIS
      CARDIS = 0.8*CARDIS
C-----IF THE VEHICLES RELATIVE POSITION IS BETWEEN 80 PERCENT AND 120
C-----PERCENT OF THE CARDIS FROM STATEMENT 4010 THEN GO TO 6010 AND
C-----ACCELERATE TO THE PREVIOUS VEHICLES SPEED
                  IF ( RELPOS . GT . CARDIS )  GO TO 6010
C-----IF THE VEHICLES OLD VELOCITY IS LE THE PREVIOUS VEHICLES VELOCITY
C-----THEN GO TO 4020 AND ACCELERATE TO THE FACTORED DESIRED SPEED
                  IF ( VELOLD . LE . PVVEL )    GO TO 4020
C-----FIND THE TIME AND VELOCITY WHEN THE VEHICLES ACCELERATION WOULD BE
C-----ZERO USING HALF THE CRITICAL SLOPE FOR THE DRIVER
      SLPNEW = 0.5*CRISLP
      T1 = -ACCOLD/SLPNEW
      VT1 = VELOLD + ACCOLD*T1 + 0.5*SLPNEW*T1**2
      SPD = AMIN1(FLOAT(ISPD),PVVEL)
C-----FIND THE ACCELERATION THE VEHICLE WOULD USE TO GET TO HIS DESIRED
C-----SPEED
      ACCMAX = AUTOL*(3.2+0.08*SPD)*DCHAR(IDRICL)
      ACC = ACCMAX*(1.0-VT1/(1.15*SPD))
                  IF ( ACC . LE . 0.0 )         GO TO 5020
C-----FIND THE TIME AND RELATIVE DISTANCE TRAVELED WHILE BRINGING THE
C-----VELOCITY BACK UP TO THE DESIRED SPEED
      T = T1 + ACC/SLPNEW + 0.5*DT
      DIST = VELOLD*T + 0.5*ACCOLD*T**2 + SLPNEW*T**3/6.0 - PVVEL*T
C-----IF THE NEW RELATIVE DISTANCE WOULD BE GE THE CAR FOLLOWING
C-----DISTANCE THEN START ACCELERATING AT HALF CRITICAL SLOPE
            IF ( RELPOS-DIST . GE . CARDIS )     GO TO 5030
 5020 CONTINUE
C-----SET THE ACC/DEC SLOPE TO MOVE THE VEHICLE BACK AWAY FROM THE
C-----PREVIOUS VEHICLE
      SLPNEW = 0.10*DECVEH*DCHAR(IDRICL)*(CARDIS-RELPOS)/CARDIS
 5030 CONTINUE
C-----BOUND THE ACC/DEC SLOPE WHEN THE VEHICLE IS LT 0.8*CARDIS AND
C-----CHECK FOR DECELERATION TO THE DESIRED SPEED
      SLPNEW = AMAX1(SLPNEW,-CRISLP)
C3          ENCODE  ( 10,703,IPFLAG )              CARDIS
C3          ENCODE  ( 10,702,JPFLAG )              RELPOS
      GO TO 7030
 6010 CONTINUE
C-----THE VEHICLES RELATIVE POSITION IS BETWEEN 80 AND 120 PERCENT OF
C-----CARDIS SO ACCELERATE TO THE MINIMUM OF THE DESIRED SPEED AND THE
C-----PREVIOUS VEHICLES VELOCITY
C3    JPFLAG = 10HCARDIS
      DESVEL = AMIN1(DESVEL,PVVEL)
      GO TO 4030
 7010 CONTINUE
C-----THE PREVIOUS VEHICLE IS GOING SLOWER THAN THIS VEHICLE BUT IF HIS
C-----RELATIVE POSITION IS GT 120 PERCENT OF CARDIS THEN ACCELERATE
                  IF ( RELPOS.GT.1.2*CARDIS )  GO TO 4020
C-----IF THE VEHICLES ACC/DEC IS VERY SMALL THEN GO TO 7020 AND SET
C-----THE VEHICLES ACC/DEC AND HIS ACC/DEC SLOPE TO ZERO
                  IF ( ABS(ACCOLD).LE.0.01 )   GO TO 7020
C-----FIND THE ACC/DEC SLOPE TO BRING THE VEHICLES ACC/DEC TO ZERO IN
C-----PIJR TIME
      SLPNEW = -1.01*ACCOLD/PIJR(IDRICL)
C-----IF THE VEHICLES ACC/DEC SLOPE OLD IS GT THE VEHICLES ACC/DEC SLOPE
C-----NEW AND THE SLOPES ARE THE SAME SIGN THEN USE THE VEHICLES OLD
C-----ACC/DEC SLOPE
      IF ( ABS(SLPOLD).GT.ABS(SLPNEW).AND.SLPOLD*SLPNEW.GT.0.0 )
     *                                        SLPNEW = SLPOLD
      SLPNEW = AMIN1(AMAX1(SLPNEW,-CRISLP),CRISLP)
      ACCNEW = ACCOLD + SLPNEW*DT
C-----IF THE ACC/DEC CHANGES SIGNS IN ONE DT THEN SET THE ACC/DEC SLOPE
C-----TO MAKE THE VEHICLES ACC/DEC ZERO IN ONE DT
                  IF ( ACCOLD*ACCNEW.LT.0.0 )  SLPNEW = -ACCOLD/DT
C3    IPFLAG = 10HREDUCE A/D
C3    JPFLAG = 10HTO 0 CARFL
      GO TO 7030
 7020 CONTINUE
C-----SET THE VEHICLES ACC/DEC AND ACC/DEC SLOPE TO ZERO
      ACCOLD = 0.0
      SLPNEW = 0.0
C3    IPFLAG = 10HSTEADY
C3    JPFLAG = 10HCARDIS
 7030 CONTINUE
C-----IF THE VEHICLES OLD VELOCITY IS LE HIS DESIRED SPEED THEN RETURN
C-----ELSE CHECK TO SEE IF THIS VEHICLE SHOULD BEGIN TO DECELERATE TO
C-----HIS DESIRED SPEED BY THE TIME HE REACHES THE END OF HIS LANE
                  IF ( VELOLD . LE . DESVEL )  RETURN
      SLOPE = -0.25*CRISLP
                  IF ( ACCOLD . LT . SLOPE )    SLOPE = 0.5*SLOPE
                  IF ( ACCOLD . EQ . 0.0 )      ACCOLD = 1.0E-6
```

```
      A = ACCOLD/6.0
      B = (2.0*VELOLD+DESVEL)/3.0
      C = POSOLD - AMIN1(PVPOS,ENDLN-DESVEL)
      RADICL = B**2 - 4.0*A*C
                    IF ( RADICL . LE . 0.0 )      GO TO 7040
      T = (-B+SQRT(RADICL))/(2.0*A)
                    IF ( T . LE . 0.0 )           GO TO 7040
C-----FIND THE ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES VELOCITY
C-----TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE AND
C-----BOUND THE ACC/DEC SLOPE
      SLOPE = AMIN1(SLOPE,2.0*(DESVEL-VELOLD-ACCOLD*T)/T**2)
 7040 CONTINUE
                    IF ( ACCOLD . GE . 0.0 )      GO TO 7050
C-----FIND THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC TO
C-----ZERO BY THE TIME THE VEHICLE REACHES HIS DESIRED SPEED
      SLOPEU = -0.5*ACCOLD**2/(DESVEL-VELOLD)
                    IF ( SLOPEU.LT.0.40*CRISLP ) GO TO 7050
C-----THE VEHICLE SHOULD START BRINGING THE ACC/DEC TO ZERO THUS BOUND
C-----THE ACC/DEC SLOPE FOR DECELERATING TO THE VEHICLES DESIRED SPEED
      SLOPE = AMIN1(SLOPEU,CRISLP)
 7050 CONTINUE
C-----BOUND THE ACC/DEC SLOPE FOR DECELERATING TO THE VEHICLES DESIRED
C-----SPEED
      SLOPE = AMAX1(SLOPE,-CRISLP)
                    IF ( SLOPE . GT . SLPNEW )    RETURN
C-----SET THE ACC/DEC SLOPE FOR DECELERATING TO THE VEHICLES DESIRED
C-----SPEED
C3    KPFLAG = 10HDEC DESPD
      SLPNEW = SLOPE
      RETURN
      END                                                         CARFOL
```

```
      SUBROUTINE ACCEL                                                    COLEASE
C   TASK,ACCEL
      COMMON / LOGICV / LTRUE,LFALSE                                      COLEASE
      COMMON / VEHD   / ISLP     ,IACC     ,IVEL     ,IPOS     ,         COLEASE
     *                  ISET     ,LCHGE    ,ISPDP    ,LEGAL    ,         COLEASE
     *                  IPRTM    ,ITIMV    ,IQDS     ,ISPDS    ,         COLEASE
     *                  ISDS     ,IDVS     ,ISTCON   ,IVMAXA   ,         COLEASE
     *                  IVMAXD   ,LATPOS   ,IDTS     ,LALT     ,         COLEASE
     *                  NORC     ,LOGFLG   ,MSTPF    ,MLAG     ,         COLEASE
     *                  MTCARS   ,MFINL    ,MSFLG    ,MPOBS    ,         COLEASE
     *                  MOASF    ,MSAOR    ,MPRO     ,MBLOCK   ,         COLEASE
     *                  MININT   ,IFVA     ,IACDS    ,ICDFS    ,         COLEASE
     *                  ISDEC    ,ISTMO    ,IACLDS   ,IRSTOP             COLEASE
      COMMON / VEHF   / IDRICL   ,IVEHCL   ,ISPD     ,NOF      ,         COLEASE
     *                  NOR      ,LNEXT    ,LPRES    ,ITURN    ,         COLEASE
     *                  ISAPS    ,IPRTLO   ,IEXTIM   ,NOBAPD             COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),OCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACOCP(3),VCARFO(20),A,ACC,ACCMAX,ACCVEH,B,C,
     *                  CRISLP,RADICL,RELPN,SLOPE,T,VT,VCRIDI(14),
     *                  VADLVA(6),VMOLDS(2),ZTEMPD(53)
      DATA      N1,N2  / 4HACCE,2HL  /
C3701 FORMAT(3HAC=F7.3)
C
C-----SUBROUTINE ACCEL ACCELERATES ACCORDING TO THE DESIRED SPEED FOR
C-----THIS VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR ACCEL
C3    IPFLAG = 10HSTEADY SPD
      CRISLP = 4.0*DCHAR(IDRICL)
                    IF ( DESVEL . LT . 0.5 )      DESVEL = 0.0
C-----IF THE VEHICLES OLD VELOCITY IS LT HIS DESIRED SPEED THEN GO TO
C-----1010 AND CHECK FOR ACCELERATION TO THE VEHICLES DESIRED SPEED
                IF ( VELOLD . LE . DESVEL-0.5*DT )   GO TO 1010
C-----IF THE VEHICLES OLD VELOCITY IS GT HIS DESIRED SPEED THEN GO TO
C-----2010 AND CHECK FOR DECELERATION TO THE VEHICLES DESIRED SPEED
                IF ( VELOLD . GT . DESVEL+1.0*DT )   GO TO 2010
C-----THE VEHICLES VELOCITY IS VERY NEAR THE VEHICLES DESIRED SPEED THUS
C-----IF THE VEHICLES ACC/DEC IS GT A VALUE THAT COULD BE REDUCED TO
C-----ZERO IN ONE DT THEN GO TO 4010 AND REDUCE THE VEHICLES ACC/DEC TO
C-----ZERO
                IF ( ABS(ACCOLD) . GT . CRISLP*DT )  GO TO 4010
C-----SET THIS VEHICLE AT HIS DESIRED SPEED WITH ACC/DEC AND ACC/DEC
C-----SLOPE OF ZERO
      SLPNEW = 0.0
      ACCOLD = 0.0
      VELOLD = DESVEL
      RETURN
 1010 CONTINUE
C-----ACCELERATE THE VEHICLE TO HIS DESIRED SPEED
C-----CALCULATE THE MAXIMUM ACCELERATION THE DRIVER WOULD USE TO GET TO
C-----HIS DESIRED SPEED IN THE LINEAR ACCELERATION MODEL
      ACCMAX = AUTOL*(3.2+0.08*DESVEL)*DCHAR(IDRICL)
C-----CALCULATE THE MAXIMUM ACCELERATION OF THE VEHICLE AT THE CURRENT
C-----VELOCITY USING THE NON-UNIFORM THEORY OF ACCELERATION
      ACCVEH = AMAX(IVEHCL)*(1.0-VELOLD/VMAX(IVEHCL))
C-----CALCULATE THE PORTION OF THE MAXIMUM ACCELERATION THAT THE DRIVER
C-----WOULD USE TO GET TO HIS DESIRED SPEED FROM HIS CURRENT VELOCITY
```

```
      ACC = AMIN1(ACCMAX,ACCVEH)*(1.0-VELOLD/(1.15*DESVEL))
C-----IF THIS VEHICLE MAY PROCEED INTO THE INTERSECTION AND IS THE FIRST
C-----VEHICLE IN HIS LANE THEN GO TO 1020 AND ACCELERATE TO ACC
      IF ( MPRO.EQ.LTRUE . AND . MFINL.EQ.LTRUE )GO TO 1020
C-----FIND THE NEW RELATIVE POSITION OF THE VEHICLE AFTER DT SECONDS IF
C-----THE ACCELERATION WAS INITIATED TO ACC
      RELPN = RELPOS + PVVEL*DT+0.5*PVACC*DTSQ - VELOLD*DT-0.5*ACC*DTSQ
C-----IF THE NEW RELATIVE POSITION IS GT 80 PERCENT OF THE OLD RELATIVE
C-----POSITION THEN GO TO 1020 AND INITIATE THE ACCELERATION TO ACC
                     IF ( RELPN.GT.0.80*RELPOS )  GO TO 1020
C-----CALCULATE THE ACC/DEC THAT WOULD MOVE THE VEHICLE NOT MORE THAN 20
C-----PERCENT OF HIS OLD RELATIVE POSITION IN DT SECONDS
      ACC = AMAX1(2.0*(0.2*RELPOS-VELOLD*DT)/DTSQ,0.0)
 1020 CONTINUE
C-----IF THE VEHICLES ACC/DEC IS LT THE DESIRED ACC/DEC THEN GO TO 3010
C-----AND MOVE THE VEHICLES ACC/DEC TO ACC IN PIJR TIME
                     IF ( ACCOLD . LT . ACC )       GO TO 3010
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC
C-----TO ACC IN DT SECONDS
      SLPNEW = (ACC-ACCOLD)/DT
C-----BOUND THE VEHICLES ACC/DEC SLOPE AND CHECK THE NEW VELOCITY
      SLPNEW = AMIN1(AMAX1(SLPNEW,-CRISLP),1.3*CRISLP)
C3    IPFLAG = 10HACCELERATE
      GO TO 3020
 2010 CONTINUE
C-----CHECK TO SEE IF THE VEHICLE SHOULD BEGIN TO DECELERATE TO HIS
C-----DESIRED SPEED BY THE TIME HE REACHES THE END OF HIS LANE
C3    IPFLAG = 10HDEC DESPD
      SLPNEW = -0.25*CRISLP
                     IF ( ACCOLD . LT . SLPNEW )  SLPNEW = 0.5*SLPNEW
                     IF ( ACCOLD . EQ . 0.0 )     ACCOLD = 1.0E-6
      A = ACCOLD/6.0
      B = (2.0*VELOLD+DESVEL)/3.0
      C = POSOLD - AMIN1(PVPOS,ENDLN-DESVEL)
      RADICL = B**2 - 4.0*A*C
                     IF ( RADICL . LE . 0.0 )     GO TO 2020
      T = (-B+SQRT(RADICL))/(2.0*A)
                     IF ( T . LE . 0.0 )          GO TO 2020
C-----FIND THE ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES VELOCITY TO
C-----HIS DESIRED SPEED BEFORE HE REACHES THE END OF HIS LANE AND BOUND
C-----THE ACC/DEC SLOPE
      SLPNEW = AMIN1(SLPNEW,2.0*(DESVEL-VELOLD-ACCOLD*T)/T**2)
 2020 CONTINUE
                     IF ( ACCOLD . GE . 0.0 )     GO TO 2030
C-----FIND THE ACC/DEC SLOPE REQUIRED TO BRING THE ACC/DEC TO ZERO BY
C-----THE TIME THE VEHICLES VELOCITY REACHES HIS DESIRED SPEED
      SLOPE = -0.5*ACCOLD**2/(DESVEL-VELOLD)
                     IF ( SLOPE.LT.0.40*CRISLP )  GO TO 2030
C-----SET THE ACC/DEC SLOPE TO BRING THE ACC/DEC TO ZERO BY THE TIME THE
C-----VEHICLES VELOCITY REACHES HIS DESIRED SPEED
      SLPNEW = SLOPE
 2030 CONTINUE
C-----BOUND THE ACC/DEC SLOPE TO DECELERATE TO HIS DESIRED SPEED
      SLPNEW = AMIN1(AMAX1(SLPNEW,-CRISLP),CRISLP)
      RETURN
 3010 CONTINUE
C-----THE VEHICLES OLD ACC/DEC IS LT THE NEW ACC/DEC THUS IF THE
C-----VEHICLES RELATIVE POSITION IS LE ZERO THEN GO TO 4010 AND REDUCE
C-----THE VEHICLES ACC/DEC TO ZERO
                     IF ( RELPOS . LE . 0.0 )     GO TO 4010
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC
C-----TO THE NEW ACC IN PIJR TIME
C3    IPFLAG = 10HMOVE ACC
      SLPNEW = 1.01*(ACC-ACCOLD)/PIJR(IDRICL)
C-----BOUND THE ACC/DEC SLOPE FOR ACCELERATION TO ACC IN PIJR TIME
      SLPNEW = AMIN1(AMAX1(SLPNEW,SLPOLD),1.3*CRISLP)
      ACCNEW = ACCOLD + SLPNEW*DT
C-----IF THE VEHICLES ACC/DEC AFTER DT SECONDS WILL STILL BE LT ACC THEN
C-----GO TO 3020 AND CHECK THE VELOCITY AFTER DT SECONDS ELSE CALCULATE
C-----THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC TO ACC IN
C-----ONE DT AND CHECK VELOCITY AFTER DT SECONDS
                     IF ( ACCNEW . LT . ACC )     GO TO 3020
      SLPNEW = (ACC-ACCOLD)/DT
 3020 CONTINUE
C-----CHECK TO SEE THAT THE VEHICLES VELOCITY WOULD NOT BE ABOVE THE
C-----DESIRED SPEED AFTER THE ACC/DEC FOR THE VEHICLE WAS REDUCED TO
C-----ZERO AT HALF THE CRITICAL SLOPE
C3          ENCODE  ( 10,701,JPFLAG )           ACC
      SLOPE = -0.50*CRISLP
      T = AMAX1(-ACCOLD/SLOPE,0.01)
      VT = VELOLD + ACCOLD*T + 0.5*SLOPE*T**2
                     IF ( VT . LT . DESVEL )      RETURN
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED SO THAT VT WOULD NOT EXCEED
C-----THE DESIRED SPEED BEFORE THE ACC/DEC COULD BE REDUCED TO ZERO AND
C-----BOUND THE ACC/DEC SLOPE
      SLPNEW = AMIN1(AMAX1((VT/DESVEL)*(-ACCOLD/T),-CRISLP),1.3*CRISLP)
      RETURN
 4010 CONTINUE
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES
C-----ACC/DEC TO ZERO IN ONE DT AND BOUND THE ACC/DEC SLOPE
C3    IPFLAG = 10HREDUCE A/D
C3    JPFLAG = 10HTO 0 ACCEL
      SLPNEW = AMIN1(AMAX1(-ACCOLD/DT,-CRISLP),CRISLP)
      RETURN
      END                                                              ACCEL
```

```
      SUBROUTINE CRIDIS (K)                                              COLEASE
C     TASK,CRIDIS,K
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / VEHD   / ISLP    ,IACC    ,IVEL    ,IPOS    ,            COLEASE
     *                  ISET    ,LCHGE   ,ISPDP   ,LEGAL   ,            COLEASE
     *                  IPRTM   ,ITIMV   ,IQDS    ,ISPDS   ,            COLEASE
     *                  ISDS    ,IDVS    ,ISTCON  ,IVMAXA  ,            COLEASE
     *                  IVMAXD  ,LATPOS  ,IDTS    ,LALT    ,            COLEASE
     *                  NORC    ,LOGFLG  ,MSTPF   ,MLAG    ,            COLEASE
     *                  MTCARS  ,MFINL   ,MSFLG   ,MPOBS   ,            COLEASE
     *                  MOASF   ,MSAOR   ,MPRO    ,MBLOCK  ,            COLEASE
     *                  MININT  ,IFVA    ,IACDS   ,ICDFS   ,            COLEASE
     *                  ISDEC   ,ISTMO   ,IACLDS  ,IRSTOP               COLEASE
      COMMON / VEHF   / IDRICL  ,IVEHCL  ,ISPD    ,NOF     ,            COLEASE
     *                  NOR     ,LNEXT   ,LPRES   ,ITURN   ,            COLEASE
     *                  IBAPS   ,IPRTLO  ,IEXTIM  ,NOBAPD               COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACDCP(3),VCARFO(20),VACCEL(12),CRISLP,DECMAX,
     *                  DENOM,OLDACC,RADICL,REACTT,RELNEW,RELOLD,T,V,
     *                  VSQT4,VT2,X,XCRIT,VADLVA(6),VHOLDS(2),ZTEMPD(53)
      DATA      F3    / -1.33333333333333 /
      DATA      N1,N2 / 4HCRID,2HIS /
C3701 FORMAT(3HDM=F7.3)
C3702 FORMAT(3HSN=F7.3)
C
C-----SUBROUTINE CRIDIS CHECKS CRITICAL STOPPING DISTANCE FOR A
C-----DECELERATION TO A STOP AND IF VIOLATED THEN INITIATES A
C-----DECELERATION TO A STOP
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME .GT. NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR CRIDIS
C3    IPFLAG = 10HSTEADY DIS
      RELOLD = PVPOS - POSOLD
      CRISLP = 4.0*DCHAR(IDRICL)
C-----INITIALIZE OLDACC AND REACTT FOR A NORMAL DECELERATION TO A STOP
C-----(REDUCE ACCOLD TO ZERO IN PIJR TIME)
      OLDACC = 0.0
      REACTT = PIJR(IDRICL)
C-----IF THIS IS THE FIRST VEHICLE IN THE LANE THAT DECIDED TO STOP ON
C-----AN AMBER SIGNAL INDICATION THEN RESET THE REACTION TIME TO ZERO
                    IF ( IGO .EQ. 2 )           REACTT = 0.0
                    IF ( ACCOLD .GE. 0.0 )      GO TO 1030
 1010 CONTINUE
C-----SET OLDACC AND REACTT FOR A QUICK DECELERATION TO A STOP AND CHECK
C-----FOR A DECELERATION FOR A STOP (DECELERATION STARTS WITH THE
C-----CURRENT VALUE OF ACCOLD AND NO REACTION TIME)
      OLDACC = ACCOLD
      REACTT = 0.0
      K = 1
      GO TO 3010
 1020 CONTINUE
C-----SET REACTT TO PIJR TIME FOR THE DRIVER
      REACTT = PIJR(IDRICL)
 1030 CONTINUE
C-----FIND THE MAXIMUM DECELERATION RATE THAT THE DRIVER WOULD USE TO
C-----STOP FROM HIS OLD VELOCITY USING LINEAR DECELERATION AND BOUND
C-----IT WITH THE MAXIMUM DECELERATION RATE FOR THE VEHICLE
      DECMAX = DUTOL*(-6.0-VELOLD/44.0)*DCHAR(IDRICL)
      DECMAX = AMAX1(DECMAX,DMAX(IVEHCL))
C-----COMPUTE THE CRITICAL STOPPING DISTANCE FOR THE VEHICLE
      XCRIT = VELOLD*(REACTT+DT+F3*VELOLD/DECMAX)
C-----SET K FOR CRITICAL STOPPING DISTANCE VIOLATED THIS DT
      K = 1
C-----IF THE CRITICAL STOPPING DISTANCE IS VIOLATED THIS DT THEN GO TO
C-----3010 AND CHECK FOR A DECELERATION FOR A STOP
                    IF ( RELOLD .LE. XCRIT )    GO TO 3010
C-----IF THIS VEHICLE IS THE FIRST VEHICLE IN THE LANE WHICH DECIDED TO
C-----STOP ON AN AMBER SIGNAL INDICATION AND THE REACTION TIME IS EQ
C-----ZERO AND CRITICAL STOPPING DISTANCE IS NOT VIOLATED THIS DT THEN
C-----GO TO 1020 AND SET REACTT TO PIJR FOR THE DRIVER AND CHECK AGAIN
               IF ( IGO.EQ.2.AND.REACTT.EQ.0.0 )    GO TO 1020
C-----SET K FOR CRITICAL STOPPING DISTANCE NOT VIOLATED THIS DT OR
C-----WITHIN PIJR TIME
      K = 2
C-----CALCULATE THE NEW RELATIVE POSITION AFTER PIJR SECONDS OR THE TIME
C-----REQUIRED TO REDUCE THE VEHICLES ACC/DEC TO ZERO AT CRISLP
      T = AMAX1(PIJR(IDRICL),ACCOLD/CRISLP)
      RELNEW = RELOLD - VELOLD*T - 0.5*ACCOLD*T**2 - SLPNEW*T**3/6.0
C-----IF THE CRITICAL STOPPING DISTANCE WILL NOT BE VIOLATED WITHIN PIJR
C-----TIME THEN RETURN AND ACCELERATE ACCORDING TO DESIRED SPEED
                    IF ( RELNEW .GT. XCRIT )    RETURN
C-----SET K FOR CRITICAL STOPPING DISTANCE VIOLATED WITHIN PIJR TIME
      K = 3
C-----IF THE VEHICLE WAS DECELERATING THEN CHECK FOR DECELERATION TO
C-----DESIRED SPEED
                    IF ( ACCOLD .LT. 0.0 )      GO TO 7020
C-----REDUCE THE VEHICLES ACCELERATION TO ZERO FOR UPCOMING DECELERATION
C-----TO A STOP
C3    IPFLAG = 10HREDUCE ACC
C3    JPFLAG = 10HFOR DECEL
      T = 0.0
 2010 CONTINUE
      T = T + DT
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO REDUCE THE ACCELERATION TO
C-----0.01 IN T SECONDS AND FIND THE VELOCITY AND POSITION OF THE
C-----VEHICLE AFTER T SECONDS
      SLPNEW = AMAX1((0.01-ACCOLD)/T,-CRISLP)
      V = VELOLD + ACCOLD*T + 0.5*SLPNEW*T**2
      X = VELOLD*T + 0.5*ACCOLD*T**2 + SLPNEW*T**3/6.0
C-----CALCULATE THE CRITICAL STOPPING DISTANCE AFTER T SECONDS
      DECMAX = DUTOL*(-6.0-V/44.0)*DCHAR(IDRICL)
      XCRIT = V*(REACTT+DT+F3*V/DECMAX)
C-----IF THE CRITICAL STOPPING DISTANCE WILL NOT BE VIOLATED WITHIN T
C-----SECONDS THUS GO TO 2010 AND INCREASE T BY DT AND CHECK AGAIN ELSE
C-----USE THE SLOPE TO CALCULATE THE NEW POS/VEL/ACC
                    IF ( RELOLD-X .GT. XCRIT ) GO TO 2010
      GO TO 3020
 3010 CONTINUE
      RELOLD = AMAX1(RELOLD,0.01)
      RELNEW = RELOLD - VELOLD*REACTT
C-----IF THE NEW RELATIVE POSITION WILL BE LT 20 PERCENT OF THE OLD
C-----RELATIVE POSITION AND THE REACTION TIME IS GT ZERO THEN GO TO 7010
C-----AND REDUCE THE REACTION TIME BY DT AND CHECK AGAIN
      IF ( RELNEW.LT.0.2*RELOLD .AND. REACTT.GT.0.0 )
     *                                                  GO TO 7010
C-----CALCULATE A DECELERATION TO A STOP
      DENOM = 6.0*RELNEW
      VT2 = 2.0*VELOLD
      VSQT4 = VT2*VT2
      RADICL = VSQT4 + DENOM*OLDACC
C-----IF THE DECELERATION TO A STOP CAN NOT BE CALCULATED THEN GO TO
C-----4010 AND REDUCE ANY DECELERATION TO ZERO
                    IF ( RADICL .LE. 0.0 )      GO TO 4010
      DECMAX = -OLDACC - (VSQT4+VT2*SQRT(RADICL))/DENOM
C-----CALCULATE THE ACC/DEC SLOPE FOR A DECELERATION TO A STOP
```

```
      SLPNEW = (OLDACC-DECMAX)*(OLDACC+DECMAX)/VT2
C-----IF THE ACC/DEC SLOPE FOR A DECELERATION TO A STOP IS GT -0.3 THEN
C-----GO TO 4010 AND REDUCE ANY DECELERATION TO ZERO
                      IF ( SLPNEW . GT . -0.3 )      GO TO 4010
C-----IF THE ACC/DEC SLOPE FOR A DECELERATION TO A STOP IS LT -1.2 TIMES
C-----CRITICAL SLOPE AND THE REACTION TIME IS GT ZERO THEN GO TO 7010
C-----AND DECREASE THE REACTION TIME BY DT AND CHECK AGAIN
      IF ( SLPNEW.LT.-2.0*CRISLP . AND . REACTT.GT.0.0 )
     *                                           GO TO 7010
C-----BOUND THE ACC/DEC SLOPE FOR A DECELERATION TO A STOP
      SLPNEW = AMAX1(SLPNEW,-12.0)
C-----IF THE LANE CHANGE ACC/DEC SLOPE IS LT THE ACC/DEC SLOPE FOR A
C-----DECELERATION TO A STOP THEN GO TO 3030 AND DO NOT INITIATE A
C-----DECELERATION TO A STOP ELSE INITIATE A DECELERATION TO A STOP
                      IF ( SLPLCH . LT . SLPNEW )  GO TO 3030
      IPRTM = REACTT/DT + 0.5
      MSFLG = LTRUE
C3    IPFLAG = 10HDECEL PIJR
C3          ENCODE  ( 10,701,JPFLAG )            DECMAX
C-----IF THERE IS REACTION TIME THEN GO TO 5010 AND HOLD THE SPEED
                      IF ( IPRTM . GT . 0 )        GO TO 5010
      ACCOLD = AMIN1(ACCOLD,0.0)
C3    IPFLAG = 10HDECEL OARS
 3020 CONTINUE
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER DT SECONDS
      CALL  NEWVEL  ( DT,DTSQ,DTCU )
      RETURN
 3030 CONTINUE
C-----THE LANE CHANGE ACC/DEC SLOPE IS LT THE ACC/DEC SLOPE FOR A
C-----DECELERATION TO A STOP THUS DO NOT INITIATE THE DECELERATION FOR
C-----A STOP
C3    IPFLAG = 10HDECEL LCHG
C3          ENCODE  ( 10,702,KPFLAG )            SLPNEW
      GO TO 3020
 4010 CONTINUE
C-----REDUCE THE VEHICLES ACC/DEC TO ZERO
C3    IPFLAG = 10HREDUCE DEC
C3    JPFLAG = 10HPOS SLOPE
C-----IF THE VEHICLES ACC/DEC IS GT -0.004 THEN GO TO 6010 AND SET K FOR
C-----CRITICAL STOPPING DISTANCE NOT VIOLATED THIS DT OR WITHIN PIJR
C-----THUS ACCELERATE ACCORDING TO DESIRED SPEED
                      IF ( ACCOLD . GE . -0.004 )  GO TO 6010
      SLPNEW = 1.3*CRISLP
      ACCNEW = ACCOLD + SLPNEW*DT
C-----IF THE ACC/DEC WILL BE LE -0.004 AFTER DT SECONDS THEN GO TO 3020
C-----AND CALCULATE THE NEW POS/VEL/ACC ELSE CALCULATE THE ACC/DEC
C-----SLOPE REQUIRED TO REDUCE THE ACC/DEC TO -0.004 IN ONE DT
                      IF ( ACCNEW . LE . -0.004 )  GO TO 3020
      SLPNEW = (-0.004-ACCOLD)/DT
      GO TO 3020
 5010 CONTINUE
C-----HOLD THE VEHICLES SPEED AT ITS CURRENT VALUE
      CALL  HOLDSP  ( IPRTM )
      RETURN
 6010 CONTINUE
C-----SET K FOR CRITICAL STOPPING DISTANCE NOT VIOLATED THIS DT OR
C-----WITHIN PIJR TIME SO ACCELERATE ACCORDING TO DESIRED SPEED
      K = 2
      RETURN
 7010 CONTINUE
C-----REDUCE THE REACTION TIME BY DT AND RE-CALCULATE A DECELERATION TO
C-----A STOP
      REACTT = REACTT - DT
      GO TO 3010
 7020 CONTINUE
C-----CHECK FOR DECELERATION TO DESIRED SPEED
      SLPNEW = 0.0
C-----IF THE VEHICLE IS BELOW HIS DESIRED SPEED THEN GO TO 3020 AND
C-----CALCULATE THE POS/VEL/ACC USING AN ACC/DEC SLOPE OF ZERO
            IF ( VELOLD . LE . DESVEL+1.0*DT )   GO TO 3020
C-----ACCELERATE ACCORDING TO THE DESIRED SPEED FOR THIS VEHICLE
      CALL  ACCEL
      GO TO 3020
      END
```

CRIDIS

```
      SUBROUTINE ADLVAI                                                 COLEASE
C     TASK,ADLVAI
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACDCP(3),VCARFO(20),VACCEL(12),VCRIDI(14),I,
     *                  IVATIN,J,JSNA,JV,MPRES,VHOLDS(2),ZTEMPD(53)
      DIMENSION         MSG908(8),MSG909(6)
      DATA      MSG908 / 4H IV ,4HALRE,4HADY ,4HON L,4HVATI,4HN = ,
     *                   4HADLV,4HAI   /
      DATA      MSG909 / 4H NVA,4HTIN ,4HGT 2,4H5 = ,4HADLV,4HAI   /
      DATA      N1,N2  / 4HADLV,2HAI /
C4701 FORMAT(9H IVATIN =,I3,9H NVATIN =,I3,9H LVATIN =,14I7,/,33X,11I7)
C4702 FORMAT(24X,9H TVATIN =,14F7.1,/,33X,11F7.1)
C
C-----SUBROUTINE ADLVAI ADDS THE STOPPED VEHICLE TO THE LIST OF VEHICLES
C-----AT THE INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                    IF ( LCHGE . NE . 2 )        GO TO 1010
C-----END THE LANE CHANGE AND RESET THE LANE CHANGE FLAG
      CALL  ENDLCH
C-----FIND AN INTERSECTION PATH FOR THIS VEHICLE BASED ON THE CURRENT
C-----APPROACH, CURRENT LANE, AND THE DESIRED OUTBOUND APPROACH
      CALL  PATHF    ( LTRUE,N1,N2 )
 1010 CONTINUE
C-----IF THE VEHICLE MAY PROCEED INTO THE INTERSECTION THEN RETURN
                    IF ( MPRO . EQ . LTRUE )     RETURN
C-----IF THE INTERSECTION IS SIGNAL CONTROLLED THEN RETURN
                    IF ( ICONTR . GT . 4 )       RETURN
      IVATIN = 1
C-----IF THERE ARE NO VEHICLES ON THE LIST OF VEHICLES AT THE
C-----INTERSECTION THEN ADD THIS VEHICLE AS THE FIRST VEHICLE ON THE
C-----LIST OF VEHICLES AT THE INTERSECTION
                    IF ( NVATIN . LE . 0 )       GO TO 4020
C-----CHECK EACH VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION TO
C-----SEE IF THIS VEHICLE IS ALREADY ON THE LIST
      DO 2010  IVATIN = 1 , NVATIN
C-----IF THIS VEHICLE IS ALREADY ON THE LIST OF VEHICLES AT THE
C-----INTERSECTION THEN ERROR
                    IF ( IV.EQ.LVATIN(IVATIN) )  GO TO 9080
 2010 CONTINUE
C-----CHECK EACH VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION TO
C-----SEE IF ANY SHOULD YIELD TO THIS VEHICLE
      DO 3010  IVATIN = 1 , NVATIN
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION ARRIVED
C-----GREATER THAN PIJR SECONDS AGO THEN SKIP TO THE NEXT VEHICLE ON THE
C-----LIST OF VEHICLES AT THE INTERSECTION
      IF ( TIME-TVATIN(IVATIN).GT.PIJR(IDRICL) ) GO TO 3010
      JV = LVATIN(IVATIN)
C     COLEASE,FIND,MPRES,VEHF,JV,LPRES
      CALL  FIND     (MPRES ,      7,JV     ,      7)                   COLEASE
C     COLEASE,FIND,JSNA,LANE,MPRES,ISNA
      CALL  FIND     (JSNA  ,      3,MPRES  ,      4)                   COLEASE
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION IS ON
C-----AN APPROACH TO THE LEFT THEN HE SHOULD YIELD TO ME
                    IF ( JSNA . EQ . IALEFT )    GO TO 3020
 3010 CONTINUE
C-----NONE OF THE VEHICLES ON THE LIST OF VEHICLES AT THE INTERSECTION
C-----SHOULD YIELD TO ME SO ADD THIS VEHICLE TO THE END OF THE LIST
      IVATIN = NVATIN + 1
 3020 CONTINUE
                    IF ( IVATIN . GT . NVATIN )  GO TO 4020
C-----MOVE EACH VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION FROM
C-----IVATIN TO NVATIN DOWN ONE TO MAKE ROOM FOR THIS VEHICLE AT IVATIN
      DO 4010  I = IVATIN , NVATIN
      J = NVATIN - I + IVATIN
      LVATIN(J+1) = LVATIN(J)
      TVATIN(J+1) = TVATIN(J)
 4010 CONTINUE
 4020 CONTINUE
C-----INCREMENT THE NUMBER OF VEHICLES AT THE INTERSECTION
      NVATIN = NVATIN + 1
                    IF ( NVATIN . GT . 25 )      GO TO 9090
C-----SET THIS VEHICLE AS THE IVATIN VEHICLE ON THE LIST OF VEHICLES
C-----AT THE INTERSECTION
      LVATIN(IVATIN) = IV
C-----SET THE TIME THIS VEHICLE ARRIVED AT THE INTERSECTION TO THE TIME
C-----THE NEXT VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION
C-----ARRIVED AT THE INTERSECTION (HE SHOULD YIELD TO ME)
      TVATIN(IVATIN) = TVATIN(IVATIN+1)
C-----IF THIS VEHICLE IS THE LAST VEHICLE ON THE LIST OF VEHICLES AT THE
C-----INTERSECTION THEN SET THE TIME THIS VEHICLE ARRIVED AT THE
C-----INTERSECTION TO THE TIME INTO THE SIMULATION
                    IF ( IVATIN . EQ . NVATIN )  TVATIN(IVATIN) = TIME
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 101
C4                  IF ( TIME . LT . TPRINT )    GO TO 101
C4    PRINT 701 , IVATIN,NVATIN,(LVATIN(I),I=1,NVATIN)
C4    PRINT 702 , (TVATIN(I),I=1,NVATIN)
C4101 CONTINUE
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9080 CONTINUE
      CALL  ABORTR   ( MSG908,30 )
      STOP  908
 9090 CONTINUE
      CALL  ABORTR   ( MSG909,22 )
      STOP  909
      END                                                               ADLVAI
```

EE M

```
      SUBROUTINE HOLDSP ( KPRTM )
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACDCP(3),VCARFO(20),VACCEL(12),VCRIDI(14),
     *                  VADLVA(6),ACCHLD,LPRTM,ZTEMPD(53)
      DATA    N1,N2   / 4HHOLD,2HSP /
C
C-----SUBROUTINE HOLDSP HOLDS THE VEHICLES SPEED AT ITS CURRENT VALUE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SAVE THE CURRENT VALUE OF SOME OF THE VEHICLES PARAMETERS
      LPRTM = KPRTM
      ACCHLD = ACCOLD
      SLPOLD = SLPNEW
C-----SET THE VEHICLES ACC/DEC AND ACC/DEC SLOPE TO ZERO TO HOLD THE
C-----SPEED
      ACCOLD = 0.0
      SLPNEW = 0.0
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER DT SECONDS
      CALL  NEWVEL  ( DT,DTSQ,DTCU )
C-----RESET THE VEHICLES ACC/DEC AND ACC/DEC SLOPE
      ACCNEW = ACCHLD - ACCHLD/LPRTM
      SLPNEW = SLPOLD
      RETURN
      END                                                               HOLDSP
```

```
C6    SUBROUTINE PVAPRT
C6    COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
C6   *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
C6   *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
C6    COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
C6   *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
C6    COMMON / PRTPVA / DISTAD(200)
C6    COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
C6   *                  NUMV
C6    COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
C6    COMMON / ZTEMPD / IFORM(2),IQACC,IQPOS,IQV,IQVEL,V,ZTEMPD(103)
C6    DATA    N1,N2   / 4HPVAP,2HRT /
C6701 FORMAT(5H(*+*,I3,3HX,*,I1,2H*))
C
C-----SUBROUTINE PVAPRT PRINTS POS/VEL/ACC FOR THE VEHICLE
C
C6    NRNAME = NRNAME + 1
C6    IRNAME(1,NRNAME) = N1
C6    IRNAME(2,NRNAME) = N2
C6                  IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----FIND THE ONES DIGIT OF THE VEHICLE NUMBER
C6    V = IQ(IV)/10.0
C6    IQV = (V-IFIX(V))*10.0 + 0.5
C-----CONVERT AND WRITE THE VEHICLES POSITION FOR THE PAGE PLOT
C6    IQPOS = MIN0(IFIX((POSNEW+DISTAD(IV))/18.5+9.5),134)
C6          ENCODE  ( 14,701,IFORM )                   IQPOS,IQV
C6    WRITE (1,IFORM)
C-----CONVERT AND WRITE THE VEHICLES VELOCITY FOR THE PAGE PLOT
C6    IQVEL = MIN0(IFIX(VELNEW*2.0+9.5),134)
C6          ENCODE  ( 14,701,IFORM )                   IQVEL,IQV
C6    WRITE (2,IFORM)
C-----CONVERT AND WRITE THE VEHICLES ACC/DEC FOR THE PAGE PLOT
C6    IQACC = MIN0(MAX0(IFIX(-ACCNEW*5.0+59.5),9),134)
C6          ENCODE  ( 14,701,IFORM )                   IQACC,IQV
C6    WRITE (3,IFORM)
C6    RETURN
C6    END                                                               PVAPRT
```

```
      SUBROUTINE INTLOG                                                 COLEASE
C     TASK,INTLOG
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP    ,IACC    ,IVEL    ,IPOS    ,            COLEASE
     *                  ISET    ,LCHGE   ,ISPDP   ,LEGAL   ,            COLEASE
     *                  IPRTM   ,ITIMV   ,IQDS    ,ISPDS   ,            COLEASE
     *                  ISDS    ,IDVS    ,ISTCON  ,IVMAXA  ,            COLEASE
     *                  IVMAXD  ,LATPOS  ,IDTS    ,LALT    ,            COLEASE
     *                  NORC    ,LOGFLG  ,MSTPF   ,MLAG    ,            COLEASE
     *                  MTCARS  ,MFINL   ,MSFLG   ,MPOBS   ,            COLEASE
     *                  MOASF   ,MSAOR   ,MPRO    ,MBLOCK  ,            COLEASE
     *                  MININT  ,IFVA    ,IACDS   ,ICDFS   ,            COLEASE
     *                  ISDEC   ,ISTMO   ,IACLDS  ,IRSTOP               COLEASE
      COMMON / VEHF   / IDRICL  ,IVEHCL  ,ISPD    ,NOF     ,            COLEASE
     *                  NOR     ,LNEXT   ,LPRES   ,ITURN   ,            COLEASE
     *                  IBAPS   ,IPRTLO  ,IEXTIM  ,NOBAPD               COLEASE
      COMMON / VEHIL  / MOEDIC  ,MINFLZ  ,MLUNC   ,MIUNC   ,            COLEASE
     *                  MLYELD  ,MLSTOP  ,MATSTL  ,MSSRED  ,            COLEASE
     *                  MLRTOR  ,MSSGRN  ,MCHKCF  ,MOUMIL  ,            COLEASE
     *                  IDEDIC  ,INFLZ   ,ILUNC   ,ILYELD  ,            COLEASE
     *                  ILSTOP  ,ICONTN  ,ICHKCF  ,IERROR               COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         MSG910(8),MSG911(11)
      DATA      F3    / -1.33333333333333 /
      DATA      MSG910 / 4H NO ,4HLANE,4H CON,4HTROL,4H SET,4H - I,
     *                  4HNTLO,4HG    /
      DATA      MSG911 / 4H NO ,4HVEHI,4HL DE,4HPEND,4HENT ,4HATTR,
     *                  4HIBUT,4HE TR,4HUE -,4H INT,4HLOG   /
      DATA      N1,N2  / 4HINTL,2HOG /
C
C-----SUBROUTINE INTLOG CHECKS THE INTERSECTION CONTROL LOGICAL
C-----DEPENDENT ATTRIBUTES AND CALL THE APPROPRIATE INTERSECTION CONTROL
C-----ROUTINES
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----CHECK THE INTERSECTION CONTROL LOGICAL DEPENDENT ATTRIBUTES
                    IF ( ICONTN . EQ . LFALSE )  GO TO 1010
C-----THE VEHICLE SHOULD CONTINUE AS PRESENTLY
      RETURN
 1010 CONTINUE
                    IF ( ILUNC . EQ . LFALSE )   GO TO 2010
C-----FOLLOW THE UNCONTROLLED LANE LOGIC (UNCONTROLLED LANE AT
C-----UNCONTROLLED INTERSECTION) THUS IF THE VEHICLE IS STOPPED
C-----AT THE STOP LINE THEN FOLLOW THE STOP SIGN CONTROLLED LOGIC ELSE
C-----CHECK SIGHT DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECK
C-----INTERSECTION CONFLICTS AND IF CLEAR THEN THIS VEHICLE MAY PROCEED
C-----INTO THE INTERSECTION
                    IF ( MATSTL . EQ . LTRUE )   GO TO 3020
      GO TO 4020
 2010 CONTINUE
                    IF ( ILYELD . EQ . LFALSE )  GO TO 3010
C-----FOLLOW THE YIELD SIGN CONTROLLED LOGIC THUS IF THIS VEHICLE IS THE
C-----FIRST VEHICLE IN THE LANE OR THE VEHICLE AHEAD MAY PROCEED INTO
C-----THE INTERSECTION THEN FOLLOW THE STOP SIGN CONTROLLED LOGIC EVEN
C-----THOUGH THIS VEHICLE IS NOT STOPPED AT THE STOP LINE ELSE RETURN
C-----AND CHECK AGAIN NEXT DT
```

```
                    IF ( NOF . EQ . 0 )          GO TO 3020
C     COLEASE,FIND,NPRO,VEHD,NOF,MPRO
      CALL  FIND    (NPRO  ,      6,NOF    ,     31)                    COLEASE
                    IF ( NPRO . EQ . LTRUE )     GO TO 3020
      RETURN
 3010 CONTINUE
                    IF ( ILSTOP . EQ . LFALSE )  GO TO 4010
 3020 CONTINUE
C-----FOLLOW THE STOP SIGN CONTROLLED LOGIC THUS IF THE VEHICLE MAY
C-----PROCEED INTO THE INTERSECTION THEN RETURN
                    IF ( MPRO . EQ . LTRUE )     RETURN
C-----CHECK TO SEE IF THE VEHICLE MAY ENTER THE INTERSECTION WITHOUT
C-----BLOCKING ANY VEHICLE STOPPED AT THE INTERSECTION BEFORE THIS
C-----VEHICLE AND IF OK THEN CHECK SIGHT DISTANCE RESTRICTIONS AND IF
C-----CLEAR THEN CHECK INTERSECTIONS CONFLICTS AND IF CLEAR THEN THE
C-----VEHICLE MAY PROCEED INTO THE INTERSECTION
      CALL  LSTOP
      RETURN
 4010 CONTINUE
                    IF ( ICHKCF . EQ . LFALSE )  GO TO 5010
 4020 CONTINUE
C-----THIS VEHICLE MUST CHECK FOR CONFLICTS THUS IF THE VEHICLE MAY
C-----PROCEED INTO THE INTERSECTION OR THE VEHICLE IS NOT THE FIRST
C-----VEHICLE IN THE LANE OR THE TRAFFIC CONTROL AHEAD REQUIRES THIS
C-----VEHICLE TO STOP THEN RETURN AND CONTINUE AS PRESENTLY
                    IF ( MPRO . EQ . LTRUE )     RETURN
                    IF ( MFINL . EQ . LFALSE )   RETURN
                    IF ( MTCARS . EQ . LTRUE )   RETURN
C-----CHECK SIGHT DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECK
C-----INTERSECTIONS CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED
C-----INTO THE INTERSECTION
      CALL  CHKSDR
      RETURN
 5010 CONTINUE
                    IF ( INFLZ . EQ . LFALSE )   GO TO 6010
C-----THIS VEHICLE SHOULD CHECK TO SEE IF IT SHOULD BE WITHIN THE
C-----INFLUENCE ZONE OF THE INTERSECTION CONTROL THUS IF THE VEHICLE HAS
C-----NOT DEDICATED HIMSELF TO AN INTERSECTION PATH THEN RETURN AND WAIT
C-----UNTIL THE VEHICLE IS DEDICATED TO AN INTERSECTION PATH
                    IF ( LNEXT . EQ . 0 )        RETURN
C-----CALCULATE THE THRESHOLD DISTANCE FROM THE END OF THE LANE THAT THE
C-----VEHICLE SHOULD BECOME WITHIN THE INFLUENCE ZONE OF THE
C-----INTERSECTION CONTROL (LET 4+PIJR SECONDS AT THE CURRENT VELOCITY
C-----PLUS THE STOPPING DISTANCE BE THE THRESHOLD DISTANCE)
      DECMAX = DUTOL*(-6.0-VELNEW/44.0)*DCHAR(IDRICL)
      DECMAX = AMAX1(DECMAX,DMAX(IVEHCL))
      XCRIT = VELNEW*(4.0+PIJR(IDRICL)+F3*VELNEW/DECMAX)
C-----LET 400 FEET BE THE MINIMUM THRESHOLD DISTANCE
      XCRIT = AMAX1(XCRIT,400.0)
C-----IF THE DISTANCE FROM THE END OF THE END OF THE LANE IS GT THE
C-----THRESHOLD DISTANCE THEN RETURN AND WAIT UNTIL THE VEHICLE IS
C-----CLOSER
                    IF ( RELEND . GT . XCRIT )   RETURN
C-----INITIALIZE THE VEHICLES INTERSECTION CONTROL LOGICAL ATTRIBUTES
C-----BASED ON THE TYPE OF TRAFFIC CONTROL FOR THIS LANE
      CALL  INFLZN
      RETURN
 6010 CONTINUE
                    IF ( IDEDIC . EQ . LFALSE )  GO TO 9100
C-----THIS VEHICLE SHOULD CHECK TO SEE IF IT SHOULD DEDICATE ITSELF TO
C-----AN INTERSECTION PATH THUS CALCULATE THE THRESHOLD DISTANCE FROM
C-----THE START OF THE LANE THAT THE VEHICLE CAN DEDICATE ITSELF TO AN
C-----INTERSECTION PATH (LET THE THRESHOLD DISTANCE BE THE ACCEPTABLE
C-----LAG GAP FOR LANE CHANGING)
      XCRIT = (4.0+1.4*VELOLD)/(FACTOR*DCHAR(IDRICL)*VCHAR(IVEHCL))
      XCRIT = XCRIT + LENV(IVEHCL) + 4.0
C-----IF THE DISTANCE FROM THE START OF THE LANE IS LT THE THRESHOLD
C-----DISTANCE THEN RETURN AND WAIT UNTIL THE VEHICLE IS FURTHER DOWN
C-----THE LANE
                    IF ( POSNEW . LT . XCRIT )   RETURN
      MOEDIC = LTRUE
```

```
                   IF ( ISET . EQ . 6 )          ISET = 5
      LOGTMP = 2
      IREPIL = LTRUE
C-----FIND AN INTERSECTION PATH FOR THIS VEHICLE BASED ON THE CURRENT
C-----APPROACH, CURRENT LANE, AND THE DESIRED OUTBOUND APPROACH
      CALL  PATHF    ( LFALSE,N1,N2 )
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9100 CONTINUE
                   IF ( IERROR . EQ . LFALSE )  GO TO 9110
      CALL  ABORTR  ( MSG910,29 )
      STOP  910
 9110 CONTINUE
      CALL  ABORTR  ( MSG911,43 )
      STOP  911
      END                                                               INTLOG
```

```
      SUBROUTINE SIGRES (JSISET)                                        COLEASE
C   TASK,SIGRES,JSISET
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         IENT6(7),MSG912(8)
      EQUIVALENCE       (IFVA,IENT6(1))
      DATA      MSG912 / 4H JSI,4HSET ,4HLE 0,4H OR ,4HGT 2,4HS = ,
     *                  4HSIGR,4HES   /
      DATA      N1,N2  / 4HSIGR,2HES /
      DATA      T3     / =0.666666666666667 /
C
C-----SUBROUTINE SIGRES DETERMINES THE APPROPRIATE DRIVER RESPONSE FOR
C-----THE NEW SIGNAL INDICATION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                   IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                   IF ( JSISET . LE . 0 )       GO TO 9120
C-----INITIALIZE THE INTERSECTION CONTROL LOGIC TIMER TO PROCESS NEXT DT
      LOGTMP = 2
C-----IF THE SIGNAL INDICATION IS NOT FOR ALL TURN CODES THEN GO TO 5010
C-----AND PROCESS THE SIGNAL INDICATION BY TURN CODES
                   IF ( JSISET . GT . 4 )       GO TO 5010
C-----PROCESS THE SIGNAL INDICATION BY THE SIGNAL SETTING NUMBER
C-----             AG    AA    AR    AP
      GO TO ( 1010,2010,3010,4010 ) , JSISET
 1010 CONTINUE
C-----GREEN LIGHT IS DISPLAYED
      IPRTM = 0
                   IF ( MFINL . EQ . LFALSE )   GO TO 1020
                   IF ( MSSGRN . EQ . LTRUE )   GO TO 1020
                   IF ( VELOLD . GT . 0.0 )     GO TO 1020
C-----THIS VEHICLE IS THE FIRST VEHICLE IN HIS LANE AND HIS LAST SIGNAL
C-----INDICATION WAS NOT GREEN AND HE IS STOPPED THUS SET THE DELAY FOR
C-----THE FIRST VEHICLE IN THE QUEUE TO DISCHARGE
      IPRTM = 0.5/DT + IPIJR(IDRICL) + 0.1
                   IF ( ITURN . GT . 1 )        GO TO 1020
C-----THIS VEHICLE IS TURNING LEFT THUS SET THE INTERSECTION CONTROL
C-----LOGIC TIMER ALSO
      LOGTMP = MIN0(2+IPRTM,15)
```

```
 1020 CONTINUE
C-----SET THE INTERSECTION CONTROL LOGIC FOR GO ON GREEN
C3    KPFLAG = 10HSIG GREEN
      MSSGRN = LTRUE
      MSSRED = LFALSE
      MTCARS = LFALSE
      MSFLG = LFALSE
      JPRTM = IPRTM
      MCHKCF = LFALSE
      MPRO = LTRUE
C   COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
      CALL  FIND    (JLCH   ,      4,LNEXT  ,    72)                    COLEASE
C-----IF THIS VEHICLE IS NOT TURNING LEFT AND HIS INTERSECTION PATH DOES
C-----NOT CHANGE LANES WITHIN THE INTERSECTION THEN GO TO 6010 AND
C-----FINISH PROCESSING ELSE SET THAT THIS VEHICLE MUST CHECK FOR
C-----CONFLICTS AND MAY NOT PROCEED INTO THE INTERSECTION AND THEN GO TO
C-----6010 AND FINISH PROCESSING
            IF ( ITURN.GT.1 . AND . JLCH.EQ.0 )  GO TO 6010
      MCHKCF = LTRUE
      MPRO = LFALSE
      GO TO 6010
 2010 CONTINUE
C-----AMBER LIGHT IS DISPLAYED
C3    KPFLAG = 10HAMBR AGAIN
C-----IF THE LAST SIGNAL INDICATION WAS NOT GREEN THEN THIS IS NOT THE
C-----FIRST TIME THE VEHICLE HAS GONE THROUGH THE AMBER DECISION CODE
C-----THUS IMPLEMENT THE DECISION FROM LAST TIME BY GOING TO 6010 AND
C-----FINISH PROCESSING
                  IF ( MSSGRN . EQ . LFALSE )  GO TO 6010
C-----SET THE INTERSECTION CONTROL LOGIC FOR FOLLOW AMBER STOP
C3    KPFLAG = 10HFOL AM STP
      MSSGRN = LFALSE
      MSSRED = LTRUE
      MPRO = LFALSE
                  IF ( MFINL . EQ . LFALSE )    GO TO 2020
                  IF ( MCHKCF . EQ . LFALSE )   GO TO 2020
                  IF ( RELPOS . GT . 10.0 )     GO TO 2020
C-----THIS VEHICLE IS THE FIRST VEHICLE IN HIS LANE AND HE MUST CHECK
C-----FOR CONFLICTS AND HE IS AT THE STOP LINE THUS SET THE INTERSECTION
C-----CONTROL FLAGS SO THAT HE MAY PROCEED INTO THE INTERSECTION IF HIS
C-----CONFLICTS CLEAR
      LATPOS = LTRUE
      LOGFLG = 2
      GO TO 2080
 2020 CONTINUE
      MTCARS = LTRUE
                  IF ( IGO . LE . 1 )           GO TO 2030
C-----THE PREVIOUS VEHICLE DECIDED TO AMBER STOP THUS FOLLOW AMBER STOP
      IGO = 3
      GO TO 6010
 2030 CONTINUE
C-----IF THE VEHICLE IS DECELERATING TO A STOP OR HIS VELOCITY IS LE 0
C-----THEN GO TO 2040 AND AMBER STOP
                  IF ( MSFLG . EQ . LTRUE )     GO TO 2040
                  IF ( VELOLD . LE . 0.0 )      GO TO 2040
C-----CALCULATE THE CRITICAL STOPPING DISTANCE FOR A DECELERATION TO A
C-----STOP AT THE STOP LINE (HARD BRAKING)
      DECMAX = DUTOL*(-7.0-VELOLD/44.0)/DCHAR(IDRICL)
      DECMAX = AMAX1(DECMAX,DMAX(IVEHCL))
      DMPDI = DECMAX + ACCOLD
      XCRIT = VELOLD*(DT+(T3*VELOLD/DMPDI)*(2.0-ACCOLD/DMPDI))
C-----IF THE CRITICAL STOPPING DISTANCE IS GT THE DISTANCE TO THE STOP
C-----LINE THEN GO TO 2070 AND AMBER GO ELSE AMBER STOP
                  IF ( XCRIT . GT . RELEND )    GO TO 2070
 2040 CONTINUE
C-----SET THE INTERSECTION CONTROL LOGIC FOR AMBER STOP
C3    KPFLAG = 10HAMBER STOP
C-----IF THE VEHICLE MAY MAKE A LEFT-TURN-ON-RED OR A RIGHT-TURN-ON-RED
C-----THEN THE TRAFFIC CONTROL AHEAD DOES NOT REQUIRE A STOP
                  IF ( MLRTOR . EQ . LTRUE )   MTCARS = LFALSE
      MSFLG = LFALSE
C-----SET ALL ACC/DEC LOGICAL DEPENDENT ATTRIBUTES FALSE
      DO 2050  I = 1 , 7
      IENT6(I) = LFALSE
 2050 CONTINUE
C-----IF THERE IS NO VEHICLE AHEAD THEN GO TO 2060 AND SET THIS VEHICLE
C-----AS THE FIRST VEHICLE IN THIS LANE ELSE CHECK THE VEHICLE AHEAD
                  IF ( NOF . EQ . 0 )          GO TO 2060
C   COLEASE,FIND,NPRO,VEHD,NOF,MPRO
      CALL  FIND    (NPRO   ,      6,NOF    ,    31)                    COLEASE
C-----IF THE VEHICLE AHEAD MAY PROCEED INTO THE INTERSECTION THEN GO TO
C-----2060 AND SET THIS VEHICLE AS THE FIRST VEHICLE IN THIS LANE
                  IF ( NPRO . EQ . LTRUE )     GO TO 2060
C-----SET THIS VEHICLE TO FOLLOW THE VEHICLE AHEAD (FOLLOW AMBER STOP)
      IGO = 3
      MFINL = LFALSE
      MOASF = LFALSE
                  IF ( PVVEL . LE . 0.1 )      MOASF = LTRUE
C-----SET THIS VEHICLES ACC/DEC LOGIC TO FOLLOW THE VEHICLE AHEAD
      IFVA = LTRUE
C-----IF THE VEHICLE AHEAD IS NOT STOPPED THEN GO TO 6010 AND FINISH
C-----PROCESSING ELSE SET THIS VEHICLES ACC/DEC LOGIC TO CHECK CRITICAL
C-----STOPPING DISTANCE FOR A DECELERATION TO A STOP BEHIND THE VEHICLE
C-----AHEAD AND GO TO 6010 AND FINISH PROCESSING
                  IF ( MOASF . EQ . LFALSE )   GO TO 6010
      IFVA = LFALSE
      ISDEC = LTRUE
      GO TO 6010
 2060 CONTINUE
C-----SET THIS VEHICLE AS THE FIRST VEHICLE IN THIS LANE AND AMBER STOP
      IGO = 2
      MFINL = LTRUE
      MOASF = LTRUE
C-----RESET THE PREVIOUS VEHICLE PARAMETERS
      PVPOS = ENDLN
      PVVEL = 0.0
      PVACC = 0.0
C-----SET THE ACC/DEC LOGIC TO CHECK CRITICAL STOPPING DISTANCE FOR A
C-----DECELERATION TO A STOP AND GO TO 6010 AND FINISH PROCESSING
      ISDEC = LTRUE
      GO TO 6010
 2070 CONTINUE
C-----SET THE INTERSECTION CONTROL LOGIC FOR AMBER GO
      MPRO = LTRUE
      MCHKCF = LFALSE
 2080 CONTINUE
C3    KPFLAG = 10HAMBER GO
      MSSGRN = LTRUE
      MSSRED = LFALSE
      MTCARS = LFALSE
      MSFLG = LFALSE
C-----GO TO 6010 AND FINISH PROCESSING
      GO TO 6010
 3010 CONTINUE
C-----RED LIGHT IS DISPLAYED
C3    KPFLAG = 10HSIG RED GO
C-----IF THE VEHICLE MAY PROCEED INTO THE INTERSECTION THEN GO TO 6010
C-----AND FINISH PROCESSING (GO ON RED INDICATION)
                  IF ( MPRO . EQ . LTRUE )     GO TO 6010
C-----SET THE INTERSECTION CONTROL LOGIC FOR STOP ON RED
C3    KPFLAG = 10HSIG RED
      MSSGRN = LFALSE
      MSSRED = LTRUE
      MTCARS = LTRUE
C-----IF THE VEHICLE MAY MAKE A LEFT-TURN-ON-RED OR A RIGHT-TURN-ON-RED
C-----THEN THE TRAFFIC CONTROL AHEAD DOES NOT REQUIRE A STOP
                  IF ( MLRTOR . EQ . LTRUE )   MTCARS = LFALSE
      MPRO = LFALSE
C-----GO TO 6010 AND FINISH PROCESSING
      GO TO 6010
 4010 CONTINUE
C-----GREEN PROTECTED LIGHT IS DISPLAYED
```

```
      IPRTM = 0
                        IF ( MFINL . EQ . LFALSE )    GO TO 4020
                        IF ( MSSGRN . EQ . LTRUE )    GO TO 4020
                        IF ( VELOLD . GT . 0.0 )      GO TO 4020
C-----THIS VEHICLE IS THE FIRST VEHICLE IN HIS LANE AND HIS LAST SIGNAL
C-----INDICATION WAS NOT GREEN AND HE IS STOPPED THUS SET THE DELAY FOR
C-----THE FIRST VEHICLE IN THE QUEUE TO DISCHARGE
      IPRTM = 0.5/DT + IPIJR(IDRICL) + 0.5
                        IF ( ITURN . GT . 1 )         GO TO 4020
C-----THIS VEHICLE IS TURNING LEFT THUS SET THE INTERSECTION CONTROL
C-----LOGIC TIMER ALSO
      LOGTMP = MIN0(2+IPRTM,15)
 4020 CONTINUE
C-----SET THE INTERSECTION CONTROL LOGIC FOR GO ON PROTECTED GREEN
C3    KPFLAG = 10HSIG P GRN
      MSSGRN = LTRUE
      MSSRED = LFALSE
      MCHKCF = LFALSE
      MTCARS = LFALSE
      MSFLG = LFALSE
      JPRTM = IPRTM
      MPRO = LTRUE
C-----GO TO 6010 AND FINISH PROCESSING
      GO TO 6010
 5010 CONTINUE
                        IF ( JSISET . GT . 10 )       GO TO 5020
C-----SET PARAMETERS FOR CHECKING LEFT TURN PRIMARY AND PROCESS THE
C-----SIGNAL INDICATION BY THE SIGNAL SETTING NUMBER
      KSISET = JSISET - 4
      JTURN = 1
      GO TO 5040
 5020 CONTINUE
                        IF ( JSISET . GT . 16 )       GO TO 5030
C-----SET PARAMETERS FOR CHECKING STRAIGHT PRIMARY AND PROCESS THE
C-----SIGNAL INDICATION BY THE SIGNAL SETTING NUMBER
      KSISET = JSISET - 10
      JTURN = 2
      GO TO 5040
 5030 CONTINUE
                        IF ( JSISET . GT . 22 )       GO TO 5060
C-----SET PARAMETERS FOR CHECKING RIGHT TURN PRIMARY AND PROCESS THE
C-----SIGNAL INDICATION BY THE SIGNAL SETTING NUMBER
      KSISET = JSISET - 16
      JTURN = 3
 5040 CONTINUE
C-----IF THE TURN CODE FOR THE VEHICLE IS NE THE PRIMARY TURN CODE THEN
C-----GO TO 5050 AND PROCESS FOR THE OTHER TURN CODE ELSE PROCESS FOR
C-----THE PRIMARY TURN CODE
                        IF ( ITURN . NE . JTURN )     GO TO 5050
C-----PROCESS FOR THE PRIMARY TURN CODE (FIRST CHARACTER IN SET OF 2)
C-----            GA    GR    AG    AR    RG    RA
      GO TO ( 1010,1010,2010,2010,3010,3010 ) , KSISET
 5050 CONTINUE
C-----PROCESS FOR THE OTHER TURN CODE (SECOND CHARACTER IN SET OF 2)
C-----            GA    GR    AG    AR    RG    RA
      GO TO ( 2010,3010,1010,3010,1010,2010 ) , KSISET
 5060 CONTINUE
                        IF ( JSISET . GT . 25 )       GO TO 9120
C-----CHECK FOR PROTECTED GREEN THUS IF THIS VEHICLE IS TURNING LEFT
C-----THEN GO TO 4010 AND PROCESS PROTECTED GREEN
                        IF ( ITURN . EQ . 1 )         GO TO 4010
C-----SET PARAMETERS FOR CHECKING PROTECTED GREEN
      KSISET = JSISET - 22
C-----PROCESS FOR THE OTHER TURN CODE (SECOND CHARACTER IN SET OF 2)
C-----            PG    PA    PR
      GO TO ( 1010,2010,3010 ) , KSISET
 6010 CONTINUE
C-----FINISH PROCESSING ALL SIGNAL INDICATIONS
      IREPIL = LTRUE
                        IF ( MPRO . EQ . LFALSE )     GO TO 6020
C-----SET CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
      CALL  SETCON
      RETURN
 6020 CONTINUE
C-----UNSET THE CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
      CALL  UNSETC
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9120 CONTINUE
      CALL  ABORTR  ( MSG912,30 )
      STOP  912
      END                                                          SIGRES
```

```
      SUBROUTINE LSTOP                                                  COLEASE
C     TASK,LSTOP
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / PATH   / LENP    ,IOPT    ,LIBL    ,LOBL    ,            COLEASE
     *                  IFVP    ,ILVP    ,LIMP    ,IPT     ,            COLEASE
     *                  NGEOCP  ,NCPSET  ,ICPSET(60),LOBAP ,            COLEASE
     *                  ILCH    ,IGEOCP(60)                             COLEASE
      COMMON / VEHD   / ISLP    ,IACC    ,IVEL    ,IPOS    ,            COLEASE
     *                  ISET    ,LCHGE   ,ISPDP   ,LEGAL   ,            COLEASE
     *                  IPRTM   ,ITIMV   ,IQDS    ,ISPDS   ,            COLEASE
     *                  ISDS    ,IDVS    ,ISTCON  ,IVMAXA  ,            COLEASE
     *                  IVMAXD  ,LATPOS  ,IDTS    ,LALT    ,            COLEASE
     *                  NORC    ,LOGFLG  ,MSTPF   ,MLAG    ,            COLEASE
     *                  MTCARS  ,MFINL   ,MSFLG   ,MPOBS   ,            COLEASE
     *                  MOASF   ,MSAOR   ,MPRO    ,MBLOCK  ,            COLEASE
     *                  MININT  ,IFVA    ,IACDS   ,ICDFS   ,            COLEASE
     *                  ISDEC   ,ISTMO   ,IACLDS  ,IRSTOP               COLEASE
      COMMON / VEHF   / IDRICL  ,IVEHCL  ,ISPD    ,NOF     ,            COLEASE
     *                  NOR     ,LNEXT   ,LPRES   ,ITURN   ,            COLEASE
     *                  IBAPS   ,IPRTLO  ,IEXTIM  ,NOBAPD               COLEASE
      COMMON / VEHIL  / MDEDIC  ,MINFLZ  ,MLUNC   ,MIUNC   ,            COLEASE
     *                  MLYELD  ,MLSTOP  ,MATSTL  ,MSSRED  ,            COLEASE
     *                  MLRTOR  ,MSSGRN  ,MCHKCF  ,MDUMIL  ,            COLEASE
     *                  IDEDIC  ,INFLZ   ,ILUNC   ,ILYELD  ,            COLEASE
     *                  ILSTOP  ,ICONTN  ,ICHKCF  ,IERROR               COLEASE
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DATA      N1,N2   / 4HLSTO,2HP  /
C
C-----SUBROUTINE LSTOP CHECKS TO SEE IF THE VEHICLE MAY ENTER THE
C-----INTERSECTION WITHOUT BLOCKING ANY VEHICLE STOPPED AT THE
C-----INTERSECTION BEFORE THIS VEHICLE AND IF OK THEN CHECKS SIGHT
C-----DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECKS INTERSECTIONS
C-----CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED INTO THE
C-----INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                     IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE INTERSECTION CONTROL LOGIC TIMER FOR PROCESSING NEXT DT
      LOGTMP = 2
C-----IF THERE ARE NO VEHICLES AT THE INTERSECTION THEN GO TO 2010 AND
C-----CHECK SIGHT DISTANCE RESTRICTIONS AND INTERSECTION CONFLICTS
                     IF ( NVATIN . EQ . 0 )        GO TO 2010
                     IF ( IPTHUP . EQ . LNEXT )    GO TO 1010
C     COLEASE,EXTRAC,PATH,LNEXT
      CALL  EXTRAC  (      4,LNEXT )                                    COLEASE
      IPTHUP = LNEXT
 1010 CONTINUE
C-----IF THE VEHICLES INTERSECTION PATH DOES NOT HAVE ANY GEOMETRIC
C-----CONFLICTS THEN GO TO 2010 AND CHECK SIGHT DISTANCE RESTRICTIONS
                     IF ( NGEOCP . LE . 0 )        GO TO 2010
C-----CHECK EACH VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION
C-----(UNTIL MYSELF) TO SEE IF THIS VEHICLE MAY ENTER THE INTERSECTION
C-----WITHOUT BLOCKING ANY VEHICLE STOPPED AT THE INTERSECTION BEFORE
C-----THIS VEHICLE
      DO 1030  IVATIN = 1 , NVATIN
      JV = LVATIN(IVATIN)
C-----IF THE NEXT VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION
C-----IS ME THEN GO TO 2010 AND CHECK SIGHT DISTANCE RESTRICTIONS AND
```

```
C-----INTERSECTION CONFLICTS
                     IF ( IV . EQ . JV )            GO TO 2010
C     COLEASE,FIND,NPRO,VEHD,JV,MPRO
      CALL  FIND     (NPRO  ,      6,JV     ,      31)                  COLEASE
                     IF ( NPRO . EQ . LTRUE )       GO TO 1030
C     COLEASE,FIND,NLUNC,VEHIL,JV,MLUNC
      CALL  FIND     (NLUNC ,      8,JV     ,       3)                  COLEASE
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION IS TO
C-----FOLLOW THE UNCONTROLLED LANE LOGIC AND THE INTERSECTION IS NOT
C-----UNCONTROLLED THEN THIS VEHICLE MAY NOT BLOCK HIM THUS RETURN
                IF ( NLUNC.EQ.LTRUE.AND.ICONTR.GT.1 )RETURN
C     COLEASE,FIND,MNEXT,VEHF,JV,LNEXT
      CALL  FIND     (MNEXT ,      7,JV     ,       6)                  COLEASE
C     COLEASE,FIND,MCPSET,PATH,MNEXT,NCPSET
      CALL  FIND     (MCPSET,      4,MNEXT  ,      10)                  COLEASE
C     COLEASE,FIND,MOGFLG,VEHD,JV,LOGFLG
      CALL  FIND     (MOGFLG,      6,JV     ,      22)                  COLEASE
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION HAS
C-----INTERSECTION CONFLICTS SET AGAINST HIM AND HE IS NOT GOING TO
C-----WAKE UP FOR INTERSECTION CONTROL LOGIC WITHIN THE NEXT DT THEN
C-----GO TO 1030 AND CHECK THE NEXT VEHICLE ON THE LIST OF VEHICLES AT
C-----THE INTERSECTION (THIS VEHICLE MAY BLOCK HIM)
      IF ( MCPSET.GT.0 . AND . MOGFLG.GT.2 )       GO TO 1030
C-----CHECK EACH OF MY INTERSECTION CONFLICTS AND SEE IF THE VEHICLE ON
C-----THE LIST OF VEHICLES AT THE INTERSECTION IS ON AN INTERSECTION
C-----PATH THAT CONFLICTS WITH MY INTERSECTION PATH
      DO 1020  INDEX = 1 , NGEOCP
      JNDEX = IGEOCP(INDEX)
C     COLEASE,FIND,ICONP1,CONFLT,JNDEX,ICONP(1)
      CALL  FIND     (ICONP1,      2,JNDEX  ,       1)                  COLEASE
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION IS ON
C-----AN INTERSECTION PATH THAT CONFLICTS WITH MY INTERSECTION PATH THEN
C-----THIS VEHICLE MAY NOT BLOCK HIM THUS RETURN
                     IF ( ICONP1 . EQ . MNEXT )   RETURN
C     COLEASE,FIND,ICONP2,CONFLT,JNDEX,ICONP(2)
      CALL  FIND     (ICONP2,      2,JNDEX  ,       2)                  COLEASE
C-----IF THE VEHICLE ON THE LIST OF VEHICLES AT THE INTERSECTION IS ON
C-----AN INTERSECTION PATH THAT CONFLICTS WITH MY INTERSECTION PATH THEN
C-----THIS VEHICLE MAY NOT BLOCK HIM THUS RETURN
                     IF ( ICONP2 . EQ . MNEXT )   RETURN
C-----END OF INTERSECTION CONFLICT LOOP
 1020 CONTINUE
C-----END OF LIST OF VEHICLES AT THE INTERSECTION LOOP
 1030 CONTINUE
C-----THIS VEHICLE MAY ENTER THE INTERSECTION WITHOUT BLOCKING ANY
C-----VEHICLE STOPPED AT THE INTERSECTION BEFORE THIS VEHICLE
 2010 CONTINUE
C-----CHECK SIGHT DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECK
C-----INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED
C-----INTO THE INTERSECTION
      CALL  CHKSDR
C-----IF THE VEHICLE HAS A SIGHT DISTANCE RESTRICTION OR AN INTERSECTION
C-----CONFLICT AND MAY NOT PROCEED INTO THE INTERSECTION THEN RETURN
C-----ELSE THE VEHICLE MAY PROCEED INTO THE INTERSECTION
                     IF ( MPRO . EQ . LFALSE )    RETURN
C3    KPFLAG = 10HI MAY ENTR
C-----IF THE VEHICLE IS NOT STOPPED AT THE STOP LINE THEN RETURN ELSE
C-----SET THE VEHICLES ACC/DEC LOGIC TIMER FOR HESITATION
                     IF ( MATSTL . EQ . LFALSE )  RETURN
      THES = 3.0*PIJR(IDRICL) + (PIJR(IDRICL)+1.0)*MIN1(NVATIN/6.0,1.5)
      IPRTM = MIN1(THES/DT+0.5,6.0/DT,15.0)
      JPRTM = IPRTM
      RETURN
      END                                                               LSTOP
```

```
      SUBROUTINE CHKSDR                                                 COLEASE
C     TASK,CHKSDR
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(15),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / ACM,DCH,DCM,DVM,ERRJUD,I,INDEX,IPNDEX,ISDR,J,JA,
     *                  JCANSE,JL,JNDEX,JP,JSDRA(5),JVEL,KCANSE,KSPD,
     *                  MAXLOG,MSDR,POSCHK,TCH,TCM,TFZ,TIM,TIMEND,
     *                  TPASSM,VCM,VCHKCO(39),AD,JD,JSLIM,JSPD,JSPDP,JV,
     *                  LGEOM4,MIMP,P,PO,SO,VO,VPREDT(21),ZTEMPD(5)
      DATA      N1,N2  / 4HCHKS,4HDR    /
C4701 FORMAT(8H VEHICLEI4,3H IS F7.2,29H SEC FROM THE END OF HIS LANE)
C4702 FORMAT(11H CHKSDR CON,I5,9H APPROACH,I3,4H PO=,F7.1,5H DCH=,
C4   *        F7.1,5H TCH=,F7.2,5H TFZ=,F7.2,5H DCM=,F7.1,5H TCM=,
C4   *        F7.2,5H VCM=,F5.1)
C
C-----SUBROUTINE CHKSDR CHECKS SIGHT DISTANCE RESTRICTIONS AND IF CLEAR
C-----THEN CHECKS INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE
C-----MAY PROCEED INTO THE INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR CHKSDR
      MPRO = LFALSE
      LOGTMP = 2
C-----IF THE VEHICLE IS NOT DEDICATED TO AN INTERSECTION PATH THEN
C-----RETURN
                    IF ( LNEXT . EQ . 0 )        RETURN
C-----IF THE LANE IS NOT UNCONTROLLED OR YIELD SIGN CONTROLLED THEN GO
C-----TO 3010 AND CHECK INTERSECTION CONFLICTS (NO SIGHT DISTANCE
C-----RESTRICTIONS FOR STOP SIGN CONTROLLED OR SIGNAL CONTROLLED)
                    IF ( LCONTR . GE . 4 )       GO TO 3010
C-----THE LANE IS UNCONTROLLED OR YIELD SIGN CONTROLLED THUS IF
C-----THERE ARE VEHICLES STOPPED AT THE INTERSECTION WAITING TO ENTER
C-----AND THIS VEHICLE IS NOT STOPPED AT THE STOP LINE AND THE
C-----INTERSECTION IS UNCONTROLLED THEN RETURN AND WAIT UNTIL THE
C-----VEHICLE IS STOPPED AT THE STOP LINE OR THERE ARE NO VEHICLES
C-----WAITING TO ENTER
      IF ( NVATIN.GT.0 . AND . MATSTL.EQ.LFALSE . AND . MIUNC.EQ.LTRUE )
     *                                           RETURN
C-----IF THERE ARE NO SIGHT DISTANCE RESTRICTIONS FOR THIS APPROACH THEN
C-----GO TO 3010 AND CHECK INTERSECTION CONFLICTS
                    IF ( NSDR   . LE . 0 )       GO TO 3010
C-----IF THE VEHICLES LANE IS UNCONTROLLED WHILE THE INTERSECTION IS
C-----NOT UNCONTROLLED (YIELD SIGN CONTROLLED) THEN THERE ARE NO SIGHT
C-----DISTANCE RESTRICTIONS THUS GO TO 3010 AND CHECK INTERSECTION
C-----CONFLICTS
      IF ( MLUNC.EQ.LTRUE.AND.MIUNC.EQ.LFALSE )  GO TO 3010
C-----IF THE VEHICLE IS STOPPED AT THE STOP LINE THEN THERE ARE NO SIGHT
C-----DISTANCE RESTRICTIONS THUS GO TO 3010 AND CHECK INTERSECTION
C-----CONFLICTS
                    IF ( MATSTL . EQ . LTRUE )   GO TO 3010
                    IF ( IPTHUP . EQ . LNEXT )   GO TO 1010
C     COLEASE,EXTRAC,PATH,LNEXT
      CALL  EXTRAC  (      4,LNEXT )                                    COLEASE
      IPTHUP = LNEXT
 1010 CONTINUE
C-----IF THE VEHICLES INTERSECTION PATH DOES NOT HAVE INTERSECTION
C-----CONFLICTS THEN THERE ARE NO SIGHT DISTANCE RESTRICTIONS THUS GO TO
C-----3010 AND CHECK INTERSECTION CONFLICTS
                    IF ( NGEOCP . LE . 0 )       GO TO 3010
                    IF ( ILVP   . EQ . 0 )       GO TO 1020
C     COLEASE,FIND,JVEL,VEHD,ILVP,IVEL
      CALL  FIND    (JVEL  ,      6,ILVP  ,      3)                     COLEASE
C-----IF THE LAST VEHICLE ON THIS VEHICLES INTERSECTION PATH IS STOPPED
C-----THEN RETURN AND WAIT UNTIL IT IS MOVING
                    IF ( JVEL . LE . 25 )        RETURN
 1020 CONTINUE
C-----SET THE MAXIMUM TIME FROM THE END OF THE LANE THAT THIS VEHICLE
C-----MAY DECIDE TO PROCEED IF THE SIGHT DISTANCE RESTRICTIONS ARE CLEAR
      TIM = 3.0
                    IF ( MLUNC . EQ . LTRUE )    TIM=TIM+TLEAD+APIJR+2.0
                    IF ( MIUNC . EQ . LTRUE )    TIM = 2.0
C-----SET THIS VEHICLES PARAMETERS FOR PREDICTING TIME AND VELOCITY TO
C-----AN INTERSECTION CONFLICT
      CALL  SETPTV
C-----SET THE POSITION OF THE CONFLICT AS THE END OF THE LANE
      P = LGEOM4
C-----PREDICT THE TIME AND VELOCITY TO THE END OF THE LANE
      CALL  PREDTV  ( TCM,VCM,ACM )
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 101
C4                  IF ( TIME . LT . TPRINT )    GO TO 101
C4    PRINT 701 , IV,TCM
C4101 CONTINUE
C-----IF THE TIME TO THE END OF THE LANE IS GT THE MAXIMUM TIME FROM
C-----THE END OF THE LANE THAT THIS VEHICLE MAY DECIDE TO PROCEED IF
C-----THE SIGHT DISTANCE RESTRICTIONS ARE CLEAR THEN GO TO 4010 AND SET
C-----THE WAKE UP TIME
                    IF ( TCM . GT . TIM )        GO TO 4010
C-----SET EACH APPROACH THAT THIS APPROACH HAS A SIGHT DISTANCE
C-----RESTRICTION WITH TO NOT CHECKED
      DO 1030  I = 1 , NSDR
      JSDRA(I) = LFALSE
 1030 CONTINUE
      MSDR = 0
C-----PROCESS THE INTERSECTION CONFLICTS FROM LAST TO FIRST
      DO 2040  I = 1 , NGEOCP
```

```
C-----IF EACH APPROACH THAT THIS APPROACH HAS A SIGHT DISTANCE
C-----RESTRICTION WITH HAS BEEN CHECKED THEN GO TO 3010 AND CHECK
C-----INTERSECTION CONFLICTS (SIGHT DISTANCE RESTRICTIONS CLEAR)
                    IF ( MSDR . EQ . NSDR )      GO TO 3010
      INDEX = NGEOCP - I + 1
      JNDEX = IGEOCP(INDEX)
                    IF ( ICONUP . EQ . JNDEX )   GO TO 1040
C     COLEASE,EXTRAC,CONFLT,JNDEX
      CALL  EXTRAC  (      2,JNDEX )                              COLEASE
      ICONUP = JNDEX
 1040 CONTINUE
C-----FIND THE LINKING INBOUND APPROACH NUMBER FOR THE CONFLICTING PATH
      J = 1
                    IF ( LNEXT . EQ . ICONP(1) ) J = 2
      JA = ICONA(J)
C-----CHECK EACH APPROACH THAT THIS APPROACH HAS A SIGHT DISTANCE
C-----RESTRICTION WITH
      DO 1050  ISDR = 1 , NSDR
C-----IF THE INTERSECTION PATH CAME FROM AN APPROACH THAT HAS A SIGHT
C-----DISTANCE RESTRICTION WITH US THEN GO TO 1060 AND CONTINUE
                    IF ( ISDRA(ISDR) . EQ . JA ) GO TO 1060
 1050 CONTINUE
C-----THE INTERSECTION PATH DID NOT COME FROM AN APPROACH THAT HAS A
C-----SIGHT DISTANCE RESTRICTION WITH US THUS SKIP TO THE NEXT
C-----INTERSECTION CONFLICT
      GO TO 2040
 1060 CONTINUE
C-----IF THE LINKING INBOUND APPROACH THAT THE INTERSECTION PATH CAME
C-----FROM HAS ALREADY BEEN CHECKED THEN SKIP TO THE NEXT INTERSECTION
C-----CONFLICT
                    IF ( JSDRA(ISDR).EQ.LTRUE )  GO TO 2040
C-----SET THE PARAMETERS FOR CHECKING SIGHT DISTANCE RESTRICTIONS
      JP = ICONP(J)
C-----SET THIS VEHICLES PARAMETERS FOR PREDICTING TIME AND VELOCITY TO
C-----AN INTERSECTION CONFLICT
      CALL  SETPTV
      P = ICOND(3-J) + LGEOM4
                    IF ( IFVA . EQ . LFALSE )    GO TO 1070
                    IF ( IVPV . EQ . 0 )         GO TO 1070
C     COLEASE,FIND,KSPD,VEHF,IVPV,ISPD
      CALL  FIND    (KSPD  ,      7,IVPV   ,     3)               COLEASE
C-----THIS VEHICLES ACC/DEC LOGIC SAYS TO FOLLOW THE VEHICLE AHEAD THUS
C-----MIN THE DESIRED SPEED WITH THE DESIRED SPEED OF THE VEHICLE AHEAD
      JSPD = MIN0(KSPD,JSPD)
 1070 CONTINUE
C4    DVM = JSPD
C4    DCM = P - PO
C-----IF THERE IS NO DISTANCE TO TRAVEL THEN GO TO 2010 AND FIND THE
C-----TIME TO THE INTERSECTION CONFLICT FOR ME
                    IF ( P-PO . LE . 0.0 )       GO TO 2010
C-----PREDICT THE TIME AND VELOCITY TO AN INTERSECTION CONFLICT
      CALL  PREDTV  ( TCM,VCM,ACM )
      GO TO 2020
 2010 CONTINUE
C-----THERE WAS NO DISTANCE TO TRAVEL THUS SET THE TIME TO THE CONFLICT
C-----FOR ME TO 0 AND THE VELOCITY AT THE CONFLICT TO MY CURRENT SPEED
      TCM = 0.0
      VCM = VO
                    IF ( VCM . LE . 0.0 )        GO TO 2020
C-----THE CURRENT SPEED IS GT 0 THUS COMPUTE THE TIME TO THE CONFLICT
      TCM = (P-PO)/VCM
 2020 CONTINUE
      TPASSM = 1.0E09
                    IF ( VCM . LE . 0 )          GO TO 2030
C-----FIND THE TIME FOR MY VEHICLE TO PASS THE CONFLICT AT THE VELOCITY
C-----AT THE CONFLICT FOR ME
      TPASSM = LENV(IVEHCL)/VCM
 2030 CONTINUE
C-----SET UP AN ARTIFICIAL VEHICLE ON THE OTHER APPROACH
C     COLEASE,FIND,JSLIM,APPRO,JA,ISLIM
      CALL  FIND    (JSLIM ,      1,JA     ,    14)               COLEASE
C-----THE VELOCITY OF THE ARTIFICIAL VEHICLE WILL BE THE SPEED LIMIT OF
C-----THE OTHER APPROACH
      VO = JSLIM
      IPNDEX = POSNEW/25.0 + 1
C     COLEASE,FIND,JCANSE,SDR,ISDR,ICANSE(IPNDEX)
      CALL  FIND    (JCANSE,      5,ISDR   ,     0+IPNDEX)        COLEASE
C-----THE POSITION OF THE ARTIFICIAL VEHICLE WILL BE AT THE POINT JUST
C-----VISIBLE BY THIS VEHICLE AROUND THE SIGHT DISTANCE RESTRICTION
      PO = JCANSE
C     COLEASE,FIND,JL,PATH,JP,LIBL
      CALL  FIND    (JL    ,      4,JP     ,     3)               COLEASE
C     COLEASE,FIND,LGEOM4,LANE,JL,LGEOM(4)
      CALL  FIND    (LGEOM4,      3,JL     ,    20)               COLEASE
C-----THE POSITION THE ARTIFICIAL VEHICLE HAS TO TRAVEL TO IS THE
C-----INTERSECTION CONFLICT
      P = ICOND(J) + LGEOM4
C4    DCH = P - PO
C-----COMPUTE THE TIME TO THE CONFLICT FOR HIM BASED ON THE DISTANCE HE
C-----HAS TO TRAVEL AND A CONSTANT SPEED (SPEED LIMIT FOR THE APPROACH)
      TCH = (P-PO)/VO
C-----FIND THE ERROR IN JUDGEMENT
      ERRJUD = AMAX1(0.0,PIJR(IDRICL)*(TCH-5.0)/5.0)
C-----FIND THE TIME THAT HIS FRONT ZONE WILL ARRIVE AT THE CONFLICT
      TFZ = TCH - TPASSM - TLEAD - PIJR(IDRICL) - ERRJUD/2.0
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 102
C4                  IF ( TIME . LT . TPRINT )    GO TO 102
C4    PRINT 702 , JNDEX,JA,PO,DCH,TCH,TFZ,DCM,TCM,VCM
C4102 CONTINUE
C-----IF THE TIME TO THE CONFLICT FOR ME IS GT THE TIME HIS FRONT ZONE
C-----WILL ARRIVE AT THE CONFLICT THEN I AM BLOCKED BY HIM THUS GO TO
C-----5010 AND SET THE WAKE UP TIME
                    IF ( TCM . GT . TFZ )        GO TO 5010
C-----SET THE OTHER APPROACH CHECKED
      JSDRA(ISDR) = LTRUE
      MSDR = MSDR + 1
C-----END OF INTERSECTION CONFLICT LOOP
 2040 CONTINUE
C-----ALL SIGHT DISTANCE RESTRICTIONS ARE CLEAR
 3010 CONTINUE
C-----CHECK INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY
C-----PROCEED INTO THE INTERSECTION
      CALL  CHKCON
      RETURN
 4010 CONTINUE
C-----THE TIME TO THE CONFLICT IS GT THE MAXIMUM TIME FROM THE END OF
C-----THE LANE THAT THIS VEHICLE MAY DECIDE TO PROCEED IF THE SIGHT
C-----DISTANCE RESTRICTIONS ARE CLEAR THUS SET THE INTERSECTION CONTROL
C-----LOGIC TIMER TO WAKE UP WHEN CLOSER
      LOGTMP = MAX0(2,MIN1(2+5.0/DT,15.0,2.0+(TCM-TIMEND-DT)/DT))
      RETURN
 5010 CONTINUE
C-----THE TIME TO THE CONFLICT FOR ME IS GT THE TIME HIS FRONT ZONE
C-----WILL ARRIVE AT THE CONFLICT THUS I AM BLOCKED BY HIM THUS SET THE
C-----WAKE UP TIME
                    IF ( VELNEW . EQ . 0.0 )     RETURN
      MAXLOG = MIN1(2.0+5.0/DT,15.0)
      POSCHK = POSNEW
C-----FIND THE NUMBER OF DTS UNTIL I AM CLOSE ENOUGH TO GO IN FRONT OF
C-----THE FRONT ZONE OF THE ARTIFICIAL VEHICLE
      DO 5020  LOGTMP = 2 , MAXLOG
      TCM = TCM - DT
      POSCHK = POSCHK + DT*VELNEW
      IPNDEX = MIN0(IFIX(POSCHK/25.0)+1,40)
C     COLEASE,FIND,KCANSE,SDR,ISDR,ICANSE(IPNDEX)
      CALL  FIND    (KCANSE,      5,ISDR   ,     0+IPNDEX)        COLEASE
      TFZ = TFZ + (JCANSE-KCANSE)/VO
                    IF ( TCM . LE . TFZ )        RETURN
      JCANSE = KCANSE
 5020 CONTINUE
      LOGTMP = MAXLOG
      RETURN
```

END

CHKSDR

```
      SUBROUTINE CHKCON                                                 COLEASE
C   TASK,CHKCON
      COMMON / CONFLT / ICONP ( 2),ICUNA ( 2),ICOND ( 2),ICUNAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLUS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VCHKSD(33),ACH,ACM,DCH,DCM,DVH,DVM,ERRJUD,I,
     *                  INDEX,IVCONF,J,JACC,JFVA,JL,JNDEX,JP,JPOS,JSLP,
     *                  JSNA,JVEL,KOUNT,KPRTM,KSPD,MGEOM4,MOR,MORC,
     *                  NININT,NOFC,SLOPE,TCH,TCM,TCRASH,TFZ,TIM,TPASSH,
     *                  TPASSM,TRZ,VCH,VCM,AO,JD,JSLIM,JSPD,JSPDP,JV,
     *                  LGEOM4,MIMP,P,PO,SO,VO,VPREDT(21),ZTEMPD(5)
      DIMENSION         IENT6(1),MSG913(6)
      EQUIVALENCE       (IFVA,IENT6(1))
      DATA      MSG913 / 4H INF,4HINIT,4HE LO,4HOP -,4H CHK,4HCON  /
      DATA      N1,N2  / 4HCHKC,2HON /
      DATA      RADIAN / 0.0174532925199 /
C4701 FORMAT(8H VEHICLEI4,3H ISF7.2,29H SEC FROM THE END OF HIS LANE)
C4702 FORMAT(4H CONI4,4H VEHI4,5H TCM=F6.2,5H VCM=F5.1,5H DVM=F5.1,
C4   *        5H DCM=F6.1,6H   VEHI4,5H TFZ=F6.2,5H TCH=F6.2,5H TRZ=F6.2,
C4   *        5H VCH=F5.1,5H DVH=F5.1,5H DCH=F6.1)
C
C-----SUBROUTINE CHKCON CHECKS INTERSECTION CONFLICTS AND IF CLEAR THEN
C-----THE VEHICLE MAY PROCEED INTO THE INTERSECTION
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                    IF ( IPTHUP . EQ . LNEXT )   GO TO 1010
C   COLEASE,EXTRAC,PATH,LNEXT
```

```
      CALL  EXTRAC  (      4,LNEXT )                                    COLEASE
      IPTHUP = LNEXT
 1010 CONTINUE
C-----IF THERE ARE NO GEOMETRIC CONFLICTING PATHS THEN GO TO 3020 AND
C-----THE VEHICLE MAY PROCEED INTO THE INTERSECTION
                    IF ( NGEOCP . LE . 0 )          GO TO 3020
                    IF ( MATSTL . EQ . LTRUE )      GO TO 1020
C-----SET THE MAXIMUM TIME FROM THE END OF THE LANE THAT THIS VEHICLE
C-----MAY DECIDE TO PROCEED IF THE INTERSECTION CONFLICTS ARE CLEAR
      TIM = 3.0
                    IF ( MLUNC . EQ . LTRUE )       TIM=TIM+TLEAD+APIJR+2.0
                    IF ( MIUNC . EQ . LTRUE )       TIM = 2.0
C-----SET THIS VEHICLES PARAMETERS FOR PREDICTING TIME AND VELOCITY TO
C-----AN INTERSECTION CONFLICT
      CALL  SETPTV
C-----SET THE POSITION OF THE CONFLICT AS THE END OF THE LANE
      P = LGEOM4
C-----PREDICT THE TIME AND VELOCITY TO THE END OF THE LANE
      CALL  PREDTV  ( TCM,VCM,ACM )
C5                  IF ( IPRTLO . EQ . 0 )          GO TO 101
C4                  IF ( TIME . LT . TPRINT )       GO TO 101
C4    PRINT 701 , IV,TCM
C4101 CONTINUE
C-----IF THE TIME TO THE END OF THE LANE IS GT THE MAXIMUM TIME FROM
C-----THE END OF THE LANE THAT THIS VEHICLE MAY DECIDE TO PROCEED IF
C-----THE INTERSECTION CONFLICTS ARE CLEAR THEN GO TO 4010 AND SET THE
C-----WAKE UP TIME
                    IF ( TCM . GT . TIM )           GO TO 4010
 1020 CONTINUE
C-----IF THERE ARE NO INTERSECTION CONFLICTS SET THEN GO TO 3020 AND THE
C-----VEHICLE MAY PROCEED INTO THE INTERSECTION
                    IF ( NCPSET . LE . 0 )          GO TO 3020
C-----CHECK EACH GEOMETRIC CONFLICTING INTERSECTION PATH
      DO 3010  INDEX = 1 , NGEOCP
C-----IF THE INTERSECTION CONFLICT IS NOT SET THEN SKIP TO THE NEXT
C-----INTERSECTION CONFLICT
                    IF ( ICPSET(INDEX).EQ.0 )       GO TO 3010
C-----INITIALIZE SOME PARAMETERS FOR CHKCON
      JNDEX = IGEOCP(INDEX)
      KOUNT = 0
                    IF ( ICONUP . EQ . JNDEX )      GO TO 1030
C   COLEASE,EXTRAC,CONFLT,JNDEX
      CALL  EXTRAC  (      2,JNDEX )                                    COLEASE
      ICONUP = JNDEX
 1030 CONTINUE
      J = 1
                    IF ( LNEXT . EQ . ICONP(1) ) J = 2
C-----SET IVCONF TO THE NEXT VEHICLE THAT HAS NOT CLEARED THE
C-----INTERSECTION CONFLICT
      IVCONF = ICONV(J)
      I = 3 - J
      JP = ICONP(J)
C   COLEASE,FIND,JL,PATH,JP,LIBL
      CALL  FIND     (JL     ,      4,JP     ,      3)                  COLEASE
C   COLEASE,FIND,MGEOM4,LANE,JL,LGEOM(4)
      CALL  FIND     (MGEOM4,       3,JL     ,     20)                  COLEASE
      TCM = 0.0
C-----SET NOFC TO THE IVCONF VEHICLE
      NOFC = IVCONF
C   COLEASE,FIND,NININT,VEHD,NOFC,MININT
      CALL  FIND     (NININT,       6,NOFC   ,     33)                  COLEASE
                    IF ( NININT . EQ . LTRUE )      GO TO 1040
C-----THE NOFC VEHICLE WAS NOT IN THE INTERSECTION THUS SET THE NOFC
C-----VEHICLE TO THE FIRST VEHICLE IN THE OTHER LANE
C   COLEASE,FIND,NOFC,LANE,JL,IFVL
      CALL  FIND     (NOFC   ,      3,JL     ,     13)                  COLEASE
 1040 CONTINUE
C-----SET THIS VEHICLES PARAMETERS FOR PREDICTING TIME AND VELOCITY TO
C-----AN INTERSECTION CONFLICT
      CALL  SETPTV
      P = ICOND(I) + LGEOM4
                    IF ( IFVA . EQ . LFALSE )       GO TO 1050
                    IF ( IVPV . EQ . 0 )            GO TO 1050
C   COLEASE,FIND,KSPD,VEHF,IVPV,ISPD
      CALL  FIND     (KSPD   ,      7,IVPV   ,      3)                  COLEASE
C-----THIS VEHICLES ACC/DEC LOGIC SAYS TO FOLLOW THE VEHICLE AHEAD THUS
C-----MIN THE DESIRED SPEED WITH THE DESIRED SPEED OF THE VEHICLE AHEAD
      JSPD = MIN0(KSPD,JSPD)
 1050 CONTINUE
C4    DVM = JSPD
C4    DCM = P - PO
C-----IF THERE IS NO DISTANCE TO TRAVEL TO THE INTERSECTION CONFLICT
C-----THEN GO TO 1070 AND FIND THE TIME TO THE INTERSECTION CONFLICT FOR
C-----ME
                    IF ( P-PO . LE . 0.0 )          GO TO 1070
                    IF ( ILVP . EQ . 0 )            GO TO 1060
C   COLEASE,FIND,JVEL,VEHD,ILVP,IVEL
      CALL  FIND     (JVEL   ,      6,ILVP   ,      3)                  COLEASE
C-----IF THE LAST VEHICLE ON THE INTERSECTION PATH IS STOPPED THEN
C-----RETURN AND WAIT UNTIL HE IS MOVING OUT
                    IF ( JVEL . LE . 25 )           RETURN
 1060 CONTINUE
C-----PREDICT THE TIME AND VELOCITY TO AN INTERSECTION CONFLICT
      CALL  PREDTV  ( TCM,VCM,ACM )
                    IF ( LCONTR . GT . 4 )          GO TO 1080
                    IF ( MATSTL . EQ . LFALSE )     GO TO 1080
C-----THE LANE IS NOT SIGNAL CONTROLLED AND THE VEHICLE IS STOPPED AT
C-----THE STOP LINE THUS INCREMENT THE TIME TO THE CONFLICT FOR ME BY
C-----THE AVERAGE HESITATION TIME
      TCM = TCM + 4.0*PIJR(IDRICL)
      GO TO 1080
 1070 CONTINUE
C-----THERE IS NO DISTANCE TO TRAVEL TO THE INTERSECTION CONFLICT THUS
C-----FIND THE TIME TO THE INTERSECTION CONFLICT FOR ME
      TCM = 0.0
      ACM = AO
      VCM = VO
                    IF ( VCM . LE . 0.0 )           GO TO 1080
      TCM = (P-PO)/VCM
 1080 CONTINUE
C-----FIND THE TIME FOR MY VEHICLE TO PASS THE INTERSECTION CONFLICT AT
C-----THE VELOCITY AT THE INTERSECTION CONFLICT FOR ME
      TPASSM = 1.0E09
                    IF ( VCM . LE . 0.0 )           GO TO 1090
      TPASSM = LENV(IVEHCL)/VCM
 1090 CONTINUE
C-----START OF LOOP FOR CHECKING FOR THIS INTERSECTION CONFLICT
      KOUNT = KOUNT + 1
                    IF ( KOUNT . GT . 50 )          GO TO 9130
C-----IF THE NOFC VEHICLE IS THE IVCONF VEHICLE THEN GO TO 1100 AND
C-----CHECK THE IVCONF VEHICLE
                    IF ( NOFC . EQ . IVCONF )       GO TO 1100
C   COLEASE,FIND,MORC,VEHD,NOFC,NORC
      CALL  FIND     (MORC   ,      6,NOFC   ,     21)                  COLEASE
C-----IF THE NOFC VEHICLE HAS NOT SET CONFLICTS THEN HE MAY NOT PROCEED
C-----INTO THE INTERSECTION THUS HE WILL BLOCK THE IVCONF VEHICLE FROM
C-----PROCEEDING INTO THE INTERSECTION ALSO THUS THERE CAN BE NO
C-----INTERSECTION CONFLICT WITH THE IVCONF VEHICLE THUS GO TO 3010 AND
C-----SKIP TO THE NEXT INTERSECTION CONFLICT (THIS ONE IS CLEAR)
                    IF ( MORC . EQ . 200+1 )        GO TO 3010
C-----SET THE NOFC VEHICLE TO THE NOR VEHICLE FOR THE CURRENT NOFC
C-----VEHICLE (IBM CAN NOT HAVE THE SAME PARAMETERS IN THE CALL)
C   COLEASE,FIND,MOR,VEHF,NOFC,NOR
      CALL  FIND     (MOR    ,      7,NOFC   ,      5)                  COLEASE
      NOFC = MOR
C-----IF THERE IS A NEW NOFC VEHICLE THEN GO TO 1090 AND CHECK AGAIN
                    IF ( NOFC . NE . 0 )            GO TO 1090
C-----THE OLD NOFC VEHICLE HAD TO BE THE LAST VEHICLE ON THE
C-----INTERSECTION PATH THUS SET THE NOFC VEHICLE TO THE FIRST VEHICLE
C-----ON THE LANE AND GO TO 1090 AND CHECK AGAIN
C   COLEASE,FIND,NOFC,LANE,JL,IFVL
      CALL  FIND     (NOFC   ,      3,JL     ,     13)                  COLEASE
```

```
      GO TO 1090
 1100 CONTINUE
C-----SET THE IVCONF VEHICLES PARAMETERS FOR PREDICTING TIME AND
C-----VELOCITY TO AN INTERSECTION CONFLICT
C    COLEASE,FIND,JSLP,VEHD,IVCONF,ISLP
      CALL  FIND     (JSLP   ,      6,IVCONF,      1)                COLEASE
      SO = JSLP/400.0 - 12.0
C    COLEASE,FIND,JACC,VEHD,IVCONF,IACC
      CALL  FIND     (JACC   ,      6,IVCONF,      2)                COLEASE
      AO = JACC/312.5 - 32.0
C    COLEASE,FIND,JVEL,VEHD,IVCONF,IVEL
      CALL  FIND     (JVEL   ,      6,IVCONF,      3)                COLEASE
      VO = JVEL/25.0
C    COLEASE,FIND,JPOS,VEHD,IVCONF,IPOS
      CALL  FIND     (JPOS   ,      6,IVCONF,      4)                COLEASE
      PO = JPOS/25.0
C    COLEASE,FIND,NININT,VEHD,IVCONF,MININT
      CALL  FIND     (NININT,       6,IVCONF,     33)                COLEASE
      LGEOM4 = MGEOM4
      PO = PO + LGEOM4
C    COLEASE,FIND,JSPD,VEHF,IVCONF,ISPD
      CALL  FIND     (JSPD   ,      7,IVCONF,      3)                COLEASE
      JSPDP = 1
      KPRTM = 0
C-----IF THE IVCONF VEHICLE IS IN THE INTERSECTION THEN GO TO 2040 AND
C-----CONTINUE ELSE SET SOME ADDITIONAL PARAMETERS
                    IF ( NININT . EQ . LTRUE )    GO TO 2040
      PO = PO - LGEOM4
C    COLEASE,FIND,JSPDP,VEHD,IVCONF,ISPDP
      CALL  FIND     (JSPDP ,       6,IVCONF,      7)                COLEASE
C    COLEASE,FIND,JSNA,LANE,JL,ISNA
      CALL  FIND     (JSNA   ,      3,JL     ,     4)                COLEASE
C    COLEASE,FIND,KPRTM,VEHD,IVCONF,IPRTM
      CALL  FIND     (KPRTM ,       6,IVCONF,      9)                COLEASE
C-----IF THE IVCONF VEHICLE HAS ALREADY SET HIS DESIRED SPEED FOR HIS
C-----INTERSECTION PATH THEN GO TO 2010 ELSE GET ADDITIONAL PARAMETERS
                    IF ( JSPDP . NE . 0 )         GO TO 2010
C    COLEASE,FIND,MIMP,PATH,JP,LIMP
      CALL  FIND     (MIMP   ,      4,JP     ,     7)                COLEASE
C    COLEASE,FIND,JSLIM,APPRO,JSNA,ISLIM
      CALL  FIND     (JSLIM ,       1,JSNA   ,    14)                COLEASE
 2010 CONTINUE
C-----CHECKING TO SEE IF IVCONF VEHICLE HAS BEEN PROCESSED THIS DT
C-----(ON AN APPROACH THAT WAS LOWER ON LIBA THAN THE IV VEHICLE)
                    IF ( IAN - LIBAR(JSNA) )      2030 , 2020 , 2040
 2020 CONTINUE
C-----IF THE APPROACH NUMBERS ARE EQUAL CHECK THE LANE NUMBERS
                    IF ( JL . LT . LPRES )        GO TO 2040
 2030 CONTINUE
C-----THE IVCONF VEHICLE HAS NOT BEEN UPDATED THIS DT THUS PREDICT HIS
C-----NEW POS/VEL/ACC
      PO = PO + VO*DT + 0.5*AO*DTSQ + SO*DTCU/6.0
      VO = VO + AO*DT + 0.5*SO*DTSQ
      AO = AO + SO*DT
      KPRTM = MAX0(KPRTM-1,0)
 2040 CONTINUE
C-----FIND ADDITIONAL PARAMETERS FOR THE IVCONF VEHICLE
C    COLEASE,FIND,JD,VEHF,IVCONF,IDRICL
      CALL  FIND     (JD     ,      7,IVCONF,      1)                COLEASE
C    COLEASE,FIND,JV,VEHF,IVCONF,IVEHCL
      CALL  FIND     (JV     ,      7,IVCONF,      2)                COLEASE
      P = ICOND(J) + LGEOM4
C    COLEASE,FIND,JFVA,VEHD,IVCONF,IFVA
      CALL  FIND     (JFVA   ,      6,IVCONF,     34)                COLEASE
C4    DVH = JSPD
C4    DCH = P - PO
C-----IF THERE IS NO DISTANCE TO TRAVEL TO THE INTERSECTION CONFLICT
C-----THEN GO TO 2050 AND FIND THE TIME TO THE INTERSECTION CONFLICT FOR
C-----HIM
                    IF ( P-PO . LE . 0.0 )        GO TO 2050
C-----PREDICT THE TIME AND VELOCITY TO AN INTERSECTION CONFLICT
```

```
      CALL  PREDTV  ( TCH,VCH,ACH )
C-----INCREMENT THE TIME TO THE CONFLICT FOR HIM BY HIS PIJR TIMER
      TCH = TCH + KPRTM*DT
      GO TO 2060
 2050 CONTINUE
C-----THERE IS NO DISTANCE TO TRAVEL TO THE INTERSECTION CONFLICT THUS
C-----FIND THE TIME TO THE INTERSECTION CONFLICT FOR HIM
      TCH = 0.0
      ACH = AO
      VCH = VO
                    IF ( VCH . LE . 0.0 )         GO TO 2060
      TCH = (P-PO)/VCH
 2060 CONTINUE
C-----FIND THE TIME FOR HIS VEHICLE TO PASS THE INTERSECTION CONFLICT AT
C-----THE VELOCITY AT THE INTERSECTION CONFLICT FOR HIM
      TPASSH = 1.0E09
                    IF ( VCH . LE . 0.0 )         GO TO 2070
      TPASSH = LENV(JV)/VCH
 2070 CONTINUE
C-----FIND THE ERROR IN JUDGEMENT
      ERRJUD = AMAX1(0.0,PIJR(IDRICL)*(TCH-5.0)/7.0)
C-----IF THE IVCONF VEHICLES TIME TO THE INTERSECTION CONFLICT IS GT 5
C-----SECONDS AND HE SHOULD FOLLOW THE VEHICLE AHEAD THEN INCREMENT THE
C-----TIME TO THE INTERSECTION CONFLICT FOR HIM BY DT
               IF ( TCH.GT.5.0.AND.JFVA.NE.LFALSE ) TCH = TCH + DT
C-----FIND THE TIME FOR THE FRONT ZONE FOR THE IVCONF VEHICLE
      TFZ = TPASSH + TLEAD + PIJR(IDRICL) + ERRJUD/2.0
C-----FIND THE TIME FOR THE REAR ZONE FOR THE IVCONF VEHICLE
      TRZ = TPASSH + TLAG  + PIJR(IDRICL) + ERRJUD/2.0
                    IF ( VCM - VCH )              2080 , 2100 , 2090
 2080 CONTINUE
C-----THIS VEHICLE WILL BE TRAVELING SLOWER THAN THE IVCONF VEHICLE AT
C-----THE INTERSECTION CONFLICT THUS MAX THE TIME FOR THE FRONT ZONE FOR
C-----THE IVCONF VEHICLE WITH THE TIME REQUIRED FOR THE IVCONF VEHICLE
C-----TO REDUCE HIS SPEED TO MY SPEED MULTIPLIED BY THE COSINE OF THE
C-----INTERSECTION CONFLICT ANGLE PLUS THIS DRIVERS REACTION TIME
      SLOPE = -0.75*4.0*DCHAR(JD)
      TCRASH = (-ACH-SQRT(ACH**2-2.0*SLOPE*(VCH-VCM)))/SLOPE
      TFZ = AMAX1(TFZ,ABS(COS(ICONAN*RADIAN))*TCRASH+PIJR(IDRICL))
      GO TO 2100
 2090 CONTINUE
C-----THIS VEHICLE WILL BE TRAVELING FASTER THAN THE IVCONF VEHICLE AT
C-----THE INTERSECTION CONFLICT THUS MAX THE TIME FOR THE FRONT ZONE FOR
C-----THIS VEHICLE WITH THE TIME REQUIRED FOR THIS VEHICLE TO REDUCE ITS
C-----SPEED TO THE SPEED OF THE IVCONF VEHICLE MULTIPLIED BY THE COSINE
C-----OF THE INTERSECTION CONFLICT ANGLE PLUS THIS DRIVERS REACTION TIME
      SLOPE = -0.75*4.0*DCHAR(IDRICL)
      TCRASH = (-ACM-SQRT(ACM**2-2.0*SLOPE*(VCM-VCH)))/SLOPE
      TRZ = AMAX1(TRZ,ABS(COS(ICONAN*RADIAN))*TCRASH+PIJR(IDRICL))
 2100 CONTINUE
C-----FIND THE TIME THE FRONT ZONE AND REAR ZONE SHOULD ARRIVE AT THE
C-----INTERSECTION CONFLICT
      TFZ = TCH - TFZ
      TRZ = TCH + TRZ
C5                  IF ( IPRTLO . EQ . 0 )        GO TO 102
C4                  IF ( TIME . LT . TPRINT )     GO TO 102
C4    PRINT 702 , JNDEX,IV,TCM,VCM,DVM,DCM,IVCONF,TFZ,TCH,TRZ,VCH,DVH,
C4   *            DCH
C4102 CONTINUE
C-----IF THE TIME TO THE INTERSECTION CONFLICT FOR ME FALLS BETWEEN THE
C-----TIME THE FRONT ZONE OF THE IVCONF VEHICLE SHOULD ARRIVE AT THE
C-----INTERSECTION CONFLICT AND THE TIME THE REAR ZONE OF THE IVCONF
C-----VEHICLE SHOULD ARRIVE AT THE INTERSECTION CONFLICT THEN GO TO 4020
C-----AND SET THE WAKE UP TIME (THERE IS AN INTERSECTION CONFLICT)
               IF ( (TCM-TFZ)*(TCM-TRZ).LT.0.0 )  GO TO 4020
C-----SET THE NOFC VEHICLE TO THE IVCONF VEHICLE AND SET THE IVCONF
C-----VEHICLE TO THE NEXT VEHICLE THAT SHOULD HAVE TO CLEAR THE SAME
C-----INTERSECTION CONFLICT
      NOFC = IVCONF
C    COLEASE,FIND,IVCONF,VEHD,NOFC,NORC
      CALL  FIND     (IVCONF,       6,NOFC   ,    21)                COLEASE
```

```
C-----IF THERE IS ANOTHER VEHICLE THAT HAS TO CLEAR THE SAME
C-----INTERSECTION CONFLICT AND THIS VEHICLE HAS TO GO BEHIND THE LAST
C-----IVCONF VEHICLE THEN GO TO 1090 AND CHECK THE NEW IVCONF VEHICLE
                IF ( IVCONF.NE.0.AND.TCM.GT.TFZ )     GO TO 1090
C-----END OF GEOMETRIC CONFLICTING PATH LOOP
 3010 CONTINUE
 3020 CONTINUE
C-----THIS VEHICLE MAY PROCEED INTO THE INTERSECTION THUS SET THE FLAGS
C    COLEASE,STORE,LFALSE,VEHIL,IV,MCHKCF
      CALL  STORE   (LFALSE,      8,IV     ,     11)                  COLEASE
      MCHKCF = LFALSE
      MPRO = LTRUE
      MTCARS = LFALSE
      MSFLG = LFALSE
      IPRTM = 0
      JPRTM = 0
C-----SET CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
      CALL  SETCON
C-----SET ALL THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES TO FALSE
      DO 3030  I = 1 , 7
      IENT6(I) = LFALSE
 3030 CONTINUE
C-----SET THE VEHICLES ACC/DEC LOGIC TO FOLLOW THE VEHICLE AHEAD
      IFVA = LTRUE
C-----IF THE PREVIOUS VEHICLE IS NOT STOPPED THEN RETURN AND FOLLOW THE
C-----VEHICLE AHEAD ELSE SET THE VEHICLES ACC/DEC LOGIC TO ACCELERATE
C-----ACCORDING TO DESIRED SPEED AND RETURN
                      IF ( PVVEL . GT . 0.0 )       RETURN
      IFVA = LFALSE
      IACDS = LTRUE
      RETURN
 4010 CONTINUE
C-----THE TIME TO THE END OF THE LANE IS GT THE MAXIMUM TIME FROM THE
C-----END OF THE LANE THAT THIS VEHICLE MAY DECIDE TO PROCEED IF THE
C-----INTERSECTION CONFLICTS ARE CLEAR THUS SET THE WAKE UP TIME
      LOGTMP = MAX0(2,MIN1(2.0+5.0/DT,15.0,2.0+(TCM-TIM-DT)/DT))
      RETURN
 4020 CONTINUE
C-----THE TIME TO THE INTERSECTION CONFLICT FOR ME FALLS BETWEEN THE
C-----TIME THE FRONT ZONE OF THE IVCONF VEHICLE SHOULD ARRIVE AT THE
C-----INTERSECTION CONFLICT AND THE TIME THE REAR ZONE OF THE IVCONF
C-----VEHICLE SHOULD ARRIVE AT THE INTERSECTION CONFLICT THUS SET THE
C-----WAKE UP TIME (THERE IS AN INTERSECTION CONFLICT)
      LOGTMP = MAX0(2,MIN1(2.0+5.0/DT,15.0,2.0+(TRZ-TCM-DT)/DT))
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9130 CONTINUE
      CALL  ABORTR  ( MSG913,23 )
      STOP  913
      END                                                              CHKCON
```

```
      SUBROUTINE SETPTV                                                COLEASE
C    TASK,SETPTV
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,   COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)   COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA     ,   COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL     ,   COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL     ,   COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                  COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL     ,   COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT      ,   COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP    ,   COLEASE
     *                  ILCH      ,IGEOCP(60)                         COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS     ,   COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL    ,   COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS    ,   COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA   ,   COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT     ,   COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG     ,   COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS    ,   COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK   ,   COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS    ,   COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP       COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF      ,   COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN    ,   COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD       COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / VCHKSD(33),VCHKCO(39),AO,JD,JSLIM,JSPD,JSPDP,JV,
     *                  LGEOM4,MIMP,P,PO,SO,VO,VPREDT(21),ZTEMPD(5)
      DATA      N1,N2   / 4HSETP,2HTV /
C
C-----SUBROUTINE SETPTV SETS THIS VEHICLES PARAMETERS FOR PREDICTING
C-----TIME AND VELOCITY TO AN INTERSECTION CONFLICT
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THIS VEHICLES PARAMETERS FOR PREDICTING TIME AND VELOCITY TO
C-----AN INTERSECTION CONFLICT
      SO     = SLPNEW
      AO     = ACCNEW
      VO     = VELNEW
      PO     = POSNEW
      JSPD   = ISPD
      JSPDP  = ISPDP
      MIMP   = LIMP
      JSLIM  = ISLIM
      LGEOM4 = LGEOM(4)
      JD     = IDRICL
      JV     = IVEHCL
      RETURN
      END                                                              SETPTV
```

```
      SUBROUTINE PREDTV ( T,VX,AX )
      COMMON / CLASS   / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                   DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER    / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                   CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                   APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VCHKSD(33),VCHKCO(39),AO,JD,JSLIM,JSPD,JSPDP,JV,
     *                   LGEOM4,MIMP,P,PO,SO,VO,A,ACC,ACCM,ACCV,AN,B,C,
     *                   CRISLP,DV,PN,RADICL,RELDIS,SLOPE,SN,SPD,TT,VN,
     *                   VTT,XCRIT,XPER,XT,ZTEMPD(5)
      DATA     N1,N2   / 4HPRED,2HTV /
C
C-----SUBROUTINE PREDTV PREDICTS THE TIME AND VELOCITY TO AN
C-----INTERSECTION CONFLICT
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR PREDTV
      DV = JSPD
      T = DT
      SPD = FLOAT(JSPD)*FLOAT(MIMP)/FLOAT(JSLIM)
      CRISLP = 4.0*DCHAR(JD)
 1001 CONTINUE
C-----IF THE VEHICLE HAS ALREADY SET THE DESIRED SPEED FOR HIS
C-----INTERSECTION PATH THEN GO TO 1003 AND CONTINUE
                      IF ( JSPDP . NE . 0 )         GO TO 1003
C-----THE CODE FROM HERE TO 1003 MIMICS SUBROUTINE CHKDSP
      RELDIS = LGEOM4 - PO
C-----IF THE DISTANCE TO THE END OF THE LANE IS LE 25 FEET THEN GO TO
C-----1002 AND SET THE DESIRED SPEED FOR THE INTERSECTION PATH
                      IF ( RELDIS . LE . 25.0 )     GO TO 1002
C-----IF THE VEHICLES OLD VELOCITY IS LT THE DESIRED SPEED FOR HIS
C-----INTERSECTION PATH THEN GO TO 1003 AND CONTINUE
                      IF ( VO . LT . SPD )          GO TO 1003
C-----FIND THE DISTANCE REQUIRED TO REDUCE THE PRESENT VELOCITY OF THE
C-----VEHICLE TO THE DESIRED SPEED OF THE INTERSECTION PATH USING SLOPE
      SLOPE = -1.5*DCHAR(JD)
      TT = (-AO-SQRT(AO**2-2.0*SLOPE*(VO-SPD)))/SLOPE + DT
      XCRIT = VO*TT + 0.5*AO*TT**2 + SLOPE*TT**3/6.0
C-----IF THE DISTANCE TO THE END OF THE LANE IS GT THE DISTANCE REQUIRED
C-----TO REDUCE THE PRESENT VELOCITY OF THE VEHICLE TO THE DESIRED SPEED
C-----OF THE INTERSECTION PATH THEN GO TO 1003 AND CONTINUE
                      IF ( RELDIS . GT . XCRIT )    GO TO 1003
 1002 CONTINUE
C-----SET THE VEHICLES DESIRED SPEED TO THE DESIRED SPEED FOR THE
C-----INTERSECTION PATH AND SET THE FLAG TO INDICATE THAT THE VEHICLES
C-----DESIRED SPEED HAS BEEN RESET
      JSPD = SPD + 0.5
      DV   = JSPD
      JSPDP = 1
 1003 CONTINUE
C-----THE CODE FROM HERE TO 5010 MIMICS SUBROUTINE ACCEL
C-----IF THE VEHICLES OLD VELOCITY IS LT HIS DESIRED SPEED THEN GO TO
C-----1010 AND CHECK FOR ACCELERATION TO THE VEHICLES DESIRED SPEED
                      IF ( VO . LE . DV-0.5*DT )    GO TO 1010
C-----IF THE VEHICLES OLD VELOCITY IS GT HIS DESIRED SPEED THEN GO TO
C-----2010 AND CHECK FOR DECELERATION TO THE VEHICLES DESIRED SPEED
                      IF ( VO . GT . DV+1.0*DT )    GO TO 2010
C-----THE VEHICLES VELOCITY IS VERY NEAR THE VEHICLES DESIRED SPEED THUS
C-----IF THE VEHICLES ACC/DEC IS GT A VALUE THAT COULD BE REDUCED TO
C-----ZERO IN ONE DT THEN GO TO 4010 AND REDUCE THE VEHICLES ACC/DEC TO
C-----ZERO
                      IF ( ABS(AO).GT.CRISLP*DT )   GO TO 4010
C-----SET THIS VEHICLE AT HIS DESIRED SPEED WITH ACC/DEC AND ACC/DEC
C-----SLOPE OF ZERO
      SN = 0.0
      AO = 0.0
      VO = DV
      GO TO 5010
 1010 CONTINUE
C-----ACCELERATE THE VEHICLE TO HIS DESIRED SPEED
C-----CALCULATE THE MAXIMUM ACCELERATION THE DRIVER WOULD USE TO GET TO
C-----HIS DESIRED SPEED IN THE LINEAR ACCELERATION MODEL
      ACCM = AUTOL*(3.2+0.08*DV)*DCHAR(JD)
C-----CALCULATE THE MAXIMUM ACCELERATION OF THE VEHICLE AT THE CURRENT
C-----VELOCITY USING THE NON-UNIFORM THEORY OF ACCELERATION
      ACCV = AMAX(JV)*(1.0-VO/VMAX(JV))
C-----CALCULATE THE PORTION OF THE MAXIMUM ACCELERATION THAT THE DRIVER
C-----WOULD USE TO GET TO HIS DESIRED SPEED FROM HIS CURRENT VELOCITY
      ACC = AMIN1(ACCM,ACCV)*(1.0-VO/(1.15*DV))
C-----IF THE VEHICLES ACC/DEC IS LT THE DESIRED ACC/DEC THEN GO TO 3010
C-----AND MOVE THE VEHICLES ACC/DEC TO ACC IN PIJR TIME
                      IF ( AO . LT . ACC )          GO TO 3010
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC
C-----TO ACC IN DT SECONDS
      SN = (ACC-AO)/DT
C-----BOUND THE VEHICLES ACC/DEC SLOPE AND CHECK THE NEW VELOCITY
      SN = AMIN1(AMAX1(SN,-CRISLP),1.3*CRISLP)
      GO TO 3020
 2010 CONTINUE
C-----CHECK TO SEE IF THE VEHICLE SHOULD BEGIN TO DECELERATE TO HIS
C-----DESIRED SPEED BY THE TIME HE REACHES THE END OF HIS LANE
      SN = -0.25*CRISLP
                      IF ( AO . LT . SN )           SN = 0.5*SN
                      IF ( AO . EQ . 0.0 )          AO = 1.0E-6
      A = AO/6.0
      B = (2.0*VO+DV)/3.0
      C = PO - (LGEOM4-DV)
      RADICL = B**2 - 4.0*A*C
                      IF ( RADICL . LE . 0.0 )      GO TO 2020
      TT = (-B+SQRT(RADICL))/(2.0*A)
                      IF ( TT . LE . 0.0 )          GO TO 2020
C-----FIND THE ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES VELOCITY TO
C-----HIS DESIRED SPEED BEFORE HE REACHES THE END OF HIS LANE AND BOUND
C-----THE ACC/DEC SLOPE
      SN = AMIN1(SN,2.0*(DV-VO-AO*TT)/TT**2)
 2020 CONTINUE
                      IF ( AO . GE . 0.0 )          GO TO 2030
C-----FIND THE ACC/DEC SLOPE REQUIRED TO BRING THE ACC/DEC TO ZERO BY
C-----THE TIME THE VEHICLES VELOCITY REACHES HIS DESIRED SPEED
      SLOPE = -0.5*AO**2/(DV-VO)
                      IF ( SLOPE.LT.0.40*CRISLP )   GO TO 2030
C-----SET THE ACC/DEC SLOPE TO BRING THE ACC/DEC TO ZERO BY THE TIME THE
C-----VEHICLES VELOCITY REACHES HIS DESIRED SPEED
      SN = SLOPE
 2030 CONTINUE
C-----BOUND THE ACC/DEC SLOPE TO DECELERATE TO HIS DESIRED SPEED
      SN = AMIN1(AMAX1(SN,-CRISLP),CRISLP)
      GO TO 5010
 3010 CONTINUE
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC
C-----TO THE NEW ACC IN PIJR
      SN = 1.01*(ACC-AO)/PIJR(JD)
C-----BOUND THE ACC/DEC SLOPE FOR ACCELERATION TO ACC IN PIJR
      SN = AMIN1(AMAX1(SN,SO),1.3*CRISLP)
      AN = AO + SN*DT
C-----IF THE VEHICLES ACC/DEC AFTER DT SECONDS WILL STILL BE LT ACC THEN
C-----GO TO 3020 AND CHECK THE VELOCITY AFTER DT SECONDS ELSE CALCULATE
C-----THE ACC/DEC SLOPE REQUIRED TO BRING THE VEHICLES ACC/DEC TO ACC IN
C-----ONE DT AND CHECK VELOCITY AFTER DT SECONDS
                      IF ( AN . LT . ACC )          GO TO 3020
      SN = (ACC-AO)/DT
 3020 CONTINUE
C-----CHECK TO SEE THAT THE VEHICLES VELOCITY WOULD NOT BE ABOVE THE
C-----DESIRED SPEED AFTER THE ACC/DEC FOR THE VEHICLE WAS REDUCED TO
C-----ZERO AT HALF THE CRITICAL SLOPE
      SLOPE = -0.50*CRISLP
      TT = AMAX1(-AO/SLOPE,0.01)
      VTT = VO + AO*TT + 0.5*SLOPE*TT**2
```

```
                    IF ( VTT . LT . DV )          GO TO 5010
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED SO THAT VTT WOULD NOT EXCEED
C-----THE DESIRED SPEED BEFORE THE ACC/DEC COULD BE REDUCED TO ZERO AND
C-----BOUND THE ACC/DEC SLOPE
      SN = AMIN1(AMAX1((VTT/DV)*(-AO/TT),-CRISLP),1.3*CRISLP)
      GO TO 5010
 4010 CONTINUE
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES
C-----ACC/DEC TO ZERO IN ONE DT AND BOUND THE ACC/DEC SLOPE
      SN = AMIN1(AMAX1(-AO/DT,-CRISLP),CRISLP)
 5010 CONTINUE
C-----UPDATE THE VEHICLES POS/VEL/ACC FOR THE NEXT DT
      AN = AO + SN*DT
      VN = VO + AO*DT + 0.5*SN*DTSQ
      PN = PO + VO*DT + 0.5*AO*DTSQ + SN*DTCU/6.0
C-----IF THE VEHICLE STOPPED THIS DT THEN GO TO 6010 AND PROCESS THE
C-----STOP
                    IF ( VN . LT . 0.0 )          GO TO 6010
 5020 CONTINUE
C-----IF THE VEHICLE PASSED THE CONFLICT THEN GO TO 5030 AND FINISH
                    IF ( PN . GT . P )            GO TO 5030
C-----INCREMENT TIME AND SET THE OLD POS/VEL/ACC TO THE NEW POS/VEL/ACC
      T = T + DT
      SO = SN
      AO = AN
      VO = VN
      PO = PN
C-----GO TO 1001 AND PROCESS ANOTHER DT
      GO TO 1001
 5030 CONTINUE
C-----THE VEHICLE PASSED THE CONFLICT THUS FIND THE PORTION OF THE DT
C-----THAT WAS USED TO GET TO THE CONFLICT
      XPER = 1.0
                    IF ( PN-PO . NE . 0.0 )       XPER = (P-PO)/(PN-PO)
C-----FIND THE TIME TO THE CONFLICT, THE ACCELERATION AT THE CONFLICT,
C-----AND THE VELOCITY AT THE CONFLICT
      T = T - DT + XPER*DT
      AX = AO + XPER*(AN-AO)
      VX = VO + XPER*(VN-VO)
      RETURN
 6010 CONTINUE
C-----THE VEHICLE STOPPED THIS DT THUS FIND THE TIME DURING THIS DT THAT
C-----THE VEHICLE STOPPED, INCREMENT TIME, AND RESET POS/VEL/ACC
      XT = DT*VO/(VO-VN)
      T = T - DT + XT
      PN = PO + VO*XT + 0.5*AO*XT**2 + SN*XT**3/6.0
      SN = 0.0
      AN = 0.0
      VN = 0.0
      GO TO 5020
      END                                                               PREDTV
```

```
      SUBROUTINE SETCON                                                 COLEASE
C   TASK,SETCON
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,     COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         MSG914(5)
      DATA      MSG914 / 4H LNE,4HXT E,4HQ 0 ,4H- SE,4HTCON /
      DATA      N1,N2  / 4HSETC,2HON /
C4701 FORMAT(30H SETTING CONFLICTS FOR VEHICLEI4,9H FOR PATHI4)
C
C-----SUBROUTINE SETCON SETS CONFLICTS FOR THE VEHICLE FOR HIS
C-----INTERSECTION PATH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE INTERSECTION CONTROL LOGIC TIMER FOR NEVER PROCESS AGAIN
      LOGTMP = 0
C-----IF THE VEHICLE HAS ALREADY SET CONFLICTS THEN RETURN
                    IF ( NORC . NE . 200+1 )     RETURN
                    IF ( LNEXT . EQ . 0 )        GO TO 9140
C-----SET THE POSITION FOR CHECKING TO THE NEW POSITION
      IPOSCK = POSNEW*25.0 + 0.5
                    IF ( LCHGE . NE . 2 )        GO TO 1010
      POSLAT = LATPOS/8.0 - 15.0
                    IF ( POSLAT . LE . 0.0 )     GO TO 1010
C-----THE VEHICLE IS LANE CHANGING TO THE LEFT THUS SET THE POSITION FOR
C-----CHECKING TO THE OLD POSITION
      IPOSCK = IPOS
 1010 CONTINUE
                    IF ( IPTHUP . EQ . LNEXT )   GO TO 1020
C   COLEASE,EXTRAC,PATH,LNEXT
      CALL  EXTRAC  (      4,LNEXT )                                    COLEASE
      IPTHUP = LNEXT
 1020 CONTINUE
C-----INITIALIZE THE VEHICLES NEAREST VEHICLE TO THE REAR FOR CONFLICT
C-----CHECKING
      NORC = 0
C5                  IF ( IPRTLO . EQ . 0 )       GO TO 101
C4                  IF ( TIME . LT . TPRINT )    GO TO 101
C4    PRINT 701 , IV,LNEXT
C4101 CONTINUE
                    IF ( NOR . EQ . 0 )          GO TO 1030
C-----WAKE UP THE NOR VEHICLE FOR INTERSECTION CONTROL LOGIC
```

```
C   COLEASE,FIND,MOGFLG,VEHD,NOR,LOGFLG
      CALL  FIND    (MOGFLG,      6,NOR    ,      22)                       COLEASE
      MOGFLG = MIN0(MOGFLG,2)
C   COLEASE,STORE,MOGFLG,VEHD,NOR,LOGFLG
      CALL  STORE   (MOGFLG,      6,NOR    ,      22)                       COLEASE
 1030 CONTINUE
C-----IF THERE ARE NO GEOMETRIC CONFLICTING PATHS THEN RETURN
                    IF ( NGEOCP . LE . 0 )        RETURN
C-----PROCESS EACH GEOMETRIC CONFLICTING PATH
      DO 1090  I = 1 , NGEOCP
C-----INITIALIZE SOME PARAMETERS FOR THIS LOOP
      JGEOCP = IGEOCP(I)
                    IF ( ICONUP . EQ . JGEOCP )   GO TO 1040
C   COLEASE,EXTRAC,CONFLT,JGEOCP
      CALL  EXTRAC  (      2,JGEOCP)                                        COLEASE
      ICONUP = JGEOCP
 1040 CONTINUE
      J = 1
                    IF ( LNEXT . EQ . ICONP(1) ) J = 2
      JP = ICONP(J)
      JCONI = ICONI(J)
C   COLEASE,FIND,JCPSET,PATH,JP,ICPSET(JCONI)
      CALL  FIND    (JCPSET,      4,JP     ,      10+JCONI )                COLEASE
      J = 3 - J
C-----IF THE OTHER INTERSECTION PATH INVOLVED IN THIS INTERSECTION
C-----CONFLICT ALREADY HAS THE INTERSECTION CONFLICT SET THEN GO TO 1050
C-----AND CHECK WHERE THIS VEHICLE FITS IN
                    IF ( JCPSET . EQ . 1 )        GO TO 1050
C-----SET THIS VEHICLE AS THE NEXT VEHICLE THAT HAS NOT CLEARED THE
C-----INTERSECTION CONFLICT
C   COLEASE,STORE,IV,CONFLT,JGEOCP,ICONV(J)
      CALL  STORE   (IV     ,      2,JGEOCP,      9+J      )                COLEASE
      ICONV(J) = IV
C-----INCREMENT THE NUMBER OF CONFLICTS SET FOR THE OTHER INTERSECTION
C-----PATH INVOLVED IN THE INTERSECTION CONFLICT
C   COLEASE,FIND,MCPSET,PATH,JP,NCPSET
      CALL  FIND    (MCPSET,      4,JP     ,      10)                       COLEASE
      MCPSET = MCPSET + 1
C   COLEASE,STORE,MCPSET,PATH,JP,NCPSET
      CALL  STORE   (MCPSET,      4,JP     ,      10)                       COLEASE
C-----SET THE CONFLICT FOR THE OTHER INTERSECTION PATH INVOLVED IN THE
C-----INTERSECTION CONFLICT
C   COLEASE,STORE,1,PATH,JP,ICPSET(JCONI)
      CALL  STORE   (1      ,      4,JP     ,      10+JCONI )                COLEASE
C-----SKIP TO THE NEXT GEOMETRIC CONFLICT FOR THIS INTERSECTION PATH
      GO TO 1090
 1050 CONTINUE
C-----THE OTHER INTERSECTION PATH INVOLVED IN THIS INTERSECTION
C-----CONFLICT ALREADY HAS THE INTERSECTION CONFLICT SET THUS CHECK
C-----WHERE THIS VEHICLE FITS IN THUS SET THE NOFC AND INOW VEHICLE TO
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE CONFLICT
      NOFC = ICONV(J)
      INOW = NOFC
 1060 CONTINUE
C-----FIND SOME ATTRIBUTES OF THE INOW VEHICLE
C   COLEASE,FIND,MORC,VEHD,INOW,NORC
      CALL  FIND    (MORC   ,      6,INOW   ,      21)                       COLEASE
C   COLEASE,FIND,JPOS,VEHD,INOW,IPOS
      CALL  FIND    (JPOS   ,      6,INOW   ,       4)                       COLEASE
C   COLEASE,FIND,NININT,VEHD,INOW,MININT
      CALL  FIND    (NININT,      6,INOW   ,      33)                       COLEASE
C-----IF THERE IS NO VEHICLE TO THE REAR OF THE INOW VEHICLE THAT HAS
C-----TO CLEAR THE SAME CONFLICT THEN GO TO 1070 AND CHECK SETTING NORC
                    IF ( MORC . EQ . 0 )          GO TO 1070
C-----IF THE INOW VEHICLE IS NOT IN THE INTERSECTION AND THIS VEHICLE IS
C-----FURTHER DOWN THE LANE THAN THE INOW VEHICLE THEN GO TO 1080 AND
C-----SET THIS VEHICLE BETWEEN THE NOFC VEHICLE TO THE FRONT AND THE
C-----INOW VEHICLE TO THE REAR
      IF ( NININT.EQ.LFALSE.AND.IPOSCK.GT.JPOS ) GO TO 1080
C-----SET THE NOFC VEHICLE TO THE INOW VEHICLE AND SET THE INOW VEHICLE
C-----TO THE NEXT VEHICLE TO THE REAR THAT HAS TO CLEAR THE SAME
C-----INTERSECTION CONFLICT AND CHECK AGAIN
      NOFC = INOW
      INOW = MORC
      GO TO 1060
 1070 CONTINUE
C-----THERE IS NO VEHICLE TO THE REAR OF THE INOW VEHICLE THAT HAS TO
C-----TO CLEAR THE SAME CONFLICT THUS IF THE INOW VEHICLE IS IN THE
C-----INTERSECTION OR THIS VEHICLE IS BEHIND THE INOW VEHICLE ON THE
C-----LANE THEN GO TO 2020 AND SET THIS VEHICLE AS THE NORC VEHICLE FOR
C-----THE INOW VEHICLE AND RETURN (THIS VEHICLES NORC IS 0)
                    IF ( NININT . EQ . LTRUE )    GO TO 2020
                    IF ( IPOSCK . LT . JPOS )     GO TO 2020
 1080 CONTINUE
C-----THE INOW VEHICLE IS NOT IN THE INTERSECTION AND THIS VEHICLE IS
C-----FURTHER DOWN THE LANE THAN THE INOW VEHICLE THUS THIS VEHICLE
C-----SHOULD FIT BETWEEN THE NOFC VEHICLE TO THE FRONT AND THE INOW
C-----VEHICLE TO THE REAR THUS SET THIS VEHICLES NORC TO THE INOW
C-----VEHICLE
      NORC = INOW
C-----IF THE INOW VEHICLE IS NOT THE NEXT VEHICLE THAT HAS NOT CLEARED
C-----THE INTERSECTION CONFLICT THEN GO TO 2010 AND SET THE NORC OF THE
C-----NOFC VEHICLE TO THIS VEHICLE AND RETURN ELSE SET THIS VEHICLE AS
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT
C-----AND CHECK THE NEXT GEOMETRIC CONFLICTING PATH
                    IF ( INOW . NE . ICONV(J) )   GO TO 2010
C   COLEASE,STORE,IV,CONFLT,JGEOCP,ICONV(J)
      CALL  STORE   (IV     ,      2,JGEOCP,      9+J      )                COLEASE
      ICONV(J) = IV
C-----END OF GEOMETRIC CONFLICTING PATH LOOP
 1090 CONTINUE
      RETURN
 2010 CONTINUE
C-----THE INOW VEHICLE IS NOT THE NEXT VEHICLE THAT HAS NOT CLEARED THE
C-----INTERSECTION CONFLICT THUS SET THE NORC OF THE NOFC VEHICLE TO
C-----THIS VEHICLE AND RETURN
C   COLEASE,STORE,IV,VEHD,NOFC,NORC
      CALL  STORE   (IV     ,      6,NOFC   ,      21)                       COLEASE
      RETURN
 2020 CONTINUE
C-----THERE IS NO VEHICLE TO THE REAR OF THE INOW VEHICLE THAT HAS TO
C-----CLEAR THE SAME CONFLICT AND THE INOW VEHICLE IS IN THE
C-----INTERSECTION OR THIS VEHICLE IS BEHIND THE INOW VEHICLE ON THE
C-----LANE THUS SET THIS VEHICLE AS THE NORC VEHICLE FOR THE INOW
C-----VEHICLE AND RETURN (THIS VEHICLES NORC IS 0)
C   COLEASE,STORE,IV,VEHD,INOW,NORC
      CALL  STORE   (IV     ,      6,INOW   ,      21)                       COLEASE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9140 CONTINUE
      CALL  ABORTR  ( MSG914,20 )
      STOP  914
      END                                                                   SETCON
```

```
      SUBROUTINE UNSETC                                                 COLEASE
C     TASK,UNSETC
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / PATH   / LENP       ,IOPT      ,LIBL      ,LOBL     ,    COLEASE
     *                  IFVP       ,ILVP      ,LIMP      ,IPT      ,    COLEASE
     *                  NGEOCP     ,NCPSET    ,ICPSET(60),LOBAP    ,    COLEASE
     *                  ILCH       ,IGEOCP(60)                          COLEASE
      COMMON / VEHD   / ISLP       ,IACC      ,IVEL      ,IPOS     ,    COLEASE
     *                  ISET       ,LCHGE     ,ISPDP     ,LEGAL    ,    COLEASE
     *                  IPRTM      ,ITIMV     ,IQDS      ,ISPDS    ,    COLEASE
     *                  ISDS       ,IDVS      ,ISTCON    ,IVMAXA   ,    COLEASE
     *                  IVMAXD     ,LATPOS    ,IDTS      ,LALT     ,    COLEASE
     *                  NORC       ,LOGFLG    ,MSTPF     ,MLAG     ,    COLEASE
     *                  MTCARS     ,MFINL     ,MSFLG     ,MPOBS    ,    COLEASE
     *                  MOASF      ,MSAOR     ,MPRO      ,MBLOCK   ,    COLEASE
     *                  MININT     ,IFVA      ,IACDS     ,ICDFS    ,    COLEASE
     *                  ISDEC      ,ISTMO     ,IACLDS    ,IRSTOP        COLEASE
      COMMON / VEHF   / IDRICL     ,IVEHCL    ,ISPD      ,NOF      ,    COLEASE
     *                  NOR        ,LNEXT     ,LPRES     ,ITURN    ,    COLEASE
     *                  IBAPS      ,IPRTLO    ,IEXTIM    ,NOBAPD        COLEASE
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DATA     N1,N2  / 4HUNSE,2HTC /
C4701 FORMAT(32H UNSETTING CONFLICTS FOR VEHICLEI4,9H FOR PATHI4)
C
C-----SUBROUTINE UNSETC UNSETS THE CONFLICTS FOR THE VEHICLE FOR HIS
C-----INTERSECTION PATH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE INTERSECTION CONTROL LOGIC TIMER FOR PROCESS NEXT DT
      LOGTMP = 2
C-----IF THE VEHICLE HAS NOT SET CONFLICTS THEN RETURN
                    IF ( NORC . EQ . 200+1 )      RETURN
                    IF ( IPTHUP . EQ . LNEXT )    GO TO 1010
C     COLEASE,EXTRAC,PATH,LNEXT
      CALL  EXTRAC  (    . 4,LNEXT )                                    COLEASE
      IPTHUP = LNEXT
 1010 CONTINUE
C5                  IF ( IPRTLO . EQ . 0 )        GO TO 101
C4                  IF ( TIME . LT . TPRINT )     GO TO 101
C4    PRINT 701 , IV,LNEXT
C4101 CONTINUE
C-----IF THERE ARE NO GEOMETRIC CONFLICTING PATHS THEN GO TO 2010 AND
C-----SET THE FLAG FOR CONFLICTS NOT SET AND RETURN
                    IF ( NGEOCP . LE . 0 )        GO TO 2010
C-----PROCESS EACH GEOMETRIC CONFLICTING PATH
      DO 1070  I = 1 , NGEOCP
C-----INITIALIZE SOME PARAMETERS FOR THE GEOMETRIC CONFLICTING PATH LOOP
      JGEOCP = IGEOCP(I)
                    IF ( ICONUP . EQ . JGEOCP )  GO TO 1020
C     COLEASE,EXTRAC,CONFLT,JGEOCP
      CALL  EXTRAC  (       2,JGEOCP)                                   COLEASE
      ICONUP = JGEOCP
 1020 CONTINUE
      J = 1
                    IF ( LNEXT . EQ . ICONP(2) ) J = 2
C-----IF THERE ARE NO MORE VEHICLES THAT HAVE NOT CLEARED THE
C-----INTERSECTION CONFLICT THEN GO TO 2010 AND SET THE FLAG FOR
C-----CONFLICTS NOT SET AND RETURN
                    IF ( ICONV(J) . EQ . 0 )      GO TO 2010
C-----IF THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT
C-----IS NOT THIS VEHICLE THEN GO TO 1040 AND CHAIN DOWN THE LINKS OF
C-----NORC VEHICLES AND REMOVE THIS VEHICLE FROM THE CHAIN
                    IF ( ICONV(J) . NE . IV )     GO TO 1040
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT IS
C-----THIS VEHICLE THUS IF THERE IS NO VEHICLE TO THE REAR THAT HAS TO
C-----CLEAR THE SAME INTERSECTION CONFLICT THEN GO TO 1030 AND CLEAR THE
C-----INTERSECTION CONFLICT ELSE SET THE NEXT VEHICLE THAT HAS NOT
C-----CLEARED THE INTERSECTION CONFLICT TO THE VEHICLE TO THE REAR OF
C-----THIS VEHICLE THAT HAS TO CLEAR THE SAME INTERSECTION CONFLICT
                    IF ( NORC . EQ . 0 )          GO TO 1030
C     COLEASE,STORE,NORC,CONFLT,JGEOCP,ICONV(J)
      CALL  STORE   (NORC   ,      2,JGEOCP,      9+J      )            COLEASE
      ICONV(J) = NORC
C-----GO TO 1070 AND CHECK THE NEXT GEOMETRIC CONFLICTING PATH
      GO TO 1070
 1030 CONTINUE
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT IS
C-----THIS VEHICLE AND THERE IS NO VEHICLE TO THE REAR THAT HAS TO CLEAR
C-----THE SAME INTERSECTION CONFLICT THUS CLEAR THE INTERSECTION
C-----CONFLICT
C     COLEASE,STORE,0,CONFLT,JGEOCP,ICONV(J)
      CALL  STORE   (0      ,      2,JGEOCP,      9+J      )            COLEASE
      ICONV(J) = 0
      J = 3 - J
      JP = ICONP(J)
      JCONI = ICONI(J)
C-----DECREMENT THE NUMBER OF CONFLICTS SET FOR THE OTHER INTERSECTION
C-----PATH INVOLVED IN THE INTERSECTION CONFLICT
C     COLEASE,FIND,MCPSET,PATH,JP,NCPSET
      CALL  FIND    (MCPSET,      4,JP     ,      10)                   COLEASE
      MCPSET = MAX0(MCPSET-1,0)
C     COLEASE,STORE,MCPSET,PATH,JP,NCPSET
      CALL  STORE   (MCPSET,      4,JP     ,      10)                   COLEASE
C-----UNSET THE CONFLICT FOR THE OTHER INTERSECTION PATH INVOLVED IN THE
C-----INTERSECTION CONFLICT
C     COLEASE,STORE,0,PATH,JP,ICPSET(JCONI)
      CALL  STORE   (0      ,      4,JP     ,      10+JCONI )           COLEASE
C-----GO TO 1070 AND CHECK THE NEXT GEOMETRIC CONFLICTING PATH
      GO TO 1070
 1040 CONTINUE
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT IS
C-----NOT THIS VEHICLE THUS CHAIN DOWN THE LINKS OF NORC VEHICLES AND
C-----REMOVE THIS VEHICLE FROM THE CHAIN THUS SET THE NOFC VEHICLE TO
C-----THE NEXT VEHICLE THAT HAS NOT CLEARED THE INTERSECTION CONFLICT
      NOFC = ICONV(J)
 1050 CONTINUE
C     COLEASE,FIND,MORC,VEHD,NOFC,NORC
      CALL  FIND    (MORC   ,      6,NOFC   ,      21)                  COLEASE
C-----IF THE NORC VEHICLE FOR THE NOFC VEHICLE IS THIS VEHICLE THEN GO
C-----TO 1060 AND SET THE NORC VEHICLE OF THE NOFC VEHICLE TO THE NORC
C-----VEHICLE FOR THIS VEHICLE (BREAK THIS VEHICLE OUT OF THE CHAIN
C-----BETWEEN THE NOFC VEHICLE AND HIS NORC VEHICLE)
                    IF ( MORC . EQ . IV )         GO TO 1060
C-----IF THERE IS NO VEHICLE TO THE REAR OF THE NOFC VEHICLE THAT HAS TO
C-----CLEAR THE SAME INTERSECTION CONFLICT THEN GO TO 2010 AND SET THE
C-----FLAG FOR CONFLICTS NOT SET AND RETURN ELSE SET THE NOFC VEHICLE TO
C-----THE NORC VEHICLE FOR THE OLD NOFC VEHICLE AND CHECK AGAIN
                    IF ( MORC . EQ . 0 )          GO TO 2010
      NOFC = MORC
      GO TO 1050
 1060 CONTINUE
C-----THE NORC VEHICLE FOR THE NOFC VEHICLE IS THIS VEHICLE THUS SET THE
C-----NORC VEHICLE OF THE NOFC VEHICLE TO THE NORC VEHICLE FOR THIS
C-----VEHICLE (BREAK THIS VEHICLE OUT OF THE CHAIN BETWEEN THE NOFC
C-----VEHICLE AND HIS NORC VEHICLE) AND GO TO 2010 AND SET THE FLAG FOR
C-----CONFLICTS NOT SET AND RETURN
C     COLEASE,STORE,NORC,VEHD,NOFC,NORC
      CALL  STORE   (NORC   ,      6,NOFC   ,      21)                  COLEASE
      GO TO 2010
C-----END OF GEOMETRIC CONFLICTING PATH LOOP
 1070 CONTINUE
 2010 CONTINUE
C-----SET THE FLAG FOR CONFLICTS NOT SET AND RETURN
```

```
      NORC = 200 + 1
      RETURN
      END                                                               UNSETC
```

```
      SUBROUTINE INFLZN                                                 COLEASE
C     TASK,INFLZN
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MOEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVBY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      DIMENSION         MSG915(6)
      DATA      MSG915 / 4H LCO,4HNTR ,4HEQ 1,4H = I,4HNFLZ,4HN    /
      DATA      N1,N2  / 4HINFL,2HZN /
C
C-----SUBROUTINE INFLZN INITIALIZES THE VEHICLES INTERSECTION CONTROL
C-----LOGICAL ATTRIBUTES BASED ON THE TYPE OF TRAFFIC CONTROL FOR THIS
C-----LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                  IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET SOME PARAMETERS FOR ALL TYPES OF LANE CONTROL
C-----(ALL INTERSECTION CONTROL LOGICAL INDEPENDENT ATTRIBUTES SET FALSE
C-----IN SUBROUTINE LOGIN)
      IREPIL = LTRUE
      MINFLZ = LTRUE
                  IF ( ICONTR . EQ . 1 )       MIUNC = LTRUE
C-----SET THE INTERSECTION CONTROL LOGIC TIMER FOR PROCESS NEXT DT
      LOGFLG = 2
      LOGTMP = 2
C-----PROCESS BASED ON THE LANE CONTROL
C-----       OUTB  UC  YSC  SSC  SIG SLTOR SRTOR
      GO TO ( 1010,2010,3010,4010,5010,5010,5010 ) , LCONTR
 1010 CONTINUE
C-----THIS LANE IS OUTBOUND OR A BLOCKED INBOUND LANE
      GO TO 9150
 2010 CONTINUE
C-----THIS LANE IS UNCONTROLLED THUS SET THAT THE TRAFFIC CONTROL AHEAD
C-----DOES NOT REQUIRE ME TO STOP
      MLUNC = LTRUE
      MTCARS = LFALSE
C-----THIS LANE IS UNCONTROLLED AND IF THE INTERSECTION IS ALSO
C-----UNCONTROLLED THEN RETURN
                  IF ( MIUNC . EQ . LTRUE )     RETURN
C-----THIS LANE IS UNCONTROLLED AND THE INTERSECTION IS CONTROLLED THUS
C-----SET THAT INTERSECTION CONFLICTS MUST BE CHECKED (FOR LEFT TURNS
```

```
C-----AND LANE CHANGES WITHIN THE INTERSECTION)
      MCHKCF = LTRUE
C-----IF THE VEHICLE IS TURNING LEFT THEN RETURN
                      IF ( ITURN . EQ . 1 )           RETURN
C   COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
      CALL  FIND    (JLCH  ,      4,LNEXT ,    72)                      COLEASE
C-----IF THE VEHICLES INTERSECTION PATH CHANGES LANES WITHIN THE
C-----INTERSECTION THEN RETURN
                      IF ( JLCH . NE . 0 )            RETURN
C-----THIS LANE IS UNCONTROLLED AND THE INTERSECTION IS CONTROLLED THUS
C-----SET THAT THE VEHICLE MAY PROCEED INTO THE INTERSECTION AND THAT
C-----INTERSECTION CONFLICTS NEED NOT BE CHECKED AND THAT THE TRAFFIC
C-----CONTROL AHEAD DOES NOT REQUIRE ME TO STOP
      MPRO = LTRUE
      MCHKCF = LFALSE
      MTCARS = LFALSE
      MSFLG = LFALSE
C-----SET CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH AND RETURN
      CALL  SETCON
      RETURN
 3010 CONTINUE
C-----THIS LANE IS YIELD SIGN CONTROLLED THUS SET THAT THE TRAFFIC
C-----CONTROL AHEAD DOES NOT REQUIRE ME TO STOP AND RETURN
      MLYELD = LTRUE
      MTCARS = LFALSE
      RETURN
 4010 CONTINUE
C-----THIS LANE IS STOP SIGN CONTROLLED
      MLSTOP = LTRUE
      RETURN
 5010 CONTINUE
C-----THIS LANE IS SIGNAL CONTROLLED THUS CHECK THE SIGNAL INDICATION
      MSSGRN = LTRUE
      JSISET = ISISET(ICAMPC,IBLN)
C-----DETERMINE THE APPROPRIATE DRIVER RESPONSE FOR THE SIGNAL
C-----INDICATION
      CALL  SIGRES  ( JSISET )
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9150 CONTINUE
      CALL  ABORTR  ( MSG915,21 )
      STOP  915
      END                                                               INFLZN
```

```
      SUBROUTINE PATHF  (IFORCE,NN1,NN2)                                COLEASE
C   TASK,PATHF,IFORCE,NN1,NN2
      COMMON / APPRO  / NLANES     ,LLANES( 6),NVIL  ( 6),ISLIM     ,   COLEASE
     *                  IALEFT     ,NSDR      ,ISDRN ( 5),ISDRA ( 5)    COLEASE
      COMMON / LANE   / LWID       ,NLL       ,NLR       ,ISNA      ,   COLEASE
     *                  NPINT      ,LINTP ( 7),IFVL      ,ILVL      ,   COLEASE
     *                  LCONTR     ,LTURN     ,LGEOM ( 4),NLDL      ,   COLEASE
     *                  LLDL  ( 5),IBLN       ,IDUMLA                   COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP       ,IACC      ,IVEL      ,IPOS      ,   COLEASE
     *                  ISET       ,LCHGE     ,ISPDP     ,LEGAL     ,   COLEASE
     *                  IPRTM      ,ITIMV     ,IQDS      ,ISPDS     ,   COLEASE
     *                  ISDS       ,IDVS      ,ISTCON    ,IVMAXA    ,   COLEASE
     *                  IVMAXD     ,LATPOS    ,IDTS      ,LALT      ,   COLEASE
     *                  NORC       ,LOGFLG    ,MSTPF     ,MLAG      ,   COLEASE
     *                  MTCARS     ,MFINL     ,MSFLG     ,MPOBS     ,   COLEASE
     *                  MOASF      ,MSADR     ,MPRO      ,MBLOCK    ,   COLEASE
     *                  MININT     ,IFVA      ,IACDS     ,ICDFS     ,   COLEASE
     *                  ISDEC      ,ISTMO     ,IACLDS    ,IRSTOP        COLEASE
      COMMON / VEHF   / IDRICL     ,IVEHCL    ,ISPD      ,NOF       ,   COLEASE
     *                  NOR        ,LNEXT     ,LPRES     ,ITURN     ,   COLEASE
     *                  IBAPS      ,IPRTLU    ,IEXTIM    ,NOBAPD        COLEASE
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         MSG916(11)
      DATA      MSG916 / 4H NO ,4HPATH,4HS FR,4HOM L,4HANE ,4HFOR ,
     *                  4HFORC,4HED P,4HATH ,4H= PA,4HTHF  /
      DATA      N1,N2  / 4HPATH,2HF  /
  701 FORMAT(8H VEHICLE,I4,12H ON APPROACH,I3,8H AT T = ,F7.1,
     *       23H WAS FORCED TO USE PATH,I4,12H TO APPROACH,
     *       I3,23H INSTEAD OF TO APPROACH,I3,2H (,A4,A2,1H))
C
C-----SUBROUTINE PATHF FINDS AN INTERSECTION PATH FOR THIS VEHICLE BASED
C-----ON THE CURRENT APPROACH, CURRENT LANE, AND THE DESIRED OUTBOUND
C-----APPROACH
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
                      IF ( IL . EQ . LPRES )       GO TO 1010
C   COLEASE,EXTRAC,LANE,LPRES
      CALL  EXTRAC  (       3,LPRES )                                   COLEASE
 1010 CONTINUE
C-----INITIALIZE THE FORCED PATH TO THE FIRST INTERSECTION PATH FOR LANE
C-----LPRES
      LFORCE = LINTP(1)
C-----IF THERE ARE NO INTERSECTION PATHS FROM LANE LPRES THEN GO TO 1030
C-----AND CHECK EACH LANE OF THIS APPROACH FOR AN INTERSECTION PATH TO
C-----THIS VEHICLES DESIRED OUTBOUND APPROACH
                      IF ( NPINT . LE . 0 )        GO TO 1030
C-----CHECK EACH INTERSECTION PATH FROM LANE LPRES
      DO 1020  I = 1 , NPINT
      LPATH = LINTP(I)
C   COLEASE,FIND,JOPT,PATH,LPATH,IOPT
      CALL  FIND    (JOPT  ,      4,LPATH ,     2)                      COLEASE
C-----IF THE INTERSECTION PATH IS AN OPTION1 PATH THEN SKIP TO THE NEXT
C-----INTERSECTION PATH
                      IF ( JOPT . NE . 0 )         GO TO 1020
C   COLEASE,FIND,MOBAP,PATH,LPATH,LOBAP
      CALL  FIND    (MOBAP ,      4,LPATH ,    71)                      COLEASE
C-----IF THE LINKING OUTBOUND APPROACH FOR THE INTERSECTION PATH IS EQ
C-----THE DESIRED OUTBOUND PATH FOR THIS VEHICLE THEN GO TO 3020 AND
C-----SET THIS VEHICLE TO USE THIS INTERSECTION PATH
                      IF ( MOBAP . EQ . NOBAPD )   GO TO 3020
C   COLEASE,FIND,JPT,PATH,LPATH,IPT
      CALL  FIND    (JPT   ,      4,LPATH ,     8)                      COLEASE
```

```
C-----IF THE INTERSECTION PATH TURN CODE IS STRAIGHT THEN SET THE FORCED
C-----PATH TO THIS INTERSECTION PATH
                    IF ( JPT . EQ . 2 )           LFORCE = LPATH
C-----END ON INTERSECTION PATH LOOP
 1020 CONTINUE
C-----SET THE INTERSECTION PATH FOR THIS VEHICLE TO THE FORCED PATH AND
C-----IF PATHF IS SUPPOSE TO FORCE A PATH THEN GO TO 3020 AND SET THIS
C-----VEHICLE TO USE THE FORCED INTERSECTION PATH
      LPATH = LFORCE
                    IF ( IFORCE . EQ . LTRUE )   GO TO 3020
 1030 CONTINUE
                    IF ( IFORCE . EQ . LTRUE )   GO TO 9160
C-----CHECK EACH LANE OF THIS APPROACH FOR AN INTERSECTION PATH TO THIS
C-----VEHICLES DESIRED OUTBOUND APPROACH
      DO 2020  II = 1 , NLANES
      ILANE = LLANES(II)
                    IF ( ILANE . EQ . LPRES )    GO TO 2020
C   COLEASE,FIND,MPINT,LANE,ILANE,NPINT
      CALL  FIND     (MPINT ,       3,ILANE ,       5)                       COLEASE
C-----IF THERE ARE NO INTERSECTION PATHS FROM LANE ILANE THEN GO TO 2020
C-----AND CHECK THE NEXT LANE
                    IF ( MPINT . LE . 0 )         GO TO 2020
C-----CHECK EACH INTERSECTION PATH FROM LANE ILANE TO SEE IF IT GOES TO
C-----THIS VEHICLES DESIRED OUTBOUND APPROACH
      DO 2010  I = 1 , MPINT
C   COLEASE,FIND,LPATH,LANE,ILANE,LINTP(I)
      CALL  FIND     (LPATH ,       3,ILANE ,       5+I      )               COLEASE
C   COLEASE,FIND,JOPT,PATH,LPATH,IOPT
      CALL  FIND     (JOPT  ,       4,LPATH ,       2)                       COLEASE
C-----IF THE INTERSECTION PATH IS AN OPTION1 PATH THEN SKIP TO THE NEXT
C-----INTERSECTION PATH
                    IF ( JOPT . NE . 0 )          GO TO 2010
C   COLEASE,FIND,MOBAP,PATH,LPATH,LOBAP
      CALL  FIND     (MOBAP ,       4,LPATH ,       71)                      COLEASE
C-----IF THE LINKING OUTBOUND APPROACH FOR THE INTERSECTION PATH IS EQ
C-----THE DESIRED OUTBOUND PATH FOR THIS VEHICLE THEN GO TO 3010 AND
C-----SET WHICH SIDE THE VEHICLE SHOULD LANE CHANGE TO
                    IF ( MOBAP . EQ . NOBAPD )   GO TO 3010
C-----END OF INTERSECTION PATH LOOP
 2010 CONTINUE
C-----END OF LANE LOOP
 2020 CONTINUE
C-----NO INTERSECTION PATH FROM ANY LANE FOR THIS APPROACH GOES TO THE
C-----DESIRED OUTBOUND APPROACH FOR THIS VEHICLE THUS THE TURN IS
C-----ILLEGAL FROM THIS APPROACH (SOMETHING IS VERY WRONG)
                    IF ( LCHGE . EQ . 2 )         GO TO 4010
      LEGAL = 5
C-----GO TO 4010 AND FINISH PROCESSING
      GO TO 4010
 3010 CONTINUE
C-----ONE OF THE LANES FOR THIS APPROACH HAS AN INTERSECTION PATH THAT
C-----GOES TO THE DESIRED OUTBOUND APPROACH FOR THIS VEHICLE THUS SET
C-----WHICH SIDE THE VEHICLE SHOULD LANE CHANGE TO
                    IF ( LCHGE . EQ . 2 )         GO TO 4010
C-----IF THE LANE NUMBER OF THE LANE THAT HAS AN INTERSECTION PATH THAT
C-----GOES TO THE DESIRED OUTBOUND APPROACH FOR THIS APPROACH IS LT THE
C-----LANE NUMBER FOR THE PRESENT LANE THEN SET THIS VEHICLE TO
C-----CHANGE LANES LEFT ELSE SET THIS VEHICLE TO CHANGE LANES RIGHT
      LEGAL = 3
                    IF ( ILANE . LT . LPRES )    LEGAL = 1
      ISET = 5
C-----GO TO 4010 AND FINISH PROCESSING
      GO TO 4010
 3020 CONTINUE
C-----SET THIS VEHICLE TO USE INTERSECTION PATH LPATH
                    IF ( LCHGE . NE . 2 )         LEGAL = 2
C-----CHECK MY LANE AND IF BLOCKED THEN SET PARAMETERS FOR BLOCKED LANE
      CALL  CHKMLN
C   COLEASE,FIND,JPT,PATH,LPATH,IPT
      CALL  FIND     (JPT   ,       4,LPATH ,       8)                       COLEASE
C-----SET THIS VEHICLES TURN CODE (1=U AND LEFT  2=STRAIGHT  3=RIGHT)
      ITURN = MAX0(1,4-JPT)
C   COLEASE,STORE,ITURN,VEHF,IV,ITURN
      CALL  STORE    (ITURN ,       7,IV     ,      8)                       COLEASE
      LNEXT = LPATH
C   COLEASE,STORE,LPATH,VEHF,IV,LNEXT
      CALL  STORE    (LPATH ,       7,IV     ,      6)                       COLEASE
                    IF ( MOBAP . EQ . NOBAPD )   GO TO 4010
      PRINT 701 , IV,IA,TIME,LNEXT,MOBAP,NOBAPD,NN1,NN2
      NOBAPD = MOBAP
C   COLEASE,STORE,MOBAP,VEHF,IV,NOBAPD
      CALL  STORE    (MOBAP ,       7,IV     ,      12)                      COLEASE
 4010 CONTINUE
C-----FINISH PROCESSING
                    IF ( IL . EQ . LPRES )        RETURN
C   COLEASE,EXTRAC,LANE,IL
      CALL  EXTRAC   (       3,IL      )                                     COLEASE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9160 CONTINUE
      CALL  ABORTR   ( MSG916,43 )
      STOP  916
      END                                                                    PATHF
```

```
      SUBROUTINE CHKMLN                                                 COLEASE
C     TASK,CHKMLN
      COMMON / LANE    / LWID       ,NLL        ,NLR        ,ISNA       ,   COLEASE
     *                   NPINT      ,LINTP ( 7),IFVL       ,ILVL       ,    COLEASE
     *                   LCONTR     ,LTURN      ,LGEOM ( 4),NLDL       ,    COLEASE
     *                   LLDL  ( 5),IBLN       ,IDUMLA                      COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                        COLEASE
      COMMON / VEHD    / ISLP       ,IACC       ,IVEL       ,IPOS       ,   COLEASE
     *                   ISET       ,LCHGE      ,ISPDP      ,LEGAL      ,   COLEASE
     *                   IPRTM      ,ITIMV      ,IQDS       ,ISPOS      ,   COLEASE
     *                   ISDS       ,IDVS       ,ISTCON     ,IVMAXA     ,   COLEASE
     *                   IVMAXD     ,LATPOS     ,IDTS       ,LALT       ,   COLEASE
     *                   NORC       ,LOGFLG     ,MSTPF      ,MLAG       ,   COLEASE
     *                   MTCARS     ,MFINL      ,MSFLG      ,MPOBS      ,   COLEASE
     *                   MOASF      ,MSAOR      ,MPRO       ,MBLOCK     ,   COLEASE
     *                   MININT     ,IFVA       ,IACDS      ,ICDFS      ,   COLEASE
     *                   ISDEC      ,ISTMO      ,IACLDS     ,IRSTOP         COLEASE
      COMMON / ABIAS   / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                   SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                   PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      DIMENSION          MSG917(10),MSG918(12)
      DATA      MSG917 / 4H LAN,4HE DO,4HES N,4HOT E,4HXIST,4H AT ,
     *                   4HPOSN,4HEW =,4H CHK,4HMLN  /
      DATA      MSG918 / 4H NO ,4HLANE,4H ALT,4HERNA,4HTIVE,4H FOR,
     *                   4H BLO,4HCKED,4H LAN,4HE = ,4HCHKM,4HLN    /
      DATA      N1,N2  / 4HCHKM,2HLN /
C
C-----SUBROUTINE CHKMLN CHECKS MY LANE AND IF BLOCKED THEN SETS
C-----PARAMETERS FOR BLOCKED LANE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                  IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE THE LANE NOT BLOCKED
      MBLOCK = LFALSE
C-----IF THE LANE IS CONTINUOUS THEN RETURN (NOT BLOCKED)
                  IF ( LGEOM(2).EQ.LGEOM(4) )  RETURN
C-----IF THE LANE ONLY EXISTS IN THE FIRST PART THEN GO TO 1020 AND
C-----CHECK A LANE THAT ONLY EXISTS IN THE FIRST PART OR A LANE THAT IS
C-----BLOCKED IN THE MIDDLE ONLY
                  IF ( LGEOM(1).NE.LGEOM(2) )  GO TO 1020
 1010 CONTINUE
C-----THE LANE EXISTS IN THE LAST PART THUS IF THE VEHICLES NEW POSITION
C-----IS GE THE START OF THIS SECTION THEN RETURN (NOT BLOCKED) ELSE THE
C-----VEHICLE IS IN THE BLOCKED PORTION OF THE LANE
            IF ( POSNEW . GE . FLOAT(LGEOM(3)) ) RETURN
      GO TO 9170
 1020 CONTINUE
C-----THE LANE ONLY EXISTS IN THE FIRST PART OR THE LANE IS BLOCKED IN
C-----THE MIDDLE ONLY THUS IF THE LANE IS BLOCKED IN THE MIDDLE ONLY
C-----THEN GO TO 1030 AND CHECK A LANE BLOCKED IN THE MIDDLE ONLY ELSE
C-----PROCESS A LANE THAT ONLY EXISTS IN THE FIRST PART
                  IF ( LGEOM(3).NE.LGEOM(4) )  GO TO 1030
      MBLOCK = LTRUE
C-----IF THE VEHICLE IS LANE CHANGING THEN RETURN (LANE BLOCKED)
                  IF ( LCHGE . EQ . 2 )        RETURN
C-----SET WHICH SIDE THE VEHICLE SHOULD CHANGE LANES INTO
      LEGAL = 1
                  IF ( NLL . EQ . 0 )          LEGAL = 3
                  IF ( NLL.EQ.0.AND.NLR.EQ.0 ) GO TO 9180
      ISET = 5
C-----IF THE VEHICLES NEW POSITION IS LT THE END OF THE BLOCKED LANE
C-----THEN RETURN (BLOCKED LANE) ELSE THE VEHICLE IS BEYOND THE END OF
C-----THE BLOCKED LANE
            IF ( POSNEW . LT . FLOAT(LGEOM(2)) ) RETURN
      GO TO 9170
 1030 CONTINUE
C-----THE LANE IS BLOCKED IN THE MIDDLE ONLY THUS IF THE VEHICLES NEW
C-----POSITION IS GT THE END OF THE FIRST PART THEN GO TO 1010 AND CHECK
C-----FOR THE VEHICLE BEING IN THE LAST PART OF THE LANE ELSE THE
C-----VEHICLE IS OK IN THE FIRST PART OF THE LANE (BLOCKED LANE)
            IF ( POSNEW . GE . FLOAT(LGEOM(2)) ) GO TO 1010
      MBLOCK = LTRUE
      MFINL = LFALSE
C-----IF THE PREVIOUS VEHICLES POSITION IS GT THE END OF THE FIRST PART
C-----OF THE LANE THEN THIS VEHICLE IS THE FIRST VEHICLE IN THE FIRST
C-----PART OF THE LANE BLOCKED IN THE MIDDLE ONLY
            IF ( PVPOS . GE . FLOAT(LGEOM(2)) )  MFINL = LTRUE
C-----IF THE VEHICLE IS CHANGING LANES THEN RETURN (LANE BLOCKED)
                  IF ( LCHGE . EQ . 2 )        RETURN
C-----SET WHICH SIDE THE VEHICLE SHOULD CHANGE LANES INTO
      LEGAL = 1
                  IF ( NLL . EQ . 0 )          LEGAL = 3
                  IF ( NLL.EQ.0.AND.NLR.EQ.0 ) GO TO 9180
      ISET = 5
C-----RETURN (LANE BLOCKED)
      RETURN
C-----PROCESS THE EXECUTION ERRORS AND STOP
 9170 CONTINUE
      CALL  ABORTR  ( MSG917,39 )
      STOP  917
 9180 CONTINUE
      CALL  ABORTR  ( MSG918,46 )
      STOP  918
      END
```

CHKMLN

```
      SUBROUTINE BANGS  (IWHERE)                                         COLEASE
C    TASK,BANGS,IWHERE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,     COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,     COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,     COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                     COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                     COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,     COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,     COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,     COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,     COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,     COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,     COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,     COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,     COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,     COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP          COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,     COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,     COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD          COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / MPRES,NININT,JA,JLN,JP,MLANES,JL,MOF,MOR,MORC,
     *                  JPOS,POS,JSLP,SLP,JSPD,JVEHCL,JDRICL,MNEXT,
     *                  MOBAPD,JSET,MEGAL,MOGFLG,MCHGE,KPRTM,MATPOS,
     *                  POSL,JSTCON,ISIG,JSIG,JTURN,JBAPS,ISAME,IDESPD,
     *                  POSLAT,ZTEMPD(76)
      DIMENSION         IAFORM(3),IPFORM(3),MSG919(10)
      DATA      IAFORM / 4H AP ,4HLN V,4HLPOS /
      DATA      IPFORM / 4H PAT,4HH  V,4HSCON /
      DATA      MSG919 / 4H NO ,4HLANE,4H ON ,4HLIST,4H MAT,4HCHES,
     *                   4H MPR,4HES -,4H BAN,4HGS  /
      DATA      N1,N2  / 4HBANG,2HS  /
  601 FORMAT(23H0***** COLLISION AT T =,F8.2,18H SECONDS - VEHICLE,I5,
     *       22H COLLIDED INTO VEHICLE,I5,9H RELVEL =,F6.1,9H RELPOS =,
     *       F6.1,6X,15H** COLLISION **)
  602 FORMAT(2A4,43HEH  NUM NOF NOR NORC VEHPOS VEHVEL VEH-ACC ,
     *       42HACC-SLP  DS VC DC NX OA ST LG LOG LCH PRT ,A4,6H   SIG,
     *       12H ITURN IBAPS)
  603 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,2X,I4,2I6)
  604 FORMAT(I4,I6,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,2I5,2X,I4,2I6)
C
C-----SUBROUTINE BANGS PRINTS THE COLLISION INFORMATION AND RESETS THE
C-----VEHICLES POS/VEL/ACC
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----PRINT THE TIME INTO THE SIMULATION AND THE VEHICLES INVOLVED IN
C-----THE COLLISION (THE FRONT VEHICLE IS THE PREVIOUS VEHICLE AND THE
C-----REAR VEHICLE IS THIS VEHICLE)
      PRINT 601 , TIME,IQ(IV),IQ(IVPV),RELVEL,RELPOS
C-----INITIALIZE SOME PARAMETERS FOR BANGS
C    COLEASE,FIND,MPRES,VEHF,IVPV,LPRES
      CALL  FIND    (MPRES ,      7,IVPV  ,     7)                       COLEASE
C    COLEASE,FIND,NININT,VEHD,IVPV,MININT
      CALL  FIND    (NININT,      6,IVPV  ,    33)                       COLEASE
      JA   = IA
      JLN  = ILN
      JP   = MPRES
      ISIG = 0
      JSIG = 0
      ISAME = LTRUE
C-----PROCESS BASED ON THE LOCATION OF THE REAR VEHICLE
C-----          IBAP INTR OBAP
      GO TO ( 1010,2010,3010 ) , IWHERE
 1010 CONTINUE
C-----THE REAR VEHICLE WAS ON AN INBOUND APPROACH THUS SET THE SIGNAL
C-----INDICATION FOR THE REAR VEHICLE AND THE FLAG THAT THE FRONT
C-----VEHICLE IS NOT ON THE SAME LINK AS THE REAR VEHICLE
      ISIG = ISISET(ICAMPC,IBLN)
      ISAME = LFALSE
C-----IF THE FRONT VEHICLE IS IN THE INTERSECTION THEN GO TO 4010 AND
C-----PRINT THE FRONT VEHICLE INFORMATION
                    IF ( NININT . EQ . LTRUE )   GO TO 4010
C-----THE FRONT VEHICLE WAS ALSO ON THE INBOUND APPROACH THUS SET THE
C-----SIGNAL INDICATION FOR THE REAR VEHICLE AND THE FLAG THAT THE FRONT
C-----VEHICLE IS ON THE SAME LINK AS THE REAR VEHICLE AND GO TO 4010 AND
C-----PRINT THE FRONT VEHICLE INFORMATION
      JSIG = ISIG
      ISAME = LTRUE
      GO TO 4010
 2010 CONTINUE
C-----THE REAR VEHICLE WAS IN THE INTERSECTION THUS IF THE FRONT VEHICLE
C-----WAS ALSO IN THE INTERSECTION THEN GO TO 4010 AND PRINT THE FRONT
C-----VEHICLE INFORMATION ELSE FIND THE OUTBOUND APPROACH INFORMATION
C-----FOR THE FRONT VEHICLE AND GO TO 4010 AND PRINT THE FRONT VEHICLE
C-----INFORMATION
                    IF ( NININT . EQ . LTRUE )   GO TO 4010
      ISAME = LFALSE
C    COLEASE,FIND,JA,LANE,MPRES,ISNA
      CALL  FIND    (JA    ,      3,MPRES ,     4)                       COLEASE
C    COLEASE,FIND,MLANES,APPRO,JA,NLANES
      CALL  FIND    (MLANES,      1,JA    ,     1)                       COLEASE
C-----FIND THE LANE NUMBER (1 TO 6) FOR THE FRONT VEHICLES OUTBOUND LANE
      DO 2020  JLN = 1 , MLANES
C    COLEASE,FIND,JL,APPRO,JA,LLANES(JLN)
      CALL  FIND    (JL    ,      1,JA    ,     1+JLN   )                COLEASE
                    IF ( JL . EQ . MPRES )       GO TO 4010
 2020 CONTINUE
      GO TO 9190
 3010 CONTINUE
C-----THE REAR VEHICLE WAS ON THE OUTBOUND APPROACH THUS THE FRONT
C-----VEHICLE MUST BE ON THE OUTBOUND APPROACH ALSO
 4010 CONTINUE
C-----FIND THE INFORMATION FOR THE FRONT VEHICLE
C    COLEASE,FIND,MOF,VEHF,IVPV,NOF
      CALL  FIND    (MOF   ,      7,IVPV  ,     4)                       COLEASE
C    COLEASE,FIND,MOR,VEHF,IVPV,NOR
      CALL  FIND    (MOR   ,      7,IVPV  ,     5)                       COLEASE
C    COLEASE,FIND,MORC,VEHD,IVPV,NORC
      CALL  FIND    (MORC  ,      6,IVPV  ,    21)                       COLEASE
C    COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
      CALL  FIND    (JPOS  ,      6,IVPV  ,     4)                       COLEASE
      POS = JPOS/25.0
C    COLEASE,FIND,JSLP,VEHD,IVPV,ISLP
      CALL  FIND    (JSLP  ,      6,IVPV  ,     1)                       COLEASE
      SLP = JSLP/400.0 - 12.0
C    COLEASE,FIND,JSPD,VEHF,IVPV,ISPD
      CALL  FIND    (JSPD  ,      7,IVPV  ,     3)                       COLEASE
C    COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
      CALL  FIND    (JVEHCL,      7,IVPV  ,     2)                       COLEASE
C    COLEASE,FIND,JDRICL,VEHF,IVPV,IDRICL
      CALL  FIND    (JDRICL,      7,IVPV  ,     1)                       COLEASE
```

```
C     COLEASE,FIND,MNEXT,VEHF,IVPV,LNEXT
      CALL  FIND    (MNEXT ,      7,IVPV   ,      6)                 COLEASE
C     COLEASE,FIND,MOBAPD,VEHF,IVPV,NOBAPD
      CALL  FIND    (MOBAPD,      7,IVPV   ,     12)                 COLEASE
C     COLEASE,FIND,JSET,VEHD,IVPV,ISET
      CALL  FIND    (JSET  ,      6,IVPV   ,      5)                 COLEASE
C     COLEASE,FIND,MEGAL,VEHD,IVPV,LEGAL
      CALL  FIND    (MEGAL ,      6,IVPV   ,      8)                 COLEASE
C     COLEASE,FIND,MOGFLG,VEHD,IVPV,LOGFLG
      CALL  FIND    (MOGFLG,      6,IVPV   ,     22)                 COLEASE
C     COLEASE,FIND,MCHGE,VEHD,IVPV,LCHGE
      CALL  FIND    (MCHGE ,      6,IVPV   ,      6)                 COLEASE
C     COLEASE,FIND,KPRTM,VEHD,IVPV,IPRTM
      CALL  FIND    (KPRTM ,      6,IVPV   ,      9)                 COLEASE
C     COLEASE,FIND,JTURN,VEHF,IVPV,ITURN
      CALL  FIND    (JTURN ,      7,IVPV   ,      8)                 COLEASE
C     COLEASE,FIND,JBAPS,VEHF,IVPV,IBAPS
      CALL  FIND    (JBAPS ,      7,IVPV   ,      9)                 COLEASE
C-----IF THE FRONT VEHICLE WAS IN THE INTERSECTION THEN GO TO 4020 AND
C-----PRINT THE INTERSECTION INFORMATION FOR THE FRONT VEHICLE ELSE
C-----PRINT THE INBOUND/OUTBOUND APPROACH INFORMATION FOR THE FRONT
C-----VEHICLE
                      IF ( NININT . EQ . LTRUE )   GO TO 4020
C     COLEASE,FIND,MATPOS,VEHD,IVPV,LATPOS
      CALL  FIND    (MATPOS,      6,IVPV   ,     18)                 COLEASE
      POSL = MATPOS/8.0 - 15.0
                      IF ( MCHGE . NE . 2 )        POSL = 0.0
      PRINT 602 , IAFORM
      PRINT 603 , JA,JLN,IVPV,IQ(IVPV),MOF,MOR,MORC,POS,PVVEL,PVACC,
     *            SLP,JSPD,JVEHCL,JDRICL,MNEXT,MOBAPD,JSET,MEGAL,
     *            MOGFLG,MCHGE,KPRTM,POSL,JSIG,JTURN,JBAPS
      GO TO 5010
 4020 CONTINUE
C-----THE FRONT VEHICLE WAS IN THE INTERSECTION THUS PRINT THE
C-----INTERSECTION INFORMATION FOR THE FRONT VEHICLE
C     COLEASE,FIND,JSTCON,VEHD,IVPV,ISTCON
      CALL  FIND    (JSTCON,      6,IVPV   ,     15)                 COLEASE
      PRINT 602 , IPFORM
      PRINT 604 , JP,IVPV,IQ(IVPV),MOF,MOR,MORC,POS,PVVEL,PVACC,
     *            SLP,JSPD,JVEHCL,JDRICL,MNEXT,MOBAPD,JSET,MEGAL,
     *            MOGFLG,MCHGE,KPRTM,ISTCON,JSIG,JTURN,JBAPS
 5010 CONTINUE
C-----SET THE PARAMETERS FOR PRINTING THE REAR VEHICLES INFORMATION
      IDESPD = DESVEL + 0.5
C-----IF THE REAR VEHICLE WAS IN THE INTERSECTION THEN GO TO 5030 AND
C-----PRINT THE INTERSECTION INFORMATION FOR THE REAR VEHICLE ELSE
C-----PRINT THE INBOUND/OUTBOUND APPROACH INFORMATION FOR THE REAR
C-----VEHICLE
                      IF ( MININT . EQ . LTRUE )   GO TO 5030
      POSLAT = LATPOS/8.0 - 15.0
                      IF ( LCHGE . NE . 2 )        POSLAT = 0.0
                      IF ( ISAME . EQ . LTRUE )    GO TO 5020
C-----THE FRONT VEHICLE AND THE REAR VEHICLE WERE NOT ON THE SAME LINK
C-----THUS PRINT THE HEADER FOR THE REAR VEHICLE
      PRINT 602 , IAFORM
 5020 CONTINUE
      PRINT 603 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
     *            SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
     *            LOGFLG,LCHGE,IPRTM,POSLAT,ISIG,ITURN,IBAPS
      GO TO 6010
 5030 CONTINUE
C-----THE REAR VEHICLE WAS IN THE INTERSECTION THUS PRINT THE
C-----INTERSECTION INFORMATION FOR THE REAR VEHICLE
                      IF ( ISAME . EQ . LTRUE )    GO TO 5040
C-----THE FRONT VEHICLE AND THE REAR VEHICLE WERE NOT ON THE SAME LINK
C-----THUS PRINT THE HEADER FOR THE REAR VEHICLE
      PRINT 602 , IPFORM
 5040 CONTINUE
      PRINT 604 , IP,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,SLPNEW,
     *            IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,LOGFLG,
     *            LCHGE,IPRTM,ISTCON,ISIG,ITURN,IBAPS
 6010 CONTINUE
C-----INCREMENT THE NUMBER OF COLLISIONS AND RESET THIS VEHICLES
C-----POS/VEL/ACC AND RETURN
C4    PRINT 602
C3    KPFLAG = 10H***BANG***
      NBANG(IBAPS) = NBANG(IBAPS) + 1
      SLPNEW = 0.0
      ACCNEW = PVACC
      VELNEW = AMIN1(0.95*PVVEL,DESVEL,VELNEW)
      POSNEW = AMAX1(0.0,PVPOS-2.0)
                      IF ( VELNEW . LE . 0.1 )     MSTPF = LTRUE
      MSFLG = LFALSE
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9190 CONTINUE
      CALL  ABORTR  ( MSG919,38 )
      STOP  919
      END                                                            BANGS
```

```
      SUBROUTINE BIAS                                                   COLEASE
C     TASK,BIAS
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / ZTEMPD / ZTEMPD(110)
      DATA     N1,N2  / 4HBIAS,2H    /
C
C-----SUBROUTINE BIAS BIASES THE VEHICLE ATTRIBUTES, SETS THE PREVIOUS
C-----VEHICLE PARAMETERS, AND UPDATES THE MAXIMUM ACC/DEC FOR THE VEHICLE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                    IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----BIAS THE VEHICLES ATTRIBUTES (COLEASE CAN ONLY STORE POSITIVE
C-----INTEGERS)
      ISLP = (SLPNEW+12.0)*400.0 + 0.5
      IACC = (ACCNEW+32.0)*312.5 + 0.5
      IVEL = VELNEW*25.0 + 0.5
                    IF ( IVEL . EQ . 0 )         MSTPF = LTRUE
                    IF ( IVEL . GT . 0 )         MSAOR = LFALSE
      IPOS = POSNEW*25.0 + 0.5
C-----SET THE PREVIOUS VEHICLE PARAMETERS
      PVACC = ACCNEW
      PVVEL = VELNEW
      PVPOS = POSNEW - LENV(IVEHCL) - 4.0
C-----IF THE VEHICLE WAS ACCELERATING THEN GO TO 1010 AND UPDATE THE
C-----MAXIMUM ACCELERATION FOR THE VEHICLE ELSE UPDATE THE MAXIMUM
C-----DECELERATION FOR THE VEHICLE
                    IF ( ACCOLD . GT . 0.0 )      GO TO 1010
      IVMAXD = MAX0(IVMAXD,IFIX(-ACCNEW*10.0+0.5))
      RETURN
 1010 CONTINUE
C-----UPDATE THE MAXIMUM ACCELERATION FOR THE VEHICLE
      IVMAXA = MAX0(IVMAXA,IFIX(ACCOLD*10.0+0.5))
      RETURN
      END                                                               BIAS
```

```
      SUBROUTINE LOGIN                                                  COLEASE
C     TASK,LOGIN
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM     ,    COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / LOGICV / LTRUE,LFALSE                                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
C6    COMMON / PRTPVA / DISTAD(200)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      DIMENSION         IENT6(1),IENT7(1),IENT8(1),MSG920(11)
      EQUIVALENCE       (ISLP,IENT6(1)),(IDRICL,IENT7(1)),
     *                  (MDEDIC,IENT8(1))
C7    DATA      IONE    / 1 /
      DATA      IQ0     / 0 /
      DATA      MSG920  / 4H MOR,4HE TH,4HAN 2,4H00 V,4HEHIC,4HLES ,
     *                    4HIN S,4HYSTE,4HM = ,4HLOGI,4HN    /
      DATA      N1,N2   / 4HLOGI,2HN   /
      DATA      ONETRD  / 0.333333333333333 /
  501 FORMAT(F10.0,7I5)
  601 FORMAT(25H0*****LANE FULL - VEHICLE,I5,21H ELIMINATED - QTIME =,
     *       F8.2,7H  VEH =,I3,7H  DRI =,I2,8H  DVEL =,I4,8H  OBAP =,
     *       I3,8H  IBAP =,I3,8H  IBLN =,I2,8H  SPRT =,I2,/)
C7701 FORMAT(F7.2,5I4,2F7.1)
C3702 FORMAT(3HLV=,F7.2)
```

```
C3703 FORMAT(3HET=F7.5)
C8704 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,2H F,I4,
C8   *       3(1X,A10))
CT704 FORMAT(2I3,I4,I5,3I4,F8.2,F7.2,2F8.3,I4,7I3,I4,I5,F5.1,2H F,I4)
C1705 FORMAT(19H INPUT QUEUE BUFFERI3,9H  VEHICLEI5,10H  READIN =F10.2,
C1   *       7I5)
CW756 FORMAT(8H VEHD   I3,2(I5,I6),3I2,2I3,2I5,I7,2I5,I3,3I4,I6,I4,I3,
CW   *                 I2,2X,11I1,2X,7I1)
CW757 FORMAT(8H VEHF   I3,1X,12I4)
CW758 FORMAT(8H VEHIL  I3,1X,12I2,1X,8I2)
C
C-----SUBROUTINE LOGIN LOGS THE NEW VEHICLE INTO THE INBOUND APPROACH
C-----AND LANE AND INITIALIZES THE VEHICLE ATTRIBUTES
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                     IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----FIND THE NEXT AVAILABLE ENTRY FOR THE VEH ENTITIES
      DO 1010  J = 1 , 200
      IQQ = IQQ + 1
                     IF ( IQQ . GT . 200 )        IQQ = 1
C-----IF ENTRY IQQ FOR THE VEH ENTITIES IS NOT IN USE THEN GO TO 1020
C-----AND USE ENTRY IQQ FOR THE VEH ENTITIES FOR THE NEW VEHICLE
                     IF ( IQ(IQQ) . LE . 0 )      GO TO 1020
 1010 CONTINUE
      GO TO 9200
 1020 CONTINUE
C-----FIND THE QUEUE BUFFER FOR THE NEW VEHICLE TO BE LOGGED IN BASED ON
C-----THE INBOUND APPROACH AND LANE NUMBER
      IB = LQ(IAN,ILN)
C-----LET THE NEW VEHICLE USE ENTRY IQQ FOR THE VEH ENTITIES
      IV = IQQ
C6    DISTAD(IV) = 0.0
C-----SAVE THE SEQUENTIAL VEHICLE NUMBER FOR THE NEW VEHICLE AND FLAG
C-----THE ENTRY FOR THE VEH ENTITIES IN USE
      IQ(IV) = IBUF(IB,8)
C-----SET THE LANE CHANGE FLAG FOR THE NEW VEHICLE TO NO LANE CHANGE
      MCHGE = 1
                     IF ( LPRES . NE . IL )       GO TO 1030
C-----THE LAST VEHICLE PROCESSED WAS ON THIS LANE THUS IF THAT VEHICLE
C-----WAS CHANGING LANES THEN SET THE LANE CHANGE FLAG FOR THE NEW
C-----VEHICLE TO FOLLOWING A LANE CHANGING VEHICLE
                     IF ( LCHGE . EQ . 2 )        MCHGE = 3
 1030 CONTINUE
      NUM = NOATTB(6)
C-----SET ALL THE VEHD ATTRIBUTES TO ZERO
      DO 2010  IZ = 1 , NUM
      IENT6(IZ) = 0
 2010 CONTINUE
C-----SET THE NEAREST VEHICLE TO THE FRONT AS LAST VEHICLE ON THIS LANE
      NOF = ILVL
C-----IF THERE IS A VEHICLE AHEAD THEN GO TO 2020 AND SET THE NEW
C-----VEHICLE AS THE NOR VEHICLE FOR THE VEHICLE AHEAD
                     IF ( NOF . NE . 0 )          GO TO 2020
C-----SET THE NEW VEHICLE AS THE FIRST VEHICLE IN THE LANE
C   COLEASE,STORE,IV,LANE,IL,IFVL
      CALL  STORE   (IV    ,      3,IL    ,     13)                  COLEASE
      IFVL = IV
C-----INITIALIZE SOME PARAMETERS FOR THE NEW VEHICLE (FIRST IN LANE)
      MFINL = LTRUE
      MOASF = LTRUE
      PVPOS = LGEOM(4)
      PVVEL = IBUF(IB,3)
      PVACC = 0.0
      GO TO 2030
 2020 CONTINUE
C-----SET THE NEW VEHICLE AS THE NOR VEHICLE FOR THE VEHICLE AHEAD
      MFINL = LFALSE
C   COLEASE,STORE,IV,VEHF,NOF,NOR
      CALL  STORE   (IV    ,      7,NOF   ,      5)                  COLEASE
```

```
      MOASF = LFALSE
                     IF ( PVVEL . LE . 0.1 )      MOASF = LTRUE
 2030 CONTINUE
C-----SET THE LAST VEHICLE IN THE LANE TO THE NEW VEHICLE
C   COLEASE,STORE,IV,LANE,IL,ILVL
      CALL  STORE   (IV    ,      3,IL    ,     14)                  COLEASE
      ILVL = IV
C-----INITIALIZE THE ACC/DEC LOGICAL INDEPENDENT ATTRIBUTES
      MSTPF = LFALSE
      MSFLG = LFALSE
      MPRO = LFALSE
      MPOBS = LFALSE
      MSAOR = LFALSE
      MTCARS = LTRUE
      MLAG = LFALSE
      MININT = LFALSE
C-----INITIALIZE THE VEHD ATTRIBUTES
      ISET = 6
      LEGAL = 4
      LALT = 5
      LCHGE = MCHGE
      LOGFLG = 2
      LNEXT = 0
      LATPOS = 0
      NOR = 0
      NORC = 200 + 1
C-----INITIALIZE THE VEHF ATTRIBUTES
      IVEHCL = IBUF(IB,1)
      IDRICL = IBUF(IB,2)
      ISPD = IBUF(IB,3)
      NOBAPD = IBUF(IB,4)
      XTIMEL = TIME - QTIME(IB)
      IEXTIM = 25.0*XTIMEL/DT + 0.5
      IBAPS = IAN
      LPRES = IL
      IPRTLO = IBUF(IB,7)
      ITURN = 0
C-----INITIALIZE THE UNBIASED VEHICLE PARAMETERS
      OLDDTS = 0.0
      SLPOLD = 0.0
      SLPNEW = 0.0
      SLPLCH = 0.0
      ACCOLD = 0.0
      ACCNEW = 0.0
      VELOLD = ISPD
      VELNEW = ISPD
      POSOLD = LGEOM(1)
      POSNEW = LGEOM(1)
      DESVEL = ISPD
C-----CHECK MY LANE AND IF BLOCKED THEN SET PARAMETERS FOR BLOCKED LANE
      CALL  CHKMLN
C-----IF THIS LANE IS BLOCKED AND THE PREVIOUS VEHICLES POSITION IS GE
C-----THE END OF THE BLOCKED LANE THEN THIS VEHICLE IS THE FIRST VEHICLE
C-----IN THIS BLOCKED LANE
      IF ( MBLOCK.EQ.LTRUE . AND . PVPOS.GE.FLOAT(LGEOM(2)) )
     *                                         MFINL = LTRUE
C-----IF THE NEW VEHICLE IS THE FIRST VEHICLE IN THE LANE AND THE LANE
C-----IS BLOCKED THEN RESET THE PREVIOUS VEHICLE POSITION TO THE END OF
C-----THE BLOCKED LANE
      IF ( MFINL.EQ.LTRUE.AND.MBLOCK.EQ.LTRUE )  PVPOS = LGEOM(2)
C-----IF THE NEW VEHICLE IS THE FIRST VEHICLE IN THE LANE THEN GO TO
C-----2070 AND CONTINUE ELSE FIND THE MAXIMUM VELOCITY THAT THE NEW
C-----VEHICLE CAN LOG IN AT
                     IF ( MFINL . EQ . LTRUE )    GO TO 2070
      DIST = PVPOS - LGEOM(1)
C-----IF THE REAR BUMPER OF THE PREVIOUS VEHICLE IS OFF THE START OF THE
C-----LANE THEN GO TO 5010 AND ELIMINATE THE NEW VEHICLE (LANE FULL)
                     IF ( DIST . LT . 0.0 )       GO TO 5010
      CRISLP = -3.0*DCHAR(IDRICL)
C-----IF THE PREVIOUS VEHICLE WAS ACCELERATING OR TRAVELING AT A STEADY
C-----SPEED THEN GO TO 2050 AND FIND THE MAXIMUM LOG IN VELOCITY WHEN
```

```
C-----THE PREVIOUS VEHICLE WAS ACCELERATING ELSE FIND THE MAXIMUM LOG IN
C-----VELOCITY WHEN THE PREVIOUS VEHICLE WAS DECELERATING
                  IF ( PVACC . GT . 0.0 )      GO TO 2050
C-----FIND THE TIME AND DISTANCE REQUIRED TO STOP THE PREVIOUS VEHICLE
C-----AT A CRITICAL ACC/DEC SLOPE OF -4.0 TIMES THE MAXIMUM DRIVER
C-----CHARACTERISTIC
      SLP = -4.0*DCHARM
      TSTOP = (-PVACC-SQRT(PVACC**2-2.0*SLP*PVVEL))/SLP
      XSTOP = DIST + PVVEL*TSTOP + 0.5*PVACC*TSTOP**2 + SLP*TSTOP**3/6.0
C-----FIND THE TIME TO STOP THIS VEHICLE BEHIND THE PREVIOUS VEHICLE
C-----(WHEN IT STOPS) USING THE CRITICAL SLOPE FOR THIS VEHICLE
      T = (-3.0*XSTOP/CRISLP)**ONETRD
C-----FIND THE VELOCITY THE VEHICLE COULD HAVE BEEN TRAVELING AND STILL
C-----STOP BEHIND THE PREVIOUS VEHICLE
      V = -0.5*CRISLP*T**2
C-----FIND THE ACC/DEC AT THE END OF THE STOP FOR THIS VEHICLE
      ACCNEW = CRISLP*T
C-----ITERATE TO FIND THE NEW VEHICLE LOG IN SPEED (DECMAX CHANGES AS
C-----VELOLD CHANGES AND 4 ITERATIONS OF THE LOOP LETS VELOLD CONVERGE
C-----ON AN ADEQUATE LOG IN VELOCITY WHERE THE NEW VEHICLE CAN STOP IN
C-----THE AVAILABLE DISTANCE WITHOUT EXCEEDING HIS CRITICAL SLOPE OR
C-----MAXIMUM DECELERATION FROM THAT LOG IN VELOCITY)
      DO 2040  I = 1 , 4
C-----FIND THE MAXIMUM DECELERATION THAT THE VEHICLE WOULD BE WILLING TO
C-----USE TO STOP FROM HIS OLD VELOCITY
      DECMAX = DUTOL*(-6.0-VELOLD/44.0)*DCHAR(IDRICL)
      VELOLD = 0.0
C-----IF THE ACC AT THE TIME OF STOPPING IS GE THE MAXIMUM DECELERATION
C-----THAT THE VEHICLE WOULD BE WILLING TO USE TO STOP FROM VELOLD THEN
C-----SET VELOLD TO THE MAXIMUM OF VELOLD AND V (ACCNEW DOES NOT EXCEED
C-----DECMAX AND THUS IS OK)
                  IF ( ACCNEW . GE . DECMAX )  VELOLD=AMAX1(VELOLD,V)
C-----FIND THE VELOCITY THE VEHICLE COULD HAVE BEEN AT TO STOP IN THE
C-----AVAILABLE DISTANCE AND NOT EXCEEDING DECMAX
      V = SQRT(-0.75*XSTOP*DECMAX)
C-----FIND THE TIME TO STOP FROM V
      T = -2.0*V/DECMAX
C-----FIND THE ACC/DEC SLOPE REQUIRED TO GET TO DECMAX IN T SECONDS
      SLOPE = DECMAX/T
C-----IF THE ACC/DEC SLOPE REQUIRED TO GET TO DECMAX IN T SECONDS IS GE
C-----THE DRIVERS CRITICAL SLOPE THEN SET VELOLD TO THE MAXIMUM OF
C-----VELOLD AND V (SLOPE DOES NOT EXCEED CRISLP AND THUS IS OK)
                  IF ( SLOPE . GE . CRISLP )  VELOLD=AMAX1(VELOLD,V)
C-----SET VELOLD TO THE MINIMUM OF VELOLD AND THE VEHICLES DESIRED SPEED
      VELOLD = AMIN1(VELOLD,DESVEL)
C-----END OF ITERATION LOOP
 2040 CONTINUE
C-----GO TO 2070 AND CONTINUE
      GO TO 2070
 2050 CONTINUE
C-----THE PREVIOUS VEHICLE WAS ACCELERATING OR TRAVELING AT A STEADY
C-----SPEED THUS DECREMENT THE AVAILABLE DISTANCE BY A CAR FOLLOWING
C-----DISTANCE
      DIST = DIST - 1.7*PVVEL/DCHAR(IDRICL)
C-----IF THE AVAILABLE DISTANCE IS LE 0 THEN SET THIS VEHICLES VELOLD TO
C-----THE PREVIOUS VEHICLES VELOCITY
                  IF ( DIST . LE . 0.0 )       VELOLD = PVVEL
C-----IF THIS VEHICLES OLD VELOCITY IS LE THE PREVIOUS VEHICLES VELOCITY
C-----THEN GO TO 2070 AND CONTINUE
                  IF ( VELOLD . LE . PVVEL )   GO TO 2070
C-----FIND THE TIME REQUIRED TO REDUCE THE VEHICLES VELOCITY TO THE
C-----PREVIOUS VEHICLES VELOCITY AT CRITICAL SLOPE AND WITHIN THE
C-----AVAILABLE DISTANCE
      T = (-3.0*DIST/CRISLP)**ONETRD
C-----FIND THE VELOCITY THE VEHICLE COULD HAVE BEEN AT AND STILL REDUCE
C-----HIS VELOCITY TO THE PREVIOUS VEHICLES VELOCITY IN THE AVAILABLE
C-----DISTANCE
      V = PVVEL - 0.5*CRISLP*T**2
C-----FIND THE ACC/DEC AT THE TIME THIS VEHICLES VELOCITY WAS REDUCED TO
C-----THE PREVIOUS VEHICLES VELOCITY
      ACCNEW = CRISLP*T
C-----ITERATE TO FIND THE NEW VEHICLE LOG IN SPEED (DECMAX CHANGES AS
C-----VELOLD CHANGES AND 4 ITERATIONS OF THE LOOP LETS VELOLD CONVERGE
C-----ON AN ADEQUATE LOG IN VELOCITY WHERE THE NEW VEHICLE CAN REDUCE
C-----HIS LOG IN VELOCITY TO THE PREVIOUS VEHICLES VELOCITY WITHIN THE
C-----AVAILABLE DISTANCE WITHOUT EXCEEDING HIS CRITICAL SLOPE OR MAXIMUM
C-----DECELERATION FROM THAT LOG IN VELOCITY)
      DO 2060  I = 1 , 4
C-----FIND THE PORTION OF THE MAXIMUM DECELERATION THAT THE DRIVER WOULD
C-----USE TO STOP HIS VEHICLE FROM VELOLD THAT HE IS WILLING TO USE TO
C-----REDUCE HIS LOG IN VELOCITY TO THE PREVIOUS VEHICLES VELOCITY
      FACT = (VELOLD**2-PVVEL**2)/VELOLD**2
C-----FIND THE MAXIMUM DECELERATION THAT THE DRIVER WOULD USE TO
C-----DECELERATE TO THE PREVIOUS VEHICLES SPEED
      DECMAX = DUTOL*(-6.0-VELOLD/44.0)*DCHAR(IDRICL)*FACT
      VELOLD = 0.0
C-----IF THE ACC AT THE TIME OF REACHING THE PREVIOUS VEHICLES VELOCITY
C-----IS GE THE MAXIMUM DECELERATION THAT THE VEHICLE WOULD BE WILLING
C-----TO USE TO DECELERATE TO THE PREVIOUS VEHICLES SPEED THEN SET
C-----VELOLD TO THE MAXIMUM OF VELOLD AND V (ACCNEW DOES NOT EXCEED
C-----DECMAX AND THUS IS OK)
                  IF ( ACCNEW . GE . DECMAX )  VELOLD=AMAX1(VELOLD,V)
C-----FIND THE VELOCITY THAT THE VEHICLE COULD HAVE BEEN AT AND STILL
C-----REDUCE IT TO THE PREVIOUS VEHICLES VELOCITY IN THE AVAILABLE
C-----DISTANCE AND NOT EXCEED DECMAX
      V = PVVEL + SQRT(-0.75*DIST*DECMAX)
C-----FIND THE TIME TO REDUCE THAT VELOCITY TO THE PREVIOUS VEHICLES
C-----VELOCITY AND NOT EXCEED DECMAX
      T = -2.0*(V-PVVEL)/DECMAX
C-----FIND THE ACC/DEC SLOPE REQUIRED TO GET TO DECMAX IN T SECONDS
      SLOPE = DECMAX/T
C-----IF THE ACC/DEC SLOPE REQUIRED TO GET TO DECMAX IN T SECONDS IS GE
C-----THE DRIVERS CRITICAL SLOPE THEN SET VELOLD TO THE MAXIMUM OF
C-----VELOLD AND V (SLOPE DOES NOT EXCEED CRISLP AND THUS IS OK)
                  IF ( SLOPE . GE . CRISLP )   VELOLD=AMAX1(VELOLD,V)
C-----SET VELOLD TO THE MINIMUM OF VELOLD AND THE DRIVERS DESIRED SPEED
      VELOLD = AMIN1(VELOLD,DESVEL)
C-----END OF ITERATION LOOP
 2060 CONTINUE
 2070 CONTINUE
      CRISLP = -4.0*DCHAR(IDRICL)
C-----INITIALIZE SOME PARAMETERS NECESSARY FOR SUBROUTINE ACCEL AND
C-----SUBROUTINE CARFOL
      ENDLN = LGEOM(4)
                  IF ( MBLOCK . EQ . LTRUE )   ENDLN = LGEOM(2)
      RELEND = ENDLN - POSOLD
 2080 CONTINUE
C-----PREDICT THE POS/VEL/ACC FOR THE VEHICLE AFTER XTIMEL SECONDS
      CALL  NEWVEL  ( XTIMEL,XTIMEL**2,XTIMEL**3 )
                  IF ( MFINL . EQ . LTRUE )    GO TO 2100
C-----CALCULATE THE ACC/DEC SLOPE REQUIRED TO FOLLOW THE VEHICLE AHEAD
      CALL  CARFOL
                  IF ( SLPNEW.GE.0.8*CRISLP )  GO TO 2090
      VELOLD = 0.95*VELOLD
      NRNAME = NRNAME - 2
      GO TO 2080
 2090 CONTINUE
C-----CALCULATE THE POS/VEL/ACC FOR THE VEHICLE AFTER XTIMEL SECONDS
      CALL  NEWVEL  ( XTIMEL,XTIMEL**2,XTIMEL**3 )
C-----IF THIS VEHICLE HAD A COLLISION WITH THE PREVIOUS VEHICLE OR THE
C-----VEHICLE STOPPED DURING THE PORTION OF THIS DT THEN GO TO 5010 AND
C-----ELIMINATE THE NEW VEHICLE
           IF ( POSNEW . GE . PVPOS+4.0 )       GO TO 5010
                  IF ( VELNEW . LT . 0.1 )      GO TO 5010
 2100 CONTINUE
C-----UPDATE THE AVERAGE PERCENT LOGIN VELOCITY TO DESIRED SPEED FOR
C-----THIS APPROACH
      PLVDV(IAN) = PLVDV(IAN) + VELOLD/FLOAT(ISPD)
      NLVDV(IAN) = NLVDV(IAN) + 1
C-----UPDATE THE NEW VEHICLES SIMULATION STATISTICS THUS IF THE
C-----VELOCITY IS LE XFPS THEN INCREMENT THE DELAY BELOW XX MPH
                  IF ( VELNEW . LE . XFPS )    IDVS = XTIMEL/DT + 0.5
```

```
C-----INCREMENT THE NUMBER OF VEHICLES IN THE SYSTEM, THE INBOUND
C-----APPROACH, AND THE INBOUND LANE
      NVSY = NVSY + 1
      NVIA(ISNA) = NVIA(ISNA) + 1
      NVIBA = NVIBA + 1
      NVILL = NVIL(ILN) + 1
C   COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
      CALL  STORE   (NVILL ,      1,ISNA   ,      7+ILN    )                COLEASE
      NVIL(ILN) = NVILL
      NUM = NOATTB(8)
C-----INITIALIZE THE VEHICLES INTERSECTION CONTROL LOGICAL ATTRIBUTES
C-----FALSE
      DO 3010  IZ = 1 , NUM
      IENT8(IZ) = LFALSE
 3010 CONTINUE
C-----BIAS THE VEHICLE ATTRIBUTES, SET THE PREVIOUS VEHICLE PARAMETERS,
C-----AND UPDATES THE MAXIMUM ACC/DEC FOR THE VEHICLE
      CALL  BIAS
C7    POSLAT = 0.0
C7    WRITE (4,701)  TIME,IQ(IV),IONE,IA,IL,IVEHCL,POSNEW,POSLAT
C0                     IF ( IPRTLO . EQ . 0 )        GO TO 101
C-----PRINT POS/VEL/ACC FOR THE VEHICLE
CZ    CALL  PVAPRT
C0101 CONTINUE
CV                     IF ( IPRTLO . EQ . 0 )        GO TO 102
CU                     IF ( TIME . LT . TPRINT )     GO TO 102
C3             ENCODE  ( 10,702,IPFLAG )             VELOLD
C3             ENCODE  ( 10,703,JPFLAG )             XTIMEL
C3    KPFLAG = 10HLOGGED IN
C8    IDESPD = DESVEL
C8    POSLAT = 0.0
C8    PRINT 704 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
C8   *              SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
C8   *              LOGFLG,LCHGE,IPRTM,POSLAT,ISISET(ICAMPC,IBLN),
C8   *              IPFLAG,JPFLAG,KPFLAG
CT    IDESPD = DESVEL
CT    POSLAT = 0.0
CT    PRINT 704 , IA,ILN,IV,IQ(IV),NOF,NOR,NORC,POSNEW,VELNEW,ACCNEW,
CT   *              SLPNEW,IDESPD,IVEHCL,IDRICL,LNEXT,NOBAPD,ISET,LEGAL,
CT   *              LOGFLG,LCHGE,IPRTM,POSLAT,ISISET(ICAMPC,IBLN)
CU102 CONTINUE
C-----PACK THE ATTRIBUTES FOR VEHICLE IV
C   COLEASE,REPACK,VEHD,IV
      CALL  REPACK  (      6,IV     )                                       COLEASE
C   COLEASE,REPACK,VEHF,IV
      CALL  REPACK  (      7,IV     )                                       COLEASE
C   COLEASE,REPACK,VEHIL,IV
      CALL  REPACK  (      8,IV     )                                       COLEASE
CY                     IF ( IPRTLO . EQ . 0 )        GO TO 103
CW                     IF ( TIME . LT . TPRINT )     GO TO 103
CW    NUM = NOATTB(6)
CW    PRINT 756 , IV,(IENT6(I),I=1,NUM)
CW    NUM = NOATTB(7)
CW    PRINT 757 , IV,(IENT7(I),I=1,NUM)
CW    NUM = NOATTB(8)
CW    PRINT 758 , IV,(IENT8(I),I=1,NUM)
CW103 CONTINUE
 4010 CONTINUE
C-----IF THERE HAS ALREADY BEEN AN END-OF-FILE ENCOUNTERED ON THE
C-----DRIVER-VEHICLE PROCESSOR TAPE THEN GO TO 4020 AND FLAG THE QUEUE
C-----BUFFER NOT IN USE, DECREMENT THE NUMBER OF VEHICLES IN THE QUEUE
C-----BUFFERS, AND SET THE END-OF-FILE FLAG TRUE
                       IF ( IEF . EQ . LTRUE )       GO TO 4020
C-----READ THE NEXT VEHICLE FROM THE DRIVER-VEHICLE PROCESSOR TAPE INTO
C-----THE QUEUE BUFFER JUST ASSIGNED
      READ  (IVEHP,501,END=4020)  QTIME(IB),(IBUF(IB,K),K=1,7)
      GO TO 4030
 4020 CONTINUE
C-----FLAG THE QUEUE BUFFER NOT IN USE, DECREMENT THE NUMBER OF VEHICLES
C-----IN THE QUEUE BUFFER, AND SET THE END-OF-FILE FLAG TRUE
      QTIME(IB) = -1.0
      IQF = IQF - 1
      IEF = LTRUE
 4030 CONTINUE
C-----SET THE SEQUENTIAL VEHICLE NUMBER FOR THIS VEHICLE
      IBUF(IB,8) = NUMV
      NUMV = NUMV + 1
C-----CLEAR THE QUEUE BUFFER POINTER
      LQ(IAN,ILN) = 0
                       IF ( IEF . EQ . LTRUE )       RETURN
C2                     IF ( IBUF(IB,7) . EQ . 0 )    GO TO 104
C1                     IF ( TIME . LT . TPRINT )     GO TO 104
C1    PRINT 705 , IB,IBUF(IB,8),QTIME(IB),(IBUF(IB,K),K=1,7)
C1104 CONTINUE
      RETURN
 5010 CONTINUE
C-----ELIMINATE THE VEHICLE FROM THE SIMULATION THUS INCREMENT THE
C-----NUMBER OF VEHICLES ELIMINATED FOR THIS APPROACH
      NELIM(IAN) = NELIM(IAN) + 1
      PRINT 601 , IBUF(IB,8),QTIME(IB),(IBUF(IB,I),I=1,7)
C-----FLAG THE ENTRY FOR THE VEH ENTITIES NOT IN USE
      IQ(IV) = 0
      IQQ = IQQ - 1
                       IF ( IQQ . LE . 0 )           IQQ = 200
C-----SET THE LAST VEHICLE IN THE LANE TO THIS VEHICLES NOF
C   COLEASE,STORE,NOF,LANE,IL,ILVL
      CALL  STORE   (NOF    ,      3,IL     ,      14)                      COLEASE
      ILVL = NOF
                       IF ( NOF . NE . 0 )           GO TO 5020
C-----THERE WAS NO NOF VEHICLE THUS SET THE FIRST VEHICLE IN THE LANE TO
C-----ZERO
C   COLEASE,STORE,0,LANE,IL,IFVL
      CALL  STORE   (0      ,      3,IL     ,      13)                      COLEASE
      IFVL = 0
      GO TO 4010
 5020 CONTINUE
C-----SET THE NOR FOR THE NOF VEHICLE TO ZERO
C   COLEASE,STORE,0,VEHF,NOF,NOR
      CALL  STORE   (0      ,      7,NOF    ,      5)                       COLEASE
      GO TO 4010
C-----PROCESS THE EXECUTION ERROR AND STOP
 9200 CONTINUE
      CALL  ABORTR  ( MSG920,41 )
      STOP  920
      END                                                                   LOGIN
```

```
      SUBROUTINE PRESIG
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / DTIME,ZTEMPD(109)
      DATA     N1,N2  / 4HPRES,2HIG /
C8601 FORMAT(3H 99,I2,I4,F8.2)
C4701 FORMAT(36H PRE-TIMED SIGNAL SETTINGS - PHASE =,I2,9H ICAMPO =,I3,
C4   *        9H ICAMPC =,I3,5H TP =,F6.1,9H TCAMPH =,F6.1,5H TR =,F6.1)
C
C-----SUBROUTINE PRESIG SIMULATES THE PRE-TIMED SIGNAL CONTROLLER
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INCREMENT THE TIME INTO THE PHASE
      TP = TP + DT
C-----DECREMENT THE TIME REMAINING IN THE PHASE
      TR = TR - DT
C-----SET THE OLD CAM STACK POSITION TO THE CURRENT CAM STACK POSITION
      ICAMPO = ICAMPC
C-----IF THE TIME REMAINING IN THE PHASE IS GT 0 THEN GO TO 1010 AND
C-----REMAIN IN THIS PHASE
                      IF ( TR . GT . 0.0 )          GO TO 1010
C-----THERE IS NO TIME REMAINING FOR THIS PHASE THUS GO TO THE NEXT CAM
C-----STACK POSITION
      ICAMPC = ICAMPC + 1
                      IF ( ICAMPC . GT . NCAMSP )  ICAMPC = 1
C-----GET THE PHASE NUMBER FOR THIS CAM STACK POSITION
      ICPHAS = ICAMPH(ICAMPC)
C8    DTIME = TIME + DT
C8    PRINT 601 , ICPHAS,ICAMPC,DTIME
C-----RESET THE TIME INTO THE PHASE AND THE TIME REMAINING IN THIS PHASE
      TP = 0.0
      TR = TCAMSP(ICAMPC)
 1010 CONTINUE
C4                    IF ( TIME . LT . TPRINT )    GO TO 101
C4    PRINT 701 , ICPHAS,ICAMPO,ICAMPC,TP,TCAMSP(ICAMPC),TR
C4101 CONTINUE
      RETURN
      END                                                              PRESIG
```

```
      SUBROUTINE ACTSIG                                                COLEASE
C   TASK,ACTSIG
      COMMON / LOGICV / LTRUE,LFALSE                                   COLEASE
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),IMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / DTIME,I,IDFP,IDOG,II,IOPHAS,IPCLTO,JJ,NPCLTO,
     *                  TMAG1,TMAG2,VCHKDF(3),VSETLD(3),ZTEMPD(93)
      DIMENSION         MSG921(13)
      LOGICAL           IDOG,IDOR,IDFP
      DATA     EOM    / 1.0E12 /
      DATA     IDOR   / .FALSE. /
      DATA     INO    / 3HNO  /
      DATA     INTER  / 1 /
      DATA     IOFF   / 3HOFF /
      DATA     IYES   / 3HYES /
      DATA     MAGSAT / 1 /
      DATA     MSG921 / 4H NO ,4HDEMA,4HND F,4HOR A,4HNY P,4HHASE,
     *                  4HS ON,4H LPH,4HNXT ,4HLIST,4H - A,4HCTSI,
     *                  4HG    /
      DATA     N1,N2  / 4HACTS,2HIG /
      DATA     TBIG   / 1.0E12 /
C8601 FORMAT(3H 88,I2,I4,F8.2)
C4701 FORMAT(9H ICAMPC =I3,9H ICPHAS =I3,8H INTER =I3,5H TP =,
C4   *        F5.1,5H TR =F5.1)
C4702 FORMAT(9H ICAMPC =I3,9H ICPHAS =I3,8H INTER =I3,5H TP =,
C4   *        F5.1,5H TR =F5.1,6H NLD =I3,6H LLD =10I2)
C4703 FORMAT(10H LDTRIP = 20L1,8H IDOG = L1,8H IDOR = L1,
C4   *        6H EOM =F5.1,6H TII =F5.1,6H TVI =F5.1,6H TMX =,F5.1)
C/704 FORMAT(19H GAP-OUT FROM PHASE,I2,10H IDUALL = ,A3)
C/705 FORMAT(28H MAG NOT SATISFIED FOR PHASE,I2,5H TP =,F6.1,
C/   *        8H TMAG1 =,F6.1,8H TMAG2 =,F6.1,9H NEXTPH =,I2,4H I =,I2)
C/706 FORMAT(19H MAX-OUT FROM PHASE,I2,10H IDUALL = ,A3,
C/   *        16H TMX(ICPHAS+1) =,F6.1,16H TMX(ICPHAS+2) =,F6.1)
C/707 FORMAT(6H PHASE,I2,9H NPCLTO =,I2,9H LPHNXT =,7I4)
C/708 FORMAT(8H NGAPO =,I5,8H NMAXO =,I5,6H TCI =,F5.1,6H TAR =,F5.1,
C/   *        9H NEXTPH =,I2,4H I =,I2)
C4709 FORMAT(9H ICAMPC =I3,9H ICPHAS =I3,10H IDUALL = ,A3,
C4   *        6H TII =F5.1,6H TVI =F5.1,6H TCI =F5.1,6H TAR =F5.1)
C
C-----SUBROUTINE ACTSIG SIMULATES THE SEMI-ACTUATED OR FULL-ACTUATED
C-----SIGNAL CONTROLLER
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----SET THE OLD CAM STACK POSITION TO THE CURRENT CAM STACK POSITION
      ICAMPO = ICAMPC
C-----INCREMENT THE TIME INTO THE PHASE
      TP = TP + DT
C-----DECREMENT THE TIME REMAINING IN THE PHASE
      TR = TR - DT
C4                    IF ( TIME . LT . TPRINT )    GO TO 102
C4    II = NLD(ICPHAS)
C4                    IF ( II . GT . 0 )           GO TO 101
C4    PRINT 701 , ICAMPC,ICPHAS,INTER,TP,TR
C4    GO TO 102
C4101 CONTINUE
C4    PRINT 702 , ICAMPC,ICPHAS,INTER,TP,TR,II,(LLD(JJ,ICPHAS),JJ=1,II)
C4102 CONTINUE
```

```
C-----PROCESS BASED ON THE INTERVAL WITHIN THE PHASE
C-----(G=GREEN  AC=AMBER CLEARANCE  RC=ALL-RED CLEARANCE)
C-----          G     AC     RC
      GO TO ( 1010,5010,6010 ) , INTER
 1010 CONTINUE
C-----THE SIGNAL IS IN THE GREEN SIGNAL INDICATION
C-----CHECK THE DEMAND FOR THE CURRENT PHASE FOR ONLY THE POSITIVE
C-----DETECTOR CONNECTIONS AND RETURN THE DEMAND ON GREEN
      CALL  CHKDFP  ( IDOG,ICPHAS,1 )
C-----IF THERE ALREADY IS A DEMAND ON RED THEN GO TO 1050 AND CONTINUE
                    IF ( IDOR )                    GO TO 1050
             IF ( IMINOR(ICPHAS) . EQ . INO )      GO TO 1020
C-----THE CURRENT SIGNAL PHASE IS THE MINOR PHASE FOR THE PARENT/MINOR
C-----SIGNAL PHASE CASE THUS THERE IS ALWAYS DEMAND ON RED
      IDOR = .TRUE.
      GO TO 1050
 1020 CONTINUE
C-----CHECK EACH SIGNAL PHASE FOR DEMAND ON RED
      DO 1040  II = 1 , NPHASE
      I = LPHASE(II)
C-----IF THE SIGNAL PHASE TO BE CHECKED IS THE CURRENT SIGNAL PHASE THEN
C-----GO TO 1040 AND SKIP TO THE NEXT SIGNAL PHASE
                    IF ( I . EQ . ICPHAS )         GO TO 1040
                    IF ( I . EQ . IARRPH )         GO TO 1040
                    IF ( IDUALL(ICPHAS).EQ.INO ) GO TO 1030
C-----THE CURRENT SIGNAL PHASE IS A DUAL LEFT PHASE THUS THE NEXT TWO
C-----PHASES AFTER IT MUST BE THE INDIVIDUAL LEFT TURN SIGNAL PHASES AND
C-----NEITHER OF THESE SINGLE LEFT TURN SIGNAL PHASES SHOULD REGISTER A
C-----DEMAND ON RED AND CAUSE THE DUAL LEFT PHASE TO PASS INTO A SINGLE
C-----LEFT TURN SIGNAL PHASE THUS IF THE SIGNAL PHASE BEING CHECKED IS
C-----EITHER OF THE SINGLE LEFT TURN SIGNAL PHASES THEN GO TO 1040 AND
C-----SKIP TO THE NEXT SIGNAL PHASE
                    IF ( I . EQ . ICPHAS+1 )       GO TO 1040
                    IF ( I . EQ . ICPHAS+2 )       GO TO 1040
 1030 CONTINUE
C-----CHECK THE DEMAND FOR THE SIGNAL PHASE BEING CHECKED USING BOTH THE
C-----POSITIVE AND NEGATIVE DETECTOR CONNECTIONS AND RETURN THE DEMAND
C-----ON RED
      CALL  CHKDFP  ( IDOR,I,2 )
C-----IF THERE IS A DEMAND ON RED THEN GO TO 1050 AND CONTINUE
                    IF ( IDOR )                    GO TO 1050
C-----END OF SIGNAL PHASE LOOP
 1040 CONTINUE
 1050 CONTINUE
C4                  IF ( TIME . LT . TPRINT )      GO TO 103
C4    PRINT 703 , (LDTRIP(II),II=1,20),IDOG,IDOR,EOM,TII(ICPHAS),
C4    *              TVI(ICPHAS),TMX(ICPHAS)
C4103 CONTINUE
C-----IF THERE IS A DEMAND ON GREEN THIS DT AND THE TIME INTO THE SIGNAL
C-----PHASE IS GE THE INITIAL INTERVAL FOR THE SIGNAL PHASE THEN SET THE
C-----TIME REMAINING IN THIS SIGNAL PHASE TO THE VEHICLE INTERVAL FOR
C-----THIS SIGNAL PHASE
      IF ( IDOG . AND . TP.GE.TII(ICPHAS) )        TR = TVI(ICPHAS)
C-----IF THERE IS DEMAND ON RED AND THE END OF MAX HAS NOT BEEN SET THEN
C-----SET END OF MAX TO THE MAXIMUM OF THE TIME INTO THE SIGNAL PHASE
C-----PLUS THE MAXIMUM EXTENSION AFTER DEMAND ON RED FOR THIS SIGNAL
C-----PHASE AND THE INITIAL INTERVAL PLUS THE VEHICLE INTERVAL FOR THIS
C-----SIGNAL PHASE
      IF ( IDOR . AND . EOM.EQ.TBIG )
     *               EOM = AMAX1(TP+TMX(ICPHAS),TII(ICPHAS)+TVI(ICPHAS))
C-----IF THERE HAS BEEN A DEMAND ON RED THUS END OF MAX HAS BEEN SET
C-----AND THE TIME REMAINING IN THIS SIGNAL PHASE IS LE 0 THEN GO TO
C-----2010 AND GAP-OUT THIS SIGNAL PHASE
             IF ( EOM.NE.TBIG . AND . TR.LE.0.0 ) GO TO 2010
C-----IF THE TIME INTO THE PHASE FOR THIS SIGNAL PHASE IS GE END OF MAX
C-----THEN GO TO 3010 AND MAX-OUT THIS SIGNAL PHASE
                    IF ( TP . GE . EOM )           GO TO 3010
             IF ( TR.LE.0.0 . AND . IARRPH.NE.0 ) GO TO 2010
C-----IF THE TIME REMAINING FOR THIS SIGNAL PHASE IS LT THE VEHICLE
C-----INTERVAL FOR THIS SIGNAL PHASE THEN RETURN ELSE THE TIME REMAINING
C-----FOR THIS SIGNAL PHASE WAS JUST SET TO THE VEHICLE INTERVAL FOR
C-----THIS SIGNAL PHASE THUS SET ALL DETECTORS CONNECTED POSITIVE TO
C-----THIS SIGNAL PHASE TO FALSE (DEMAND HAS BEEN SATISFIED BY RESETTING
C-----THE TIME REMAINING IN THIS SIGNAL PHASE TO THE VEHICLE INTERVAL
C-----FOR THIS SIGNAL PHASE)
                    IF ( TR . LT . TVI(ICPHAS) ) RETURN
C-----SET THE DETECTORS CONNECTED POSITIVE TO THE CURRENT SIGNAL PHASE
C-----TO FALSE
      CALL  SETLDF
      RETURN
 2010 CONTINUE
C-----GAP-OUT FROM THE CURRENT SIGNAL PHASE (DOG=F, DOR=T, AND TR LE 0)
C-----SET THE STARTING INDEX NUMBER FOR THE LPHNXT ARRAY OF /PHASES/
C-----THAT THE NEXT SIGNAL PHASE FINDER WILL USE TO 1 (START THE AT
C-----BEGINNING OF THE LPHNXT ARRAY)
      IPCLTO = 1
C/                  IF ( TIME . LT . TPRINT )      GO TO 104
C/    PRINT 704 , ICPHAS,IDUALL(ICPHAS)
C/104 CONTINUE
                    IF ( TIME . LE . STRTIM )      GO TO 2020
C-----INCREMENT THE NUMBER OF GAP-OUTS FOR THIS SIGNAL PHASE AND ADD THE
C-----TIME INTO THIS SIGNAL PHASE FOR THE AVERAGE TIME INTO THE SIGNAL
C-----PHASE FOR GAP-OUT
      NGAPO(ICPHAS) = NGAPO(ICPHAS) + 1
      TGAPO(ICPHAS) = TGAPO(ICPHAS) + TP
 2020 CONTINUE
                    IF ( IDUALL(ICPHAS).EQ.IYES )GO TO 2030
C-----THIS SIGNAL PHASE IS NOT THE DUAL LEFT PHASE THUS SET THE
C-----DETECTORS CONNECTED POSITIVE TO THE CURRENT SIGNAL PHASE TO FALSE
      CALL  SETLDF
      GO TO 4010
 2030 CONTINUE
C-----SET TMAG1 TO THE MINIMUM ASSURED GREEN FOR THE FIRST SINGLE LEFT
C-----SIGNAL PHASE FOLLOWING THE DUAL LEFT SIGNAL PHASE
      TMAG1 = TII(ICPHAS+1) + TVI(ICPHAS+1)
C-----SET TMAG2 TO THE MINIMUM ASSURED GREEN FOR THE SECOND SINGLE LEFT
C-----SIGNAL PHASE FOLLOWING THE DUAL LEFT SIGNAL PHASE
      TMAG2 = TII(ICPHAS+2) + TVI(ICPHAS+2)
                    IF ( TMAG2 - TMAG1 )           2040 , 4010 , 2050
 2040 CONTINUE
C-----TMAG1 IS LONGER THAN TMAG2 THUS IF THE TIME INTO THE SIGNAL PHASE
C-----IS GE TMAG1 THEN GO TO 4010 AND FIND THE NEXT SIGNAL PHASE ELSE
C-----SET THE NEXT SIGNAL PHASE TO THE FIRST SINGLE LEFT TURN SIGNAL
C-----PHASE (THE MINIMUM ASSURED GREEN FOR THE FIRST SINGLE LEFT TURN
C-----SIGNAL PHASE HAS NOT BEEN SATISFIED)
                    IF ( TP . GE . TMAG1 )         GO TO 4010
      NEXTPH = ICPHAS + 1
      I = 1
      GO TO 2060
 2050 CONTINUE
C-----TMAG2 IS LONGER THAN TMAG1 THUS IF THE TIME INTO THE SIGNAL PHASE
C-----IS GE TMAG2 THEN GO TO 4010 AND FIND THE NEXT SIGNAL PHASE ELSE
C-----SET THE NEXT SIGNAL PHASE TO THE SECOND SINGLE LEFT TURN SIGNAL
C-----PHASE (THE MINIMUM ASSURED GREEN FOR THE SECOND SINGLE LEFT TURN
C-----SIGNAL PHASE HAS NOT BEEN SATISFIED)
                    IF ( TP . GE . TMAG2 )         GO TO 4010
      NEXTPH = ICPHAS + 2
      I = 2
 2060 CONTINUE
C/                  IF ( TIME . LT . TPRINT )      GO TO 105
C/    PRINT 705 , ICPHAS,TP,TMAG1,TMAG2,NEXTPH,I
C/105 CONTINUE
C-----SET THE FLAG FOR MINIMUM ASSURED GREEN HAS NOT BEEN SATISFIED AND
C-----ENTER THE AMBER CLEARANCE INTERVAL
      MAGSAT = LFALSE
      GO TO 4050
 3010 CONTINUE
C-----MAX-OUT FROM THE CURRENT SIGNAL PHASE (DOG=DOR=T AND TP GE EOM)
C-----SET THE STARTING INDEX NUMBER FOR THE LPHNXT ARRAY OF /PHASES/
C-----THAT THE NEXT SIGNAL PHASE FINDER WILL USE TO 1 (START THE AT
C-----BEGINNING OF THE LPHNXT ARRAY)
      IPCLTO = 1
```

```
C/                  IF ( TIME . LT . TPRINT )     GO TO 106
C/    PRINT 706 , ICPHAS,IDUALL(ICPHAS),TMX(ICPHAS+1),TMX(ICPHAS+2)
C/106 CONTINUE
                    IF ( TIME . LE . STRTIM )     GO TO 3020
C-----INCREMENT THE NUMBER OF MAX-OUTS FOR THIS SIGNAL PHASE AND ADD THE
C-----TIME INTO THIS SIGNAL PHASE FOR THE AVERAGE TIME INTO THE SIGNAL
C-----PHASE FOR MAX-OUT
      NMAXO(ICPHAS) = NMAXO(ICPHAS) + 1
      TMAXO(ICPHAS) = TMAXO(ICPHAS) + TP
 3020 CONTINUE
C-----IF THIS SIGNAL PHASE IS NOT A DUAL LEFT SIGNAL PHASE THEN GO TO
C-----4010 AND FIND THE NEXT SIGNAL PHASE
                    IF ( IDUALL(ICPHAS).EQ.INO ) GO TO 4010
C-----THE CURRENT SIGNAL PHASE IS A DUAL LEFT SIGNAL PHASE THUS SET THE
C-----STARTING INDEX NUMBER FOR THE LPHNXT ARRAY OF /PHASES/ THAT THE
C-----NEXT SIGNAL PHASE FINDER WILL USE TO 3 (SKIP BOTH SINGLE LEFT TURN
C-----SIGNAL PHASES AFTER THE DUAL LEFT SIGNAL PHASE MAX-OUT)
      IPCLTO = 3
            IF ( TMX(ICPHAS+2) - TMX(ICPHAS+1) ) 3030 , 4010 , 3040
 3030 CONTINUE
C-----THE MAXIMUM EXTENSION AFTER DEMAND ON RED FOR THE FIRST SINGLE
C-----LEFT TURN SIGNAL PHASE IS GT THE MAXIMUM EXTENSION AFTER DEMAND ON
C-----RED FOR THE SECOND SINGLE LEFT TURN SIGNAL PHASE THUS SET THE NEXT
C-----SIGNAL PHASE TO THE FIRST SINGLE LEFT TURN SIGNAL PHASE AND ENTER
C-----THE AMBER CLEARANCE INTERVAL
      NEXTPH = ICPHAS + 1
      I = 1
      GO TO 4050
 3040 CONTINUE
C-----THE MAXIMUM EXTENSION AFTER DEMAND ON RED FOR THE SECOND SINGLE
C-----LEFT TURN SIGNAL PHASE IS GT THE MAXIMUM EXTENSION AFTER DEMAND ON
C-----RED FOR THE FIRST SINGLE LEFT TURN SIGNAL PHASE THUS SET THE NEXT
C-----SIGNAL PHASE TO THE SECOND SINGLE LEFT TURN SIGNAL PHASE AND ENTER
C-----THE AMBER CLEARANCE INTERVAL
      NEXTPH = ICPHAS + 2
      I = 2
      GO TO 4050
 4010 CONTINUE
C-----FORCED CLEARANCES HAVE NOT BEEN MANDATED THUS CHECK EACH SIGNAL
C-----PHASE THAT THIS SIGNAL PHASE CAN CLEAR TO STARTING AT IPCLTO AND
C-----SET THE NEXT SIGNAL PHASE TO THE FIRST SIGNAL PHASE ON THE LIST OF
C-----SIGNAL PHASES THAT THIS SIGNAL PHASE CAN CLEAR TO WHICH HAS DEMAND
C-----FOR THE SIGNAL PHASE
      NPCLTO = NPHNXT(ICPHAS)
C/                  IF ( TIME . LT . TPRINT )     GO TO 107
C/    PRINT 707 , ICPHAS,NPCLTO,(LPHNXT(I,ICPHAS),I=1,NPCLTO)
C/107 CONTINUE
      DO 4020  I = IPCLTO , NPCLTO
      NEXTPH = LPHNXT(I,ICPHAS)
C-----IF THE SKIP PHASE SWITCH FOR THE NEXTPH SIGNAL PHASE IS OFF THEN
C-----THAT SIGNAL PHASE CAN NOT BE SKIPPED THUS GO TO 4030 AND USE THE
C-----NEXTPH SIGNAL PHASE
                    IF ( ISKP(NEXTPH).EQ.IOFF )  GO TO 4030
C-----CHECK THE DEMAND FOR THE NEXTPH SIGNAL PHASE USING BOTH THE
C-----POSITIVE AND NEGATIVE DETECTOR CONNECTIONS AND RETURN THE DEMAND
C-----FOR THE NEXTPH SIGNAL PHASE
      CALL  CHKDFP  ( IDFP,NEXTPH,2 )
C-----IF THERE IS DEMAND FOR THE NEXTPH SIGNAL PHASE THEN GO TO 4030 AND
C-----USE THE NEXTPH SIGNAL PHASE
                    IF ( IDFP )                  GO TO 4030
 4020 CONTINUE
C-----IN THE ABSENCE OF DEMAND THE SIGNAL SHOULD GO TO THE LAST SIGNAL
C-----PHASE ON THE LIST OF SIGNAL PHASES THAT THIS SIGNAL PHASE CAN
C-----CLEAR TO
      I = NPCLTO
C-----IF THIS SIGNAL PHASE IS THE MINOR SIGNAL PHASE FOR THE
C-----PARENT/MINOR CASE THEN USE THE LAST SIGNAL PHASE ELSE ERROR
            IF ( IMINOR(ICPHAS) . EQ . IYES )    GO TO 4030
      GO TO 9210
 4030 CONTINUE
                    IF ( IDUALL(ICPHAS).EQ.INO ) GO TO 4040
```

```
C-----THE CURRENT SIGNAL PHASE IS A DUAL LEFT SIGNAL PHASE THUS IF THE
C-----NEXT SIGNAL PHASE IS ONE OF THE SINGLE LEFT TURN SIGNAL PHASES
C-----THEN DO NOT RESET END OF MAX TO NOT SET (KEEP THE CLOCK RUNNING)
                    IF ( NEXTPH . EQ . ICPHAS+1 )GO TO 4050
                    IF ( NEXTPH . EQ . ICPHAS+2 )GO TO 4050
 4040 CONTINUE
C-----RESET THE END OF MAX TO NOT SET
      EOM = TBIG
 4050 CONTINUE
C-----BEGIN THE AMBER CLEARANCE INTERVAL
      TR = TCI(ICPHAS)
      ICAMPC = ICAMPC + I
      INTER = 2
C/                  IF ( TIME . LT . TPRINT )     GO TO 108
C/    PRINT 708 , NGAPO(ICPHAS),NMAXO(ICPHAS),TCI(ICPHAS),TAR(ICPHAS),
C/   *            NEXTPH,I
C/108 CONTINUE
 5010 CONTINUE
C-----THE SIGNAL IS IN THE AMBER CLEARANCE INTERVAL THUS IF THE TIME
C-----REMAINING IN THIS INTERVAL IS GT 0 THEN RETURN
                    IF ( TR . GT . 0.0 )          RETURN
C-----BEGIN THE ALL-RED CLEARANCE INTERVAL
      TR = TAR(ICPHAS)
      ICAMPC = ICAMPS(ICPHAS) + NPHNXT(ICPHAS) + 1
      INTER = 3
 6010 CONTINUE
C-----THE SIGNAL IS IN THE ALL-RED CLEARANCE INTERVAL THUS IF THE TIME
C-----REMAINING IN THIS INTERVAL IS GT 0 THEN RETURN
                    IF ( TR . GT . 0.0 )          RETURN
C-----BEGIN THE GREEN INTERVAL ON THE NEW PHASE
      IOPHAS = ICPHAS
      ICPHAS = NEXTPH
      ICAMPC = ICAMPS(NEXTPH)
      INTER = 1
C8    DTIME = TIME + DT
C8    PRINT 601 , ICPHAS,ICAMPC,DTIME
C4                  IF ( TIME . LT . TPRINT )     GO TO 109
C4    PRINT 709 , ICAMPC,ICPHAS,IDUALL(ICPHAS),TII(ICPHAS),TVI(ICPHAS),
C4   *            TCI(ICPHAS),TAR(ICPHAS)
C4109 CONTINUE
C-----INITIALIZE THE DEMAND ON RED FOR THE NEW SIGNAL PHASE TO FALSE
      IDOR = .FALSE.
C-----IF END OF MAX HAS BEEN RESET TO NOT SET THEN GO TO 6020 AND SET
C-----THE TIME INTO THE NEW SIGNAL PHASE TO ZERO AND THE TIME REMAINING
C-----IN THE NEW SIGNAL PHASE TO THE INITIAL INTERVAL PLUS THE VEHICLE
C-----INTERVAL FOR THE NEW SIGNAL PHASE
                    IF ( EOM . EQ . TBIG )        GO TO 6020
C-----THE NEW SIGNAL PHASE IS A SINGLE LEFT TURN SIGNAL PHASE FOLLOWING
C-----THE DUAL LEFT SIGNAL PHASE THUS RESET END OF MAX FOR THE MAXIMUM
C-----EXTENSION AFTER DEMAND ON RED FOR THE NEW SIGNAL PHASE (THE TIME
C-----INTO THE SIGNAL PHASE HAS CONTINUED TO BE UPDATED EACH DT DURING
C-----THE AMBER CLEARANCE AND THE ALL-RED CLEARANCE INTERVAL)
      EOM = EOM - TMX(IOPHAS) + TMX(NEXTPH)
C-----IF THE MINIMUM ASSURED GREEN HAS BEEN SATISFIED THEN GO TO 1010
C-----AND CHECK THE GREEN INTERVAL FOR THE NEW SIGNAL PHASE (THE NEW
C-----SIGNAL PHASE MAY HAVE MAX-OUT OR GAP-OUT DURING THE AMBER
C-----CLEARANCE OR THE ALL-RED CLEARANCE INTERVAL)
                    IF ( MAGSAT . EQ . LTRUE )    GO TO 1010
C-----SET THE TIME REMAINING FOR THE NEW SIGNAL PHASE AND SET THAT
C-----MINIMUM ASSURED GREEN HAS BEEN SATISFIED AND GO TO 1010 AND CHECK
C-----THE GREEN INTERVAL FOR THE NEW SIGNAL PHASE (THE NEW SIGNAL PHASE
C-----MAY HAVE MAX-OUT OR GAP-OUT DURING THE AMBER CLEARANCE OR THE ALL-
C-----RED CLEARANCE INTERVAL)
      TR = TII(NEXTPH) + TVI(NEXTPH) - TP
      MAGSAT = LTRUE
      GO TO 1010
 6020 CONTINUE
C-----SET THE TIME INTO THE NEW SIGNAL PHASE TO ZERO AND THE TIME
C-----REMAINING IN THE NEW SIGNAL PHASE TO THE INITIAL INTERVAL PLUS THE
C-----VEHICLE INTERVAL FOR THE NEW SIGNAL PHASE
      TP = 0.0
```

```
      TR = TII(NEXTPH) + TVI(NEXTPH)
      RETURN
C-----PROCESS THE EXECUTION ERROR AND STOP
 9210 CONTINUE
      CALL  ABORTR  ( MSG921,40 )
      STOP  921
      END
```

ACTSIG

```
      SUBROUTINE CHKDFP ( IDFP,IP,ITYPE )
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINUR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACTSI(11),ILD,JLD,NUMLD,VSETLD(3),ZTEMPD(93)
      LOGICAL           IDFP
      DATA      ION    / 3HON  /
      DATA      JAND   / 3HAND /
      DATA      N1,N2  / 4HCHKD,2HFP /
C/701 FORMAT(17H DEMAND FOR PHASE,I2,4H IS ,L1,
C/   *        23H DETECTOR CONNECTION = ,A3,8H NUMLD =,I3,6H LLD =,10I4)
C/702 FORMAT(17H DEMAND FOR PHASE,I2,4H IS ,L1)
C
C-----SUBROUTINE CHKDFP CHECKS THE DEMAND FOR THE IP SIGNAL PHASE
C-----(WHEN ITYPE IS EQ 1 THEN ONLY THE POSITIVE DETECTOR CONNECTIONS
C-----ARE CHECKED AND WHEN ITYPE IS EQ 2 THEN BOTH THE POSITIVE AND
C-----NEGATIVE CONNECTIONS ARE CHECKED)
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----INITIALIZE SOME PARAMETERS FOR CHKDFP
      NUMLD = NLD(IP)
      IDFP = .TRUE.
C-----IF THE RECALL SWITCH IN ON THEN THERE IS DEMAND FOR THE IP SIGNAL
C-----PHASE THUS GO TO 3010 AND FINISH PROCESSING
                      IF ( IREC(IP) . EQ . ION )   GO TO 3010
C-----INITIALIZE THE DEMAND FOR THE IP SIGNAL PHASE TO THE VALUE OF
C-----THE FIRST DETECTOR CONNECTED TO THE IP SIGNAL PHASE (THE FIRST
C-----DETECTOR CONNECTED TO ANY SIGNAL PHASE MUST BE POSITIVE)
      JLD = LLD(1,IP)
                      IF ( JLD . LT . 0 )          GO TO 1010
      IDFP = LDTRIP(JLD)
      GO TO 1020
 1010 CONTINUE
                      IF ( ITYPE . EQ . 1 )        GO TO 3010
      IDFP = . NOT . LDTRIP(-JLD)
 1020 CONTINUE
C-----IF THERE WAS ONLY ONE DETECTOR CONNECTED TO THE IP SIGNAL PHASE
C-----THEN GO TO 3010 AND FINISH PROCESSING
                      IF ( NUMLD . LT . 2 )        GO TO 3010
C-----CHECK EACH DETECTOR CONNECTED TO THE IP SIGNAL PHASE (START AT THE
C-----SECOND DETECTOR FOR THE IP SIGNAL PHASE BECAUSE THE FIRST DETECTOR
C-----HAS BEEN USED TO INITIALIZE THE VALUE FOR IDFP)
      DO 2040  ILD = 2 , NUMLD
      JLD = LLD(ILD,IP)
C-----IF THE DETECTOR CONNECTION TYPE IS AND THEN GO TO 2020 AND PROCESS
C-----THE AND CONNECTIONS ELSE PROCESS THE OR CONNECTIONS
                      IF ( IANDOR(IP).EQ.JAND )    GO TO 2020
C-----IF THE DETECTOR IS A NEGATIVE CONNECTION THEN GO TO 2010 AND
C-----PROCESS THE NEGATIVE CONNECTION ELSE PROCESS THE POSITIVE
C-----CONNECTION
                      IF ( JLD . LT . 0 )          GO TO 2010
      IDFP = IDFP . OR . LDTRIP(JLD)
      GO TO 2040
 2010 CONTINUE
C-----IF ONLY THE POSITIVE CONNECTIONS ARE TO BE CHECKED THEN GO TO 2040
C-----AND SKIP TO THE NEXT DETECTOR
                      IF ( ITYPE . EQ . 1 )        GO TO 2040
      IDFP = IDFP . OR . ( . NOT . LDTRIP(-JLD) )
      GO TO 2040
 2020 CONTINUE
```

```
C-----PROCESS THE AND CONNECTION THUS IF THE DETECTOR IS A NEGATIVE
C-----CONNECTION THEN GO TO 2030 AND PROCESS THE NEGATIVE CONNECTION
C-----ELSE PROCESS THE POSITIVE CONNECTION
                      IF ( JLD . LT . 0  )          GO TO 2030
      IDFP = IDFP . AND . LDTRIP(JLD)
      GO TO 2040
 2030 CONTINUE
C-----IF ONLY THE POSITIVE CONNECTIONS ARE TO BE CHECKED THEN GO TO 2040
C-----AND SKIP TO THE NEXT DETECTOR
                      IF ( ITYPE . EQ . 1 )         GO TO 2040
      IDFP = IDFP . AND . ( . NOT . LDTRIP(-JLD) )
C-----END OF DETECTOR LOOP
 2040 CONTINUE
 3010 CONTINUE
C-----FINISH PROCESSING
C/                    IF ( TIME . LT . TPRINT )     GO TO 102
C/                    IF ( NUMLD . LE . 0 )         GO TO 101
C/    PRINT 701 , IP,IDFP,IANDOR(IP),NUMLD,(LLD(ILD,IP),ILD=1,NUMLD)
C/    GO TO 102
C/101 CONTINUE
C/    PRINT 702 , IP,IDFP
C/102 CONTINUE
      RETURN
      END                                                          CHKDFP
```

```
      SUBROUTINE SETLDF
      COMMON / LOOPS  / STRTLD(20),STOPLD(20),LDTRIP(20),ITYPLD(20),
     *                  NLOOPS,LLOOPS(20)
      LOGICAL           LDTRIP
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VACTSI(11),VCHKDF(3),ILD,JLD,NUMLD,ZTEMPD(93)
      DATA      N1,N2   / 4HSETL,2HDF /
C/701 FORMAT(31H MEMORY FOR DETECTORS FOR PHASE,I2,10H SET FALSE)
C
C-----SUBROUTINE SETLDF SETS THE DETECTORS CONNECTED POSITIVE TO THE
C-----CURRENT SIGNAL PHASE TO FALSE
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
      NUMLD = NLD(ICPHAS)
C-----IF THERE ARE NO DETECTORS CONNECTED TO THE CURRENT SIGNAL PHASE
C-----THEN RETURN
                      IF ( NUMLD . LT . 1 )         RETURN
C-----CHECK EACH DETECTOR CONNECTED TO THE CURRENT SIGNAL PHASE
      DO 1010  ILD = 1 , NUMLD
      JLD = LLD(ILD,ICPHAS)
C-----IF DETECTOR JLD IS NOT CONNECTED POSITIVE TO THE CURRENT SIGNAL
C-----PHASE THEN GO TO 1010 AND SKIP TO THE NEXT DETECTOR ELSE SET
C-----DETECTOR JLD TO FALSE
                      IF ( JLD . LT . 0 )           GO TO 1010
      LDTRIP(JLD) = .FALSE.
 1010 CONTINUE
C/                    IF ( TIME . LT . TPRINT )     GO TO 101
C/    PRINT 701 , ICPHAS
C/101 CONTINUE
      RETURN
      END                                                          SETLDF
```

```
C9     SUBROUTINE INTSTA ( IPAGE )
C9     COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
C9    *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
C9    *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
C9    *                  LIBAR(12),LOBAR(12)
C9     COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
       COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
C9    *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
C9    *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
C9    *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
C9    *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
C9    *                  PLVDV(6),NLVDV(6),TMTIME(5)
C9     COMMON / TITLE  / ITITLE(20)
C9     COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
C9    *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
C9    *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
C9     COMMON / ZTEMPD / II,I1TO3,KK,MIBA,NUM,OASD,PDELAY,PTURN,SUMDEL,
C9    *                  SUMVOL,TIMNOW,TOTDEL,TOTVOL,VOLUME,ZTEMPD(96)
C9     DIMENSION         IPTURN(3,3)
C9     DATA      IPTURN / 4HU AN,4HD LE,4HFT  ,
C9    *                  4HSTRA,4HIGHT,4H    ,
C9    *                  4HRIGH,4HT   ,4H    /
C9     DATA      N1,N2  / 4HINTS,2HTA /
C9601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
C9    *        14HLATION PACKAGE,//,1X,20A4,//,23H TIME INTO SIMULATION =,
C9    *        F8.1,8H SECONDS/)
C9602 FORMAT(/,40H SUMMARY STATISTICS FOR INBOUND APPROACHI3)
C9603 FORMAT(9H VOLUME =F6.1,8H  OASD =F6.1,9H  PTURN =F6.1,
C9    *        10H  PDELAY =F6.1,2X,3A4)
C9604 FORMAT(9H VOLUME =F6.1,8H  OASD =F6.1,14H  FOR APPROACH)
C9605 FORMAT(9H0VOLUME =F6.1,8H  OASD =F6.1,18H  FOR INTERSECTION)
C9606 FORMAT(26H-TMTIME SINCE LAST CALL = ,F7.2,14H TOTAL TMTIME ,
C9    *        29HSINCE END OF START-UP TIME = ,F7.2,7H (SECS))
C9607 FORMAT(I1)
C
C-----SUBROUTINE INTSTA PRINTS THE INTERMEDIATE STATISTICS
C
C9     NRNAME = NRNAME + 1
C9     IRNAME(1,NRNAME) = N1
C9     IRNAME(2,NRNAME) = N2
C9                   IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----FIND THE TIME INTO THE SIMULATION SINCE START-UP TIME
C9     TIMNOW = TIME - STRTIM
C-----IF THE TIME INTO THE SIMULATION SINCE START-UP TIME IS LE ONE DT
C-----THEN RETURN
C9                   IF ( TIMNOW . LE . DT )       RETURN
C9                   IF ( TMTIME(5) . GT . 0.0 )  GO TO 101
C9     TMTIME(5) = TMTIME(3)
C9101 CONTINUE
C9     TMTIME(4) = TMTIME(5)
C9     CALL  EXTIME  ( 5 )
C9     PRINT 601 , ITITLE,TIMNOW
C9     SUMVOL = 0.0
C9     SUMDEL = 0.0
C-----PROCESS EACH INBOUND APPROACH
C9     DO 104  II = 1 , NIBA
C9     MIBA = LIBA(II)
C-----FIND THE TOTAL VOLUME AND TOTAL STOPPED TIME DELAY FOR INBOUND
C-----APPROACH MIBA
C9     TOTVOL = NUMPRO(II,1) + NUMPRO(II,2) + NUMPRO(II,3)
C9     TOTDEL = SD(II,1)      + SD(II,2)      + SD(II,3)
C-----IF THE TOTAL VOLUME FOR INBOUND APPROACH MIBA IS LE 0 THEN GO TO
C-----104 AND SKIP TO THE NEXT INBOUND APPROACH
C9                   IF ( TOTVOL . LE . 0.0 )      GO TO 104
C9     PRINT 602 , MIBA
C-----PROCESS EACH TURN CODE FOR INBOUND APPROACH MIBA
C9     DO 103  KK = 1 , 3
C9     NUM = NUMPRO(II,KK)
C-----IF THE NUMBER OF VEHICLES PROCESSED FOR TURN CODE KK AND INBOUND
C-----APPROACH MIBA IS LE 0 THEN GO TO 103 AND SKIP TO THE NEXT TURN
C-----CODE
C9                   IF ( NUM . LE . 0 )           GO TO 103
C-----FIND THE EQUIVALENT HOURLY VOLUME PROCESSED
C9     VOLUME = NUM/(TIMNOW/3600.0)
C-----FIND THE OVERALL AVERAGE STOPPED DELAY
C9     OASD = SD(II,KK)/NUM
C-----FIND THE PERCENT OF VEHICLES MAKING TURN CODE KK FOR INBOUND
C-----APPROACH MIBA
C9     PTURN = 100.0*NUM/TOTVOL
C9     PDELAY = 0.0
C9                   IF ( TOTDEL . LE . 0.0 )      GO TO 102
C-----FIND THE PERCENT STOPPED DELAY FOR TURN CODE KK FOR INBOUND
C-----APPROACH MIBA
C9     PDELAY = 100.0*SD(II,KK)/TOTDEL
C9102 CONTINUE
C9     PRINT 603 , VOLUME,OASD,PTURN,PDELAY,(IPTURN(I1TO3,KK),I1TO3=1,3)
C-----END OF TURN CODE LOOP
C9103 CONTINUE
C-----FIND THE OVERALL AVERAGE STOPPED DELAY FOR INBOUND APPROACH MIBA
C9     OASD = TOTDEL/TOTVOL
C9     SUMDEL = SUMDEL + TOTDEL
C9     SUMVOL = SUMVOL + TOTVOL
C-----FIND THE EQUIVALENT HOURLY VOLUME PROCESSED FOR INBOUND APPROACH
C-----MIBA
C9     TOTVOL = TOTVOL/(TIMNOW/3600.0)
C9     PRINT 604 , TOTVOL,OASD
C-----END OF INBOUND APPROACH LOOP
C9104 CONTINUE
C9     OASD = SUMDEL/SUMVOL
C9     TOTVOL = SUMVOL/(TIMNOW/3600.0)
C9     PRINT 605 , TOTVOL,OASD
C9     TMINT = TMTIME(5) - TMTIME(4)
C9     TMSIM = TMTIME(5) - TMTIME(3)
C9     PRINT 606 , TMINT,TMSIM
C9     PRINT 607 , IPAGE
C9     RETURN
C9     END                                                              INTSTA
```

```
      SUBROUTINE SUMARY
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / APLVDV,IAN,II,ITC,JA,LANE,MPLVDV,NUMC,NUME,
     *                  NUMTA,PTURN(3),QUEUEL,TPLVDV,VPSTAT(27),
     *                  VADDST(1),VACTST(7),VTIMST(16),ZTEMPD(44)
      DIMENSION         IPTURN(3,3)
      DATA      IPTURN / 4HU AN,4HD LE,4HFT   ,
     *                   4HSTRA,4HIGHT,4H     ,
     *                   4HRIGH,4HT   ,4H     /
      DATA      NINE   / 9 /
      DATA      NYES   / 3HYES /
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *       14HLATION PACKAGE,//,1X,20A4,/)
  602 FORMAT(40H SUMMARY STATISTICS FOR INBOUND APPROACHI3,9H FOR TURN,
     *       8H CODE = 3A4)
  603 FORMAT(/,
     *54H PERCENT OF APPROACH VEHICLES MAKING MOVEMENT ------ =F9.1/)
  604 FORMAT(40H SUMMARY STATISTICS FOR INBOUND APPROACHI3,/)
  605 FORMAT(/,
     *54H PERCENT OF VEHICLES MAKING A U-TURN OR A LEFT TURN  =F9.1/,
     *54H PERCENT OF VEHICLES GOING STRAIGHT ---------------- =F9.1/,
     *54H PERCENT OF VEHICLES MAKING A RIGHT TURN ----------- =F9.1/)
  606 FORMAT(30H AVERAGE QUEUE LENGTH FOR LANEI2,1H ,19(1H-),2H =,F9.1,
     *       7H  MAX =,I3)
  607 FORMAT(
     *54H NUMBER OF COLLISIONS ---------------------------- =I7)
  608 FORMAT(
     *54H NUMBER OF VEHICLES ELIMINATED (LANE FULL) --------- =I7)
  609 FORMAT(
     *54H AVERAGE OF LOGIN SPEED/DESIRED SPEED (PERCENT) ---- =F9.1)
  610 FORMAT(38H SUMMARY STATISTICS FOR ALL APPROACHES//)
  701 FORMAT(5F6.1,2F6.3,F6.1,I2,2(1X,A3),2F5.2,2I3)
  702 FORMAT(I2,I1)
C
C-----SUBROUTINE SUMARY PRINTS THE SUMMARY STATISTICS
C
C-----GET THE TM TIME FOR THIS JOB AT THE END OF SIMULATION TIME
      CALL  EXTIME  ( 4 )
C-----FIND THE ACTUAL SIMULATION TIME
      SIMTIM = TIME - STRTIM - DT
                      IF ( IPUNCH . NE . NYES )     GO TO 1010
      TLEAD = TLEAD + APIJR
      TLAG  = TLAG  + APIJR
      WRITE (7,701)  STRTIM,SIMTIM,DT,XFPS,XQDIST,CAREQL,CAREQM,CAREQA,
     *               ICONTR,IPTC,IPAP,TLEAD,TLAG,NIBA,NPHASE
C-----INITIALIZE SOME PARAMETERS FOR SUMARY
 1010 CONTINUE
      NUME = 0
      NUMC = 0
      TPLVDV = 0.0
      MPLVDV = 0
C-----PROCESS EACH INBOUND APPROACH
      DO 2060  IAN = 1 , NIBA
      JA = LIBA(IAN)
C-----SUM THE TOTAL NUMBER OF COLLISIONS AND VEHICLES ELIMINATED FOR THE
C-----INTERSECTION
      NUMC = NUMC + NBANG(IAN)
      NUME = NUME + NELIM(IAN)
C-----SUM THE PERCENT LOG IN VELOCITY TO DESIRED SPEED FOR THE
C-----INTERSECTION
      TPLVDV = TPLVDV + PLVDV(IAN)
      MPLVDV = MPLVDV + NLVDV(IAN)
C-----FIND THE NUMBER OF VEHICLES PROCESSED FOR INBOUND APPROACH JA
      NUMTA = NUMPRO(IAN,1) + NUMPRO(IAN,2) + NUMPRO(IAN,3)
C-----IF NO VEHICLES WERE PROCESSED FOR INBOUND APPROACH JA THEN GO TO
C-----2050 AND SKIP TO THE NEXT INBOUND APPROACH
                      IF ( NUMTA . LE . 0 )         GO TO 2050
C-----PROCESS EACH TURN CODE
      DO 1050  ITC = 1 , 3
C-----FIND THE ACTUAL NUMBER OF VEHICLES PROCESSED DURING START-UP TIME
      NUMPSU = NUMPSU - NUMPRO(IAN,ITC)
C-----FIND THE PERCENT OF VEHICLES ON INBOUND APPROACH JA MAKING TURN
C-----CODE ITC
      PTURN(ITC) = 100.0*NUMPRO(IAN,ITC)/NUMTA
C-----IF NO VEHICLES WERE PROCESSED FOR INBOUND APPROACH JA AND TURN
C-----CODE ITC THEN GO TO 1040 AND SKIP TO THE NEXT TURN CODE
                      IF ( NUMPRO(IAN,ITC).LE.0 )  GO TO 1040
C-----IF SUMMARY STATISTICS WERE NOT REQUESTED TO BE PRINTED BY TURN
C-----CODE THEN GO TO 1020 AND CONTINUE ELSE PRINT THE SUMMARY
C-----STATISTICS BY TURN CODE
                      IF ( IPTC . NE . NYES )       GO TO 1020
      PRINT 601 , ITITLE
      PRINT 602 , JA,(IPTURN(II,ITC),II=1,3)
C-----PRINT SUMMARY STATISTICS FOR INBOUND APPROACH IAN AND TURN CODE
C-----ITC AND OPTIONALLY WRITE THE STATISTICS ONTO TAPE 7 USING APPROACH
C-----NUMBER JA AND TURN CODE ITC
 1020 CONTINUE
      CALL  PSTATS  ( IAN,ITC,JA,ITC,IPTC )
                      IF ( IPTC . NE . NYES )       GO TO 1030
      PRINT 603 , PTURN(ITC)
 1030 CONTINUE
C-----IF THIS IS THE FIRST TURN CODE THEN GO TO 1040 AND SKIP TO THE
C-----NEXT TURN CODE
                      IF ( ITC . EQ . 1 )           GO TO 1040
C-----ADD THE SUMMARY STATISTICS FROM (IAN,ITC) TO (IAN,1) (SUM FOR TURN
C-----TURN CODE)
      CALL  ADDSTA  ( IAN,IAN,ITC )
 1040 CONTINUE
C-----IF NO VEHICLES WERE PROCESSED FOR TURN CODE ITC THEN WRITE A DUMMY
C-----CARD ONTO TAPE 7
      IF ( IPUNCH.EQ.NYES . AND . NUMPRO(IAN,ITC).LE.0 )
     *WRITE (7,702)  JA,ITC
C-----END OF TURN CODE LOOP
 1050 CONTINUE
C-----IF SUMMARY STATISTICS WERE NOT REQUESTED TO BE PRINTED BY APPROACH
C-----THEN GO TO 2010 AND CONTINUE ELSE PRINT THE SUMMARY STATISTICS FOR
C-----INBOUND APPROACH JA
                      IF ( IPAP . NE . NYES )       GO TO 2010
      PRINT 601 , ITITLE
      PRINT 604 , JA
C-----PRINT SUMMARY STATISTICS FOR INBOUND APPROACH IAN AND TURN CODE 1
C-----(SUM FOR ALL THE TURN CODES) AND OPTIONALLY WRITE THE STATISTICS
C-----ONTO TAPE 7 USING APPROACH NUMBER JA AND TURN CODE 9 (SUM FOR ALL
C-----THE TURN CODES)
 2010 CONTINUE
      CALL  PSTATS  ( IAN,1,JA,9,IPAP )
                      IF ( IPAP . NE . NYES )       GO TO 2040
      PRINT 605 , PTURN
C-----PROCESS EACH LANE OF INBOUND APPROACH JA FOR QUEUE LENGTH
      DO 2020  LANE = 1 , 6
C-----IF THERE WERE NO VEHICLES QUEUED FOR INBOUND APPROACH JA AND LANE
C-----LANE THEN GO TO 2020 AND SKIP TO THE NEXT LANE
                      IF ( LQUEUE(IAN,LANE).LE.0 ) GO TO 2020
C-----FIND THE AVERAGE QUEUE LENGTH FOR INBOUND APPROACH JA AND LANE
C-----LANE (TIME AVERAGE OVER THE ENTIRE SIMULATION TIME)
```

```
      QUEUEL = LQUEUE(IAN,LANE)*DT/SIMTIM
      PRINT 606 , LANE,QUEUEL,MQUEUE(IAN,LANE)
 2020 CONTINUE
C-----PRINT THE NUMBER OF COLLISIONS AND VEHICLES ELIMINATED FROM THE
C-----SIMULATION FOR INBOUND APPROACH JA
      PRINT 701
      IF ( NBANG(IAN) . GT . 0 )
     *PRINT 607 , NBANG(IAN)
      IF ( NELIM(IAN) . GT . 0 )
     *PRINT 608 , NELIM(IAN)
 2030 CONTINUE
                      IF ( NLVDV(IAN) . LE . 0 )   GO TO 2040
C-----FIND THE AVERAGE PERCENT LOG IN VELOCITY PER DESIRED SPEED FOR
C-----INBOUND APPROACH JA
      APLVDV = 100.0*PLVDV(IAN)/NLVDV(IAN)
      PRINT 609 , APLVDV
 2040 CONTINUE
C-----IF THIS IS THE FIRST INBOUND APPROACH THEN GO TO 2050 AND SKIP TO
C-----THE NEXT INBOUND APPROACH
                      IF ( IAN . EQ . 1 )          GO TO 2050
C-----ADD THE SUMMARY STATISTICS FROM (IAN,1) TO (1,1) (SUM FOR
C-----APPROACH)
      CALL  ADDSTA  ( 1,IAN,1 )
 2050 CONTINUE
C-----IF NO VEHICLES WERE PROCESSED FOR APPROACH JA THEN WRITE 4 DUMMY
C-----CARDS ONTO TAPE 7
      IF ( IPUNCH.EQ.NYES . AND . NUMTA.LE.0 )
     *WRITE (7,702)   ((JA,ITC),ITC=1,3),JA,NINE
C-----END OF INBOUND APPROACH LOOP
 2060 CONTINUE
C-----IF NO VEHICLES WERE PROCESSED FOR THE INTERSECTION THEN GO TO 3020
C-----AND FINISH PROCESSING
                      IF ( NUMPRO(1,1) . LE . 0 )  GO TO 3020
      PRINT 601 , ITITLE
      PRINT 610
C-----PRINT SUMMARY STATISTICS FOR INBOUND APPROACH 1 AND TURN CODE 1
C-----(THE INTERSECTION) AND OPTIONALLY WRITE THE STATISTICS ONTO TAPE 7
C-----USING APPROACH NUMBER 99 AND TURN CODE 9 (THE INTERSECTION)
      CALL  PSTATS  ( 1,1,99,9,NYES )
C-----PRINT THE NUMBER OF COLLISIONS AND VEHICLES ELIMINATED FOR THE
C-----INTERSECTION
      PRINT 701
      IF ( NUMC . GT . 0 )
     *PRINT 607 , NUMC
      IF ( NUME . GT . 0 )
     *PRINT 608 , NUME
 3010 CONTINUE
                      IF ( MPLVDV . LE . 0 )       GO TO 3020
C-----FIND THE AVERAGE PERCENT OF LOG IN VELOCITY PER DESIRED SPEED FOR
C-----THE INTERSECTION
      APLVDV = 100.0*TPLVDV/MPLVDV
      PRINT 609 , APLVDV
 3020 CONTINUE
                      IF ( ICONTR . LT . 6 )       GO TO 3030
C-----PRINT THE ACTUATED SIGNAL CONTROLLER STATISTICS AND OPTIONALLY
C-----WRITE THE ACTUATED SIGNAL CONTROLLER STATISTICS ONTO TAPE 7
      CALL  ACTSTA
 3030 CONTINUE
C-----PRINT THE COMPUTER TIME STATISTICS
      CALL  TIMSTA
      RETURN
      END                                                               SUMARY
```

```
      SUBROUTINE PSTATS ( I,J,IWIA,IWTC,IPRINT )
      COMMON / SUMSTA / TD(18),NTD(18),QD(18),NQD(18),SD(18),NNVSY,
     *                  NSD(18),DMPH(18),NDMPH(18),VMT(18),
     *                  STIME(18),NUMPRO(18),ASPEED(18),ADESPD(18),
     *                  VMAXA(18),VMAXD(18),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VSUMAR(15),ADMAST,ADMPH,ADSPD,AMAXV,AQD,AQDAST,
     *                  ASD,ASDAST,ASTIM,ATD,ATDAST,AVMT,DMAXV,INDEX,
     *                  NUM,OADMPH,OAQD,OASD,OATD,PDMPH,PQD,PSD,PTD,
     *                  SMSPD,TMSPD,VOLUME,XMPH,VADDST(1),VACTST(7),
     *                  VTIMST(16),ZTEMPD(44)
      DATA      NYES   / 3HYES /
  601 FORMAT(/,
     *54H TOTAL DELAY (VEHICLE-SECONDS) -------------------- =F9.1/,
     *54H NUMBER OF VEHICLES INCURRING TOTAL DELAY ---------- =I7/,
     *54H PERCENT OF VEHICLES INCURRING TOTAL DELAY --------- =F9.1/,
     *54H AVERAGE TOTAL DELAY (SECONDS) --------------------- =F9.1/,
     *54H AVERAGE TOTAL DELAY/AVERAGE TRAVEL TIME ----------- =F9.1,
     * 8H PERCENT,//,
     *54H QUEUE DELAY (VEHICLE-SECONDS) --------------------- =F9.1/,
     *54H NUMBER OF VEHICLES INCURRING QUEUE DELAY ---------- =I7/,
     *54H PERCENT OF VEHICLES INCURRING QUEUE DELAY --------- =F9.1/,
     *54H AVERAGE QUEUE DELAY (SECONDS) --------------------- =F9.1/,
     *54H AVERAGE QUEUE DELAY/AVERAGE TRAVEL TIME ----------- =F9.1,
     * 8H PERCENT,//,
     *54H STOPPED DELAY (VEHICLE-SECONDS) ------------------- =F9.1/,
     *54H NUMBER OF VEHICLES INCURRING STOPPED DELAY -------- =I7/,
     *54H PERCENT OF VEHICLES INCURRING STOPPED DELAY ------- =F9.1/,
     *54H AVERAGE STOPPED DELAY (SECONDS) ------------------- =F9.1/,
     *54H AVERAGE STOPPED DELAY/AVERAGE TRAVEL TIME --------- =F9.1,
     * 8H PERCENT)
  602 FORMAT(/,
     *12H DELAY BELOW,F5.1,37H MPH (VEHICLE-SECONDS) ------------ =F9.1/,
     *41H NUMBER OF VEHICLES INCURRING DELAY BELOW,F5.1,8H MPH - =I7/,
     *42H PERCENT OF VEHICLES INCURRING DELAY BELOW,F5.1,7H MPH  =F9.1/,
     *20H AVERAGE DELAY BELOW,F5.1,29H MPH (SECONDS) ------------ =F9.1/,
     *20H AVERAGE DELAY BELOW,F5.1,29H MPH/AVERAGE TRAVEL TIME -- =F9.1,
     * 8H PERCENT)
  603 FORMAT(/,
     *54H VEHICLE-MILES OF TRAVEL -------------------------- =F11.3/,
     *54H AVERAGE VEHICLE-MILES OF TRAVEL ------------------ =F11.3/,
     *54H TRAVEL TIME (VEHICLE-SECONDS) ---------------------=F9.1/,
     *54H AVERAGE TRAVEL TIME (SECONDS) --------------------- =F9.1/,
     *54H NUMBER OF VEHICLES PROCESSED ---------------------- =I7/,
     *54H VOLUME PROCESSED (VEHICLES/HOUR) ------------------ =F9.1/,
     *54H TIME MEAN SPEED (MPH) = MEAN OF ALL VEHICLE SPEEDS  =F9.1/,
     *54H SPACE MEAN SPEED (MPH) = TOT DIST / TOT TRAVEL TIME =F9.1/,
     *54H AVERAGE DESIRED SPEED (MPH) ----------------------- =F9.1/,
     *54H AVERAGE MAXIMUM ACCELERATION (FT/SEC/SEC) --------- =F9.1/,
     *54H AVERAGE MAXIMUM DECELERATION (FT/SEC/SEC) --------- =F9.1)
  604 FORMAT(/,
     *54H OVERALL AVERAGE TOTAL DELAY (SECONDS) ------------- =F9.1/,
     *54H OVERALL AVERAGE QUEUE DELAY (SECONDS) ------------- =F9.1/,
     *54H OVERALL AVERAGE STOPPED DELAY (SECONDS) ----------- =F9.1/,
     *28H OVERALL AVERAGE DELAY BELOW,F5.1,21H MPH (SECONDS) ---- =F9.1)
  701 FORMAT(I2,I1,4(F7.1,I4),F5.3,F6.2,I4,3F4.1,2F3.1)
C
C-----SUBROUTINE PSTATS PRINTS SUMMARY STATISTICS FOR INBOUND APPROACH
C-----I AND TURN CODE J AND OPTIONALLY WRITE THE STATISTICS ONTO TAPE 7
C-----USING APPROACH NUMBER IWIA AND TURN CODE IWTC
C
      IF ( IPRINT.NE.NYES.AND.IPUNCH.NE.NYES )    RETURN
C-----FIND THE SINGLE DIMENSION INDEX FOR (I,J)
      INDEX  = (J-1)*6 + I
C-----INITIALIZE SOME PARAMETERS FOR PSTATS
      XMPH    = XFPS*60.0/88.0
      NUM     = NUMPRO(INDEX)
```

```
C-----IF NO VEHICLES WERE PROCESSED FOR INBOUND APPROACH I AND TURN
C-----CODE J THEN RETURN
                    IF ( NUM . LE . 0 )           RETURN
C-----FIND THE AVERAGE TRAVEL TIME
      ASTIM  = STIME(INDEX)/NUM
C-----PROCESS TOTAL DELAY
      PTD    = 100.0*NTD(INDEX)/NUM
      ATD    = 0.0
                    IF ( NTD(INDEX) . LE . 0 )   GO TO 1010
      ATD    = TD(INDEX)/NTD(INDEX)
 1010 CONTINUE
      ATDAST = 100.0*ATD/ASTIM
C-----PROCESS QUEUE DELAY
      PQD    = 100.0*NQD(INDEX)/NUM
      AQD    = 0.0
                    IF ( NQD(INDEX) . LE . 0 )   GO TO 1020
      AQD    = QD(INDEX)/NQD(INDEX)
 1020 CONTINUE
      AQDAST = 100.0*AQD/ASTIM
C-----PROCESS STOPPED DELAY
      PSD    = 100.0*NSD(INDEX)/NUM
      ASD    = 0.0
                    IF ( NSD(INDEX) . LE . 0 )   GO TO 1030
      ASD    = SD(INDEX)/NSD(INDEX)
 1030 CONTINUE
      ASDAST = 100.0*ASD/ASTIM
C-----PROCESS DELAY BELOW XX MPH
      PDMPH  = 100.0*NDMPH(INDEX)/NUM
      ADMPH  = 0.0
                    IF ( NDMPH(INDEX) . LE . 0 ) GO TO 1040
      ADMPH  = DMPH(INDEX)/NDMPH(INDEX)
 1040 CONTINUE
      ADMAST = 100.0*ADMPH/ASTIM
C-----FIND THE AVERAGE VEHICLE MILES OF TRAVEL
      AVMT   = VMT(INDEX)/NUM
C-----FIND THE EQUIVALENT HOURLY VOLUME PROCESSED
      VOLUME = NUM/(SIMTIM/3600.0)
C-----FIND THE TIME MEAN SPEED, THE SPACE MEAN SPEED, AND THE AVERAGE
C-----DESIRED SPEED
      TMSPD  = ASPEED(INDEX)/NUM
      SMSPD  = 3600.0*VMT(INDEX)/STIME(INDEX)
      ADSPD  = ADESPD(INDEX)/NUM
C-----FIND THE AVERAGE MAXIMUM ACCELERATION AND DECELERATION FOR THE
C-----VEHICLE
      AMAXV  = VMAXA(INDEX)/NUM
      DMAXV  = VMAXD(INDEX)/NUM
C-----FIND THE OVERALL AVERAGE DELAYS
      OATD   = TD(INDEX)/NUM
      OAQD   = QD(INDEX)/NUM
      OASD   = SD(INDEX)/NUM
      OADMPH = DMPH(INDEX)/NUM
                    IF ( IPRINT . NE . NYES )    GO TO 1050
C-----PRINT SUMMARY STATISTICS FOR INBOUND APPROACH I AND TURN CODE J
      PRINT 601 , TD(INDEX),NTD(INDEX),PTD,ATD,ATDAST,
     *            QD(INDEX),NQD(INDEX),PQD,AQD,AQDAST,
     *            SD(INDEX),NSD(INDEX),PSD,ASD,ASDAST
      PRINT 602 , XMPH,DMPH(INDEX),XMPH,NDMPH(INDEX),XMPH,PDMPH,
     *            XMPH,ADMPH,XMPH,ADMAST
      PRINT 603 , VMT(INDEX),AVMT,STIME(INDEX),ASTIM,NUM,VOLUME,TMSPD,
     *            SMSPD,ADSPD,AMAXV,DMAXV
      PRINT 604 , OATD,OAQD,OASD,XMPH,OADMPH
C-----OPTIONALLY WRITE THE STATISTICS ONTO TAPE 7 USING APPROACH IWIA
C-----AND TURN CODE IWTC
 1050 CONTINUE
                    IF ( IPUNCH . NE . NYES )    RETURN
      WRITE (7,701)  IWIA,IWTC,TD(INDEX),NTD(INDEX),QD(INDEX),
     *               NQD(INDEX),SD(INDEX),NSD(INDEX),DMPH(INDEX),
     *               NDMPH(INDEX),AVMT,ASTIM,NUM,TMSPD,SMSPD,ADSPD,
     *               AMAXV,DMAXV
      RETURN
      END                                                           PSTATS
```

```
      SUBROUTINE ADDSTA ( I,J,K )
      COMMON / SUMSTA / TD(18),NTD(18),QD(18),NQD(18),SD(18),MNVSY,
     *                  NSD(18),DMPH(18),NDMPH(18),VMT(18),
     *                  STIME(18),NUMPRO(18),ASPEED(18),ADESPD(18),
     *                  VMAXA(18),VMAXD(18),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / ZTEMPD / VSUMAR(15),VPSTAT(27),INDEX,VACTST(7),
     *                  VTIMST(16),ZTEMPD(44)
C
C-----SUBROUTINE ADDSTA ADDS THE SUMMARY STATISTICS FROM (J,K) TO (I,1)
C
C-----FIND THE SINGLE DIMENSION INDEX FOR (J,K)
      INDEX      = (K-1)*6    + J
C-----ADD THE SUMMARY STATISTICS FROM (INDEX) TO (I)
      TD(I)      = TD(I)      + TD(INDEX)
      NTD(I)     = NTD(I)     + NTD(INDEX)
      QD(I)      = QD(I)      + QD(INDEX)
      NQD(I)     = NQD(I)     + NQD(INDEX)
      SD(I)      = SD(I)      + SD(INDEX)
      NSD(I)     = NSD(I)     + NSD(INDEX)
      DMPH(I)    = DMPH(I)    + DMPH(INDEX)
      NDMPH(I)   = NDMPH(I)   + NDMPH(INDEX)
      VMT(I)     = VMT(I)     + VMT(INDEX)
      STIME(I)   = STIME(I)   + STIME(INDEX)
      NUMPRO(I)  = NUMPRO(I)  + NUMPRO(INDEX)
      ASPEED(I)  = ASPEED(I)  + ASPEED(INDEX)
      ADESPD(I)  = ADESPD(I)  + ADESPD(INDEX)
      VMAXA(I)   = VMAXA(I)   + VMAXA(INDEX)
      VMAXD(I)   = VMAXD(I)   + VMAXD(INDEX)
      RETURN
      END                                                           ADDSTA
```

```
      SUBROUTINE ACTSTA
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / PHASES / TII(8),TVI(8),TCI(8),TAR(8),TMX(8),ISKP(8),
     *                  IREC(8),NMAXO(8),TMAXO(8),NGAPO(8),TGAPO(8),
     *                  NLD(8),LLD(10,8),ICAMPS(8),IANDOR(8),IDUALL(8),
     *                  NPHNXT(8),LPHNXT(7,8),IMINOR(8),NPHASE,LPHASE(8)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VSUMAR(15),VPSTAT(27),VADDST(1),ATGAPO,ATMAXO,I,
     *                  IST,J,N,NN,VTIMST(16),ZTEMPD(44)
      DATA      NYES   / 3HYES /
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *        14HLATION PACKAGE,//,1X,20A4,//)
  602 FORMAT(44H SUMMARY STATISTICS FOR SEMI-ACTUATED SIGNAL//,
     *58H MAIN STREET PHASE NUMBER -------------------------- =   1/,
     *54H MAIN STREET MINIMUM ASSURED GREEN (SECONDS) ------- =F6.1/,
     *54H MAIN STREET AMBER CLEARANCE INTERVAL (SECONDS) ---- =F6.1/,
     *54H MAIN STREET ALL-RED CLEARANCE INTERVAL (SECONDS) -- =F6.1/,
     *54H MAIN STREET NUMBER OF PHASES CLEARED TO ------------=I4/,
     *54H MAIN STREET LIST OF PHASES CLEARED TO --------------=7I4)
  603 FORMAT(
     *54H NUMBER OF MAIN STREET GREEN PHASES----------------- =,I4/,
     *54H AVERAGE LENGTH OF MAIN STREET GREEN (SECONDS) ----- =F6.1)
  604 FORMAT(44H SUMMARY STATISTICS FOR FULL-ACTUATED SIGNAL)
  605 FORMAT(//,
     *54H SIGNAL PHASE NUMBER ------------------------------- =I4/,
     *54H INITIAL INTERVAL (SECONDS) ------------------------ =F6.1/,
     *54H VEHICLE INTERVAL (SECONDS) ------------------------ =F6.1/,
     *54H AMBER CLEARANCE INTERVAL (SECONDS) ---------------- =F6.1/,
     *54H ALL-RED CLEARANCE INTERVAL (SECONDS) -------------- =F6.1/,
     *54H MAXIMUM EXTENSION AFTER DEMAND ON RED (SECONDS) --- =F6.1/,
     *54H SKIP-PHASE SWITCH (ON/OFF) ------------------------ =3X,A3/,
     *54H AUTO-RECALL SWITCH (ON/OFF) ----------------------- =3X,A3/,
     *54H PARENT/MINOR MOVEMENT PHASE OPTION (YES/NO) ------- =3X,A3/,
     *54H DUAL LEFT OPTION (YES/NO) ------------------------- =3X,A3/,
     *54H DETECTOR CONNECTION TYPE (AND/OR) ----------------- =3X,A3/,
     *54H NUMBER OF DETECTORS CONNECTED TO PHASE ------------ =I4/,
     *54H NUMBER OF PHASES CLEARED TO ----------------------- =I4/,
     *54H LIST OF PHASES CLEARED TO ------------------------- =7I4)
  606 FORMAT(54H LIST OF DETECTORS CONNECTED TO PHASE -------------- =,
     *        5I4,/,54X,5I4)
  607 FORMAT(
     *54H NUMBER OF MAX-OUTS -------------------------------- =,I4/,
     *54H AVERAGE TIME INTO PHASE FOR MAX-OUT (SECONDS) ----- =F6.1/,
     *54H NUMBER OF GAP-OUTS -------------------------------- =I4/,
     *54H AVERAGE TIME INTO PHASE FOR GAP-OUT (SECONDS) ----- =F6.1)
  608 FORMAT(1H1)
  701 FORMAT(I2,5F5.1,5A3,2(F5.1,I3))
C
C-----SUBROUTINE ACTSTA PRINTS THE ACTUATED SIGNAL CONTROLLER STATISTICS
C-----AND OPTIONALLY WRITES THE ACTUATED SIGNAL CONTROLLER STATISTICS
C-----ONTO TAPE 7
C
      PRINT 601 , ITITLE
C-----IF THE INTERSECTION IS FULL-ACTUATED SIGNAL CONTROLLED THEN GO TO
C-----1020 AND PRINT THE FULL-ACTUATED SIGNAL CONTROLLER STATISTICS
C-----ELSE PRINT THE MAIN STREET SEMI-ACTUATED SIGNAL CONTROLLER
C-----STATISTICS
                    IF ( ICONTR . NE . 6 )        GO TO 1020
C-----SET THE STARTING INDEX FOR /PHASES/ PRINT TO 2 (THE FIRST IS THE
C-----MAIN STREET SEMI-ACTUATED SIGNAL CONTROLLER PHASE)
      IST = 2
      ATMAXO = 0.0
                    IF ( NMAXO(1) . LE . 0 )      GO TO 1010
C-----FIND THE AVERAGE TIME INTO THE SIGNAL PHASE TO MAX-OUT
      ATMAXO = TMAXO(1)/NMAXO(1)
 1010 CONTINUE
      NN = NPHNXT(1)
      PRINT 602 , TII(1),TCI(1),TAR(1),NN,(LPHNXT(J,1),J=1,NN)
      PRINT 603 , NMAXO(1),ATMAXO
C-----OPTIONALLY WRITE THE ACTUATED SIGNAL CONTROLLER STATISTICS ONTO
C-----TAPE 7
                    IF ( IPUNCH . NE . NYES )     GO TO 1030
      I = 1
      ATGAPO = 0.0
      WRITE (7,701)  I,TII(1),TVI(1),TCI(1),TAR(1),TMX(1),ISKP(1),
     *               IREC(1),IMINOR(1),IDUALL(1),IANDOR(1),ATMAXO,
     *               NMAXO(1),ATGAPO,NGAPO(1)
      GO TO 1030
 1020 CONTINUE
C-----SET THE STARTING INDEX FOR /PHASES/ PRINT TO 1 (START AT FIRST)
      IST = 1
      PRINT 604
 1030 CONTINUE
C-----PROCESS EACH ACTUATED SIGNAL CONTROLLER PHASE STARTING AT IST
      DO 2040  I = IST , NPHASE
      ATMAXO = 0.0
                    IF ( NMAXO(I) . EQ . 0 )      GO TO 2010
C-----FIND THE AVERAGE TIME INTO THE SIGNAL PHASE TO MAX-OUT
      ATMAXO = TMAXO(I)/NMAXO(I)
 2010 CONTINUE
      ATGAPO = 0.0
                    IF ( NGAPO(I) . EQ . 0 )      GO TO 2020
C-----FIND THE AVERAGE TIME INTO THE SIGNAL PHASE TO GAP-OUT
      ATGAPO = TGAPO(I)/NGAPO(I)
 2020 CONTINUE
      N = NLD(I)
      NN = NPHNXT(I)
      PRINT 605 , I,TII(I),TVI(I),TCI(I),TAR(I),TMX(I),ISKP(I),IREC(I),
     *            IMINOR(I),IDUALL(I),IANDOR(I),N,NN,
     *            (LPHNXT(J,I),J=1,NN)
                    IF ( N . LE . 0 )             GO TO 2030
      PRINT 606 , (LLD(J,I),J=1,N)
 2030 CONTINUE
      PRINT 607 , NMAXO(I),ATMAXO,NGAPO(I),ATGAPO
                IF ( (I/2)*2.EQ.I.AND.I.NE.NPHASE )  PRINT 608
C-----OPTIONALLY WRITE THE ACTUATED SIGNAL CONTROLLER STATISTICS ONTO
C-----TAPE 7
                    IF ( IPUNCH . NE . NYES )     GO TO 2040
      WRITE (7,701)  I,TII(I),TVI(I),TCI(I),TAR(I),TMX(I),ISKP(I),
     *               IREC(I),IMINOR(I),IDUALL(I),IANDOR(I),ATMAXO,
     *               NMAXO(I),ATGAPO,NGAPO(I)
 2040 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TIMSTA
      COMMON / INTER  / NVATIN,LVATIN(25),TVATIN(25),NIBA,LIBA(6),NOBA,
     *                  LOBA(6),NVSY,NVIA(12),NVIBA,NVOBA,NVIN,NPATHS,
     *                  NVIP(125),NOCONF,ICONTR,NUMSDR,NIBL,NRLAN,
     *                  LIBAR(12),LOBAR(12)
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / TITLE  / ITITLE(20)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / VSUMAR(15),VPSTAT(27),VADDST(1),VACTST(7),ANVSY,
     *                  COSTIN,COSTSI,COSTSS,COSTSU,COSTTO,IOUT,TMIN,
     *                  TMRAT,TMRDT,TMRSI,TMRSU,TMSI,TMSS,TMSU,TMTO,
     *                  ZTEMPD(44)
  601 FORMAT(1H1,10X,47HSIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMU,
     *       14HLATION PACKAGE,//,1X,20A4,//)
  602 FORMAT(23H        START-UP TIME    =F9.3,8H SECONDS,
     *       36H        NUMBER OF VEHICLES PROCESSED =I5,//,
     *       23H        SIMULATION TIME =F9.3,8H SECONDS,
     *       36H        NUMBER OF VEHICLES PROCESSED =I5,//,
     *       51H        NUMBER OF VEHICLES IN THE SYSTEM AT SUMMARY =I5,/,
     *       51H        AVERAGE NUMBER OF VEHICLES IN THE SYSTEM -- =F7.1,
     *7H  MAX =,I4)
  603 FORMAT(////,
     *26H      INITIAL     TM TIME =F9.3,8H SECONDS,3X,8HCOST = $F6.2//,
     *26H      START-UP    TM TIME =F9.3,8H SECONDS,3X,8HCOST = $F6.2/,
     *26H                  REAL/TM =F9.3,8H         ,//,
     *26H      SIMULATION TM TIME =F9.3,8H SECONDS,3X,8HCOST = $F6.2/,
     *26H                  REAL/TM =F9.3,8H         ,//,
     *26H      SUMMARY     TM TIME =F9.3,8H SECONDS,3X,8HCOST = $F6.2///,
     *26H      TOTAL       TM TIME =F9.3,8H SECONDS,3X,8HCOST = $F6.2///)
  604 FORMAT(49H        VEHICLE-SECONDS OF SIMULATION PER TM TIME =F9.3//,
     *       35H        VEHICLE UPDATES PER TM TIME =F9.3)
C
C-----SUBROUTINE TIMSTA PRINTS THE COMPUTER TIME STATISTICS
C
C-----FIND THE AVERAGE NUMBER OF VEHICLES IN THE SYSTEM DURING
C-----SIMULATION TIME
      ANVSY = NVSYA*DT/SIMTIM
C-----FIND THE TM TIME FOR INITIAL
      TMIN = TMTIME(2) - TMTIME(1)
C-----FIND THE TM TIME FOR START-UP
      TMSU = AMAX1(TMTIME(3)-TMTIME(2),0.00000001)
      TMRSU = STRTIM / TMSU
C-----FIND THE TM TIME FOR SIMULATION
      TMSI = TMTIME(4) - TMTIME(3)
      TMRSI = SIMTIM / TMSI
C-----GET THE TM TIME FOR THIS JOB AT THE END OF SUMMARY
      CALL  EXTIME  ( 5 )
C-----FIND THE TM TIME FOR SUMMARY
      TMSS = TMTIME(5) - TMTIME(4)
C-----FIND THE TM TIME FOR THE TOTAL JOB
      TMTO = TMTIME(5) - TMTIME(1)
C-----FIND THE SIMULATION REAL TIME TO COMPUTER TM TIME RATIO
      TMRAT = STIME(1,1) / TMSI
      TMRDT = TMRAT/DT
C-----FIND THE COSTS (ONE CDC TM HOUR = 230 DOLLARS)
C¬    COSTIN = TMIN*230.00/3600.0
C¬    COSTSU = TMSU*230.00/3600.0
C¬    COSTSI = TMSI*230.00/3600.0
C¬    COSTSS = TMSS*230.00/3600.0
C¬    COSTTO = TMTO*230.00/3600.0
C-----FIND THE COSTS (ONE IBM CPU MINUTE = 10 DOLLARS - REDUCED RATE)
C!    COSTIN = TMIN*600.00/3600.0
C!    COSTSU = TMSU*600.00/3600.0
C!    COSTSI = TMSI*600.00/3600.0
C!    COSTSS = TMSS*600.00/3600.0
C!    COSTTO = TMTO*600.00/3600.0
      PRINT 601 , ITITLE
      PRINT 602 , STRTIM,NUMPSU,SIMTIM,NUMPRO(1,1),NVSY,ANVSY,MNVSY
      PRINT 603 , TMIN,COSTIN,TMSU,COSTSU,TMRSU,TMSI,COSTSI,TMRSI,
     *            TMSS,COSTSS,TMTO,COSTTO
      PRINT 604 , TMRAT,TMRDT
C¬    IOUT = 6LOUTPUT
C¬    ENDFILE IOUT
      RETURN
      END                                                         TIMSTA
```

```
      SUBROUTINE EXTIME ( I )
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      DATA     N1,N2  / 4HEXTI,2HME /
C
C-----SUBROUTINE EXTIME GETS THE TM TIME FOR THIS JOB
C
      NRNAME = NRNAME + 1
      IRNAME(1,NRNAME) = N1
      IRNAME(2,NRNAME) = N2
                      IF ( NRNAME . GT . NRNAMM )  CALL ABORTR ( MSGR,NR )
C-----GET THE TM TIME FOR THIS JOB (CDC)
C=    CALL  JOBINFO ( 0,ITM )
C=    TMTIME(I) = ITM/1000.0
C-----GET CPU TIME FOR THIS JOB (IBM)
C;                    IF ( I . EQ . 1 )              CALL  CLOCK1
C;    TMTIME(I) = CLOCK ( 0.0 )*60.0*26.04166/26.0
      RETURN
      END                                                               EXTIME

      SUBROUTINE ABORTR (MSG,    NCHS)                                  COLEASE
C   TASK,ABORTR,MSG,NCHS
      COMMON / APPRO  / NLANES    ,LLANES( 6),NVIL  ( 6),ISLIM    ,     COLEASE
     *                  IALEFT    ,NSDR      ,ISDRN ( 5),ISDRA ( 5)     COLEASE
      COMMON / CONFLT / ICONP ( 2),ICONA ( 2),ICOND ( 2),ICONAN    ,    COLEASE
     *                  ICONI ( 2),ICONV ( 2),IDUMCO                    COLEASE
      COMMON / LANE   / LWID      ,NLL       ,NLR       ,ISNA      ,    COLEASE
     *                  NPINT     ,LINTP ( 7),IFVL      ,ILVL      ,    COLEASE
     *                  LCONTR    ,LTURN     ,LGEOM ( 4),NLDL      ,    COLEASE
     *                  LLDL  ( 5),IBLN      ,IDUMLA                    COLEASE
      COMMON / NOATTB / NOATTB( 8)                                      COLEASE
      COMMON / PATH   / LENP      ,IOPT      ,LIBL      ,LOBL      ,    COLEASE
     *                  IFVP      ,ILVP      ,LIMP      ,IPT       ,    COLEASE
     *                  NGEOCP    ,NCPSET    ,ICPSET(60),LOBAP     ,    COLEASE
     *                  ILCH      ,IGEOCP(60)                           COLEASE
      COMMON / SDR    / ICANSE(40)                                      COLEASE
      COMMON / VEHD   / ISLP      ,IACC      ,IVEL      ,IPOS      ,    COLEASE
     *                  ISET      ,LCHGE     ,ISPDP     ,LEGAL     ,    COLEASE
     *                  IPRTM     ,ITIMV     ,IQDS      ,ISPDS     ,    COLEASE
     *                  ISDS      ,IDVS      ,ISTCON    ,IVMAXA    ,    COLEASE
     *                  IVMAXD    ,LATPOS    ,IDTS      ,LALT      ,    COLEASE
     *                  NORC      ,LOGFLG    ,MSTPF     ,MLAG      ,    COLEASE
     *                  MTCARS    ,MFINL     ,MSFLG     ,MPOBS     ,    COLEASE
     *                  MOASF     ,MSAOR     ,MPRO      ,MBLOCK    ,    COLEASE
     *                  MININT    ,IFVA      ,IACDS     ,ICDFS     ,    COLEASE
     *                  ISDEC     ,ISTMO     ,IACLDS    ,IRSTOP         COLEASE
      COMMON / VEHF   / IDRICL    ,IVEHCL    ,ISPD      ,NOF       ,    COLEASE
     *                  NOR       ,LNEXT     ,LPRES     ,ITURN     ,    COLEASE
     *                  IBAPS     ,IPRTLO    ,IEXTIM    ,NOBAPD         COLEASE
      COMMON / VEHIL  / MDEDIC    ,MINFLZ    ,MLUNC     ,MIUNC     ,    COLEASE
     *                  MLYELD    ,MLSTOP    ,MATSTL    ,MSSRED    ,    COLEASE
     *                  MLRTOR    ,MSSGRN    ,MCHKCF    ,MDUMIL    ,    COLEASE
     *                  IDEDIC    ,INFLZ     ,ILUNC     ,ILYELD    ,    COLEASE
     *                  ILSTOP    ,ICONTN    ,ICHKCF    ,IERROR         COLEASE
      COMMON / ATTB   / IAT (3,  310)
      COMMON / ENTITY / IEN (9,    8)
      COMMON / FUN    / IFU (2,   31)
      COMMON / ABIAS  / SLPOLD,ACCOLD,VELOLD,POSOLD,
     *                  SLPNEW,ACCNEW,VELNEW,POSNEW,RELVEL,RELPOS,
     *                  PVACC,PVVEL,PVPOS,ENDLN,RELEND,OLDDTS,DESVEL
      COMMON / CLASS  / LENV(15),VCHAR(15),DCHAR(5),IPIJR(5),PIJR(5),
     *                  DMAX(15),AMAX(15),VMAX(15),IRMIN(15),DCHARM
      COMMON / INDEX  / IV,IVN,IL,ILN,IA,IAN,IP,LOGTMP,JPRTM,ICONUP,
     *                  IPTHUP,IREPIL,IREPFX,IVPV,IPFLAG,JPFLAG,KPFLAG
      COMMON / LANECH / PVSF,VVSF,AVSF,PVSR,VVSR,AVSR,SLPLCH,FACTOR,
     *                  ISIDE,LEADSP,LAGSPD,NOSF,NOSR
      COMMON / QUE    / IBUF(25,8),QTIME(25),LQ(6,6),IQ(200),IEF,IQF,
     *                  NUMV
      COMMON / RUTINE / NRNAME,IRNAME(2,36),MSGR(4),NRNAMM,NR
      COMMON / SIGCAM / TCAMSP(72),ICAMPH(72),NCAMSP,ICAMPC,ICAMPO,
     *                  ISISET(72,25),ICPHAS,TP,TR,IGO,IARRPH
      COMMON / SUMSTA / TD(6,3),NTD(6,3),QD(6,3),NQD(6,3),SD(6,3),MNVSY,
     *                  NSD(6,3),DMPH(6,3),NDMPH(6,3),VMT(6,3),
     *                  STIME(6,3),NUMPRO(6,3),ASPEED(6,3),ADESPD(6,3),
     *                  VMAXA(6,3),VMAXD(6,3),NUMPSU,XFPS,XQDIST,
     *                  LQUEUE(6,6),MQUEUE(6,6),NVSYA,NBANG(6),NELIM(6),
     *                  PLVDV(6),NLVDV(6),TMTIME(5)
      COMMON / USER   / STRTIM,SIMTIM,TIME,DT,DTSQ,DTCU,TPRINT,TSTATS,
     *                  CAREQL,CAREQM,CAREQA,TLEAD,TLAG,DUTOL,AUTOL,
     *                  APIJR,INPUT,IGEOP,IVEHP,IPTC,IPAP,IPUNCH,IPOLL
      COMMON / ZTEMPD / ZTEMPD(95),I,ICHS,IRN,ITIME,MSGPP(9),NUM,NWDS
      DIMENSION         COM01(1),COM02(1),COM03(1),COM04(1),COM05(1),
     *                  COM06(1),COM07(1),COM08(1),COM09(1),COM10(1),
     *                  COM11(1),COM12(1),COM13(1),COM14(1),COM15(1),
     *                  COM16(1)
      DIMENSION         IC(2,19),MSG(1)
      DIMENSION         NCOM01(2,026),NCOM02(2,012),NCOM03(2,028),
     *                  NCOM04(2,132),               NCOM06(2,040),
     *                  NCOM07(2,012),NCOM08(2,020),NCOM09(2,017),
     *                                NCOM11(2,017),NCOM12(2,013),
     *                                NCOM14(2,006),
```

```
     *                    NCOM16(2,023)
C     DIMENSION           NCOM05(2,040),NCOM10(2,005),NCOM13(2,300),
C    *                    NCOM15(2,005)
      EQUIVALENCE         (NLANES   ,COM01(1)),(ICONP (1),COM02(1)),
     *                    (LWID     ,COM03(1)),(LENP     ,COM04(1)),
     *                    (ICANSE(1),COM05(1)),(ISLP     ,COM06(1)),
     *                    (IDRICL   ,COM07(1)),(MDEDIC   ,CUM08(1)),
     *                    (SLPOLD   ,COM09(1)),(IPIJR (1),COM10(1)),
     *                    (IV       ,COM11(1)),(PVSF     ,COM12(1)),
     *                    (QTIME (1),COM13(1)),(NCAMSP   ,COM14(1)),
     *                    (NUMPSU   ,COM15(1)),(STRTIM   ,COM16(1))
      DATA      IC      / 4HAPPR,2HO ,4HCONF,2HLT,4HLANE,2H  ,4HPATH,2H  ,
     *                    4HSDR ,2H  ,4HVEHD,2H  ,4HVEHF,2H  ,4HVEHI,2HL ,
     *                    4HABIA,2HS ,4HCLAS,2HS ,4HINDE,2HX ,4HLANE,2HCH,
     *                    4HQUE ,2H  ,4HSIGC,2HAM,4HSUMS,2HTA,4HUSER,2H  ,
     *                    4HATTB,2H  ,4HENTI,2HTY,4HFUN ,2H  /
      DATA      NCOM01  / 4HNLAN,2HES,4HLLAN,2HES,10*1H        ,4HNVIL,2H  ,
     *                    10*1H      ,4HISLI,2HM ,4HIALE,2HFT,4HNSDR,2H  ,
     *                    4HIDSR,2HN ,8*1H       ,4HIDSR,2HA ,8*1H       /
      DATA      NCOM02  / 4HICON,2HP ,2*1H       ,4HICON,2HA ,2*1H       ,
     *                    4HICON,2HD ,2*1H       ,4HICON,2HAN,4HICON,2HI ,
     *                    2*1H       ,4HICON,2HV ,2*1H       ,4HIDUM,2HCO/
      DATA      NCOM03  / 4HLWID,2H  ,4HNLL ,2H  ,4HNLR ,2H  ,4HISNA,2H  ,
     *                    4HNPIN,2HT ,4HLINT,2HP ,12*1H       ,4HIFVL,2H  ,
     *                    4HILVL,2H  ,4HLCON,2HTR,4HLTUR,2HN ,4HLGEO,2HM ,
     *                    6*1H       ,4HNLDL,2H  ,4HLLDL,2H  ,8*1H       ,
     *                    4HIBLN,2H  ,4HIDUM,2HLA/
      DATA      NCOM04  / 4HLENP,2H  ,4HIOPT,2H  ,4HLIBL,2H  ,4HLOBL,2H  ,
     *                    4HIFVP,2H  ,4HILVP,2H  ,4HLIMP,2H  ,4HIPT ,2H  ,
     *                    4HNGEO,2HCP,4HNCPS,2HET,4HICPS,2HET,118*1H     ,
     *                    4HLOBA,2HP ,4HILCH,2H  ,4HIGEO,2HCP,118*1H     /
C     DATA      NCOM05  / 4HICAN,2HSE,78*1H      /
      DATA      NCOM06  / 4HISLP,2H  ,4HIACC,2H  ,4HIVEL,2H  ,4HIPOS,2H  ,
     *                    4HISET,2H  ,4HLCHG,2HE ,4HISPD,2HP ,4HLEGA,2HL ,
     *                    4HIPRT,2HM ,4HITIM,2HV ,4HIQDS,2H  ,4HISPD,2HS ,
     *                    4HISDS,2H  ,4HIDVS,2H  ,4HISTC,2HON,4HIVMA,2HXA,
     *                    4HIVMA,2HXD,4HLATP,2HOS,4HIDTS,2H  ,4HLALT,2H  ,
     *                    4HNORC,2H  ,4HLOGF,2HLG,4HMSTP,2HF ,4HMLAG,2H  ,
     *                    4HMTCA,2HRS,4HMFIN,2HL ,4HMSFL,2HG ,4HMPOS,2HS ,
     *                    4HMOAS,2HF ,4HMSAD,2HR ,4HMPRO,2H  ,4HMBLO,2HCK,
     *                    4HMINI,2HNT,4HIFVA,2H  ,4HIACD,2HS ,4HICDF,2HS ,
     *                    4HISDE,2HC ,4HISTM,2HO ,4HIACL,2HDS,4HIRST,2HOP/
      DATA      NCOM07  / 4HIDRI,2HCL,4HIVEH,2HCL,4HISPD,2H  ,4HNOF ,2H  ,
     *                    4HNOR ,2H  ,4HLNEX,2HT ,4HLPRE,2HS ,4HITUR,2HN ,
     *                    4HIBAP,2HS ,4HIPRT,2HLO,4HIEXT,2HIM,4HNOBA,2HPD/
      DATA      NCOM08  / 4HMDED,2HIC,4HMINF,2HLZ,4HMLUN,2HC ,4HMIUN,2HC ,
     *                    4HMLYE,2HLD,4HMLST,2HOP,4HMATS,2HTL,4HMSSR,2HED,
     *                    4HMLRT,2HOR,4HMSSG,2HRN,4HMCHK,2HCF,4HIDUM,2HIL,
     *                    4HIDED,2HIC,4HINFL,2HZ ,4HILUN,2HC ,4HILYE,2HLD,
     *                    4HILST,2HOP,4HICON,2HTN,4HICHK,2HCF,4HIERR,2HOR/
      DATA      NCOM09  / 4HSLPO,2HLD,4HACCO,2HLD,4HVELO,2HLD,4HPOSO,2HLD,
     *                    4HSLPN,2HEW,4HACCN,2HEW,4HVELN,2HEW,4HPOSN,2HEW,
     *                    4HRELV,2HEL,4HRELP,2HOS,4HPVAC,2HC ,4HPVVE,2HL ,
     *                    4HPVPO,2HS ,4HENDL,2HN ,4HRELE,2HND,4HOLDD,2HTS,
     *                    4HDESV,2HEL/
C     DATA      NCOM10  / 4HIPIJ,2HR ,8*1H       /
      DATA      NCOM11  / 4HIV  ,2H  ,4HIVN ,2H  ,4HIL  ,2H  ,4HILN ,2H  ,
     *                    4HIA  ,2H  ,4HIAN ,2H  ,4HIP  ,2H  ,4HLOGT,2HMP,
     *                    4HJPRT,2HM ,4HICON,2HUP,4HIPTH,2HUP,4HIREP,2HIL,
     *                    4HIREP,2HFX,4HIVPV,2H  ,4HIPFL,2HAG,4HJPFL,2HAG,
     *                    4HKPFL,2HAG/
      DATA      NCOM12  / 4HPVSF,2H  ,4HVVSF,2H  ,4HAVSF,2H  ,4HPVSR,2H  ,
     *                    4HVVSR,2H  ,4HAVSR,2H  ,4HSLPL,2HCH,4HFACT,2HOR,
     *                    4HISID,2HE ,4HLEAD,2HSP,4HLAGS,2HPD,4HNOSF,2H  ,
     *                    4HNOSR,2H  /
C     DATA      NCOM13  / 4HQTIM,2HE ,48*1H      ,4HLQ  ,2H  ,142*1H     ,
C    *                    4HIQ  ,2H  ,398*1H     ,4HIEF ,2H  ,4HIQF ,2H  ,
C    *                    4HNUMV,2H  /
      DATA      NCOM14  / 4HNCAM,2HSP,4HICAM,2HPC,4HICAM,2HPO,4HICPH,2HAS,
     *                    4HTP  ,2H  ,4HTR  ,2H  /
C     DATA      NCOM15  / 4HNUMP,2HSU,4HXFPS,2H  ,4HXQDI,2HST,4HNVSY,2HA ,
C    *                    4HNBAN,2HG /
      DATA      NCOM16  / 4HSTRT,2HIM,4HSIMT,2HIM,4HTIME,2H  ,4HDT  ,2H  ,
     *                    4HDTSQ,2H  ,4HDTCU,2H  ,4HTPRI,2HNT,4HTSTA,2HTS,
     *                    4HCARE,2HQL,4HCARE,2HQM,4HCARE,2HQA,4HTLEA,2HD ,
     *                    4HTLAG,2H  ,4HDUTO,2HL ,4HAUTO,2HL ,4HAPIJ,2HR ,
     *                    4HINPU,2HT ,4HIGEO,2HP ,4HIVEH,2HP ,4HIPTC,2H  ,
     *                    4HIPAP,2H  ,4HIPUN,2HCH,4HIPOL,2HL /
  601 FORMAT(20A4)
  602 FORMAT(15H0 COMMON BLOCK ,A4,A2,/)
C¬603 FORMAT(2X,A4,A2,3H =  ,O20,5HB =  I8)
C;603 FORMAT(2X,A4,A2,3H =  ,Z8 ,5HZ =  I8)
C¬604 FORMAT(2X,A4,A2,3H =  ,O20,5HB =  F17.8)
C;604 FORMAT(2X,A4,A2,3H =  ,Z8 ,5HZ =  F17.8)
C¬605 FORMAT(2X,A4,A2,3H =  ,O20,5HB = =,A10,1H=)
C;605 FORMAT(2X,A4,A2,3H =  ,Z8 ,5HZ = =,A4,1H=)
  606 FORMAT(1H1)
C 607 FORMAT(12H  ATTRIBUTE ,I3,7H WORD =,I3,8H SHIFT =,I3,8H MASK = ,
C    *         O20,1HB)
C 607 FORMAT(12H  ATTRIBUTE ,I3,7H WORD =,I3,8H SHIFT =,I3,8H MASK = ,
C    *         Z8,1HZ)
C 608 FORMAT(9H  ENTITY ,I3,7H DATA =,9I5)
C 609 FORMAT(16H  FUNCTION MASK ,O20,20HB  ATTRIBUTE NUMBER ,I4)
C 609 FORMAT(16H  FUNCTION MASK ,Z8 ,20HZ  ATTRIBUTE NUMBER ,I4)
  610 FORMAT(18(1X,A4,A2))
C¬701 FORMAT(2HT=,F6.1)
C
C-----SUBROUTINE ABORTR PROCESSES SYSTEM AND USER ERRORS
C
C¬    ASSIGN 101 TO IRECAD
C¬    ASSIGN 102 TO JRECAD
C¬    CALL  XMIT    ( IRECAD )
C-----PRINT THE ERROR MESSAGE
      NWDS = (NCHS+3)/4
      PRINT 601
      PRINT 601 , (MSG(I),I=1,NWDS)
      PRINT 601
C-----PRINT THE NAMES OF ALL ROUTINES CALLED
      PRINT 610 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
C-----PRINT THE CURRENT VALUE OF THE ATTRIBUTES IN EACH ENTITY
      PRINT 606
      NUM = NOATTB(1)
      PRINT 602 , IC(1,01),IC(2,01)
      PRINT 603 , (NCOM01(1,I),NCOM01(2,I),COM01(I),COM01(I),I=1,NUM)
      NUM = NOATTB(2)
      PRINT 602 , IC(1,02),IC(2,02)
      PRINT 603 , (NCOM02(1,I),NCOM02(2,I),COM02(I),COM02(I),I=1,NUM)
      PRINT 606
      NUM = NOATTB(3)
      PRINT 602 , IC(1,03),IC(2,03)
      PRINT 603 , (NCOM03(1,I),NCOM03(2,I),COM03(I),COM03(I),I=1,NUM)
      NUM = 10 + NGEOCP
      PRINT 602 , IC(1,04),IC(2,04)
      PRINT 603 , (NCOM04(1,I),NCOM04(2,I),COM04(I),COM04(I),I=1,NUM)
      NUM = 72 + NGEOCP
      PRINT 603 , (NCOM04(1,I),NCOM04(2,I),COM04(I),COM04(I),I=71,NUM)
C     NUM = NOATTB(5)
C     PRINT 602 , IC(1,05),IC(2,05)
C     PRINT 603 , (NCOM05(1,I),NCOM05(2,I),COM05(I),COM05(I),I=1,NUM)
      PRINT 606
      NUM = NOATTB(6)
      PRINT 602 , IC(1,06),IC(2,06)
      PRINT 603 , (NCOM06(1,I),NCOM06(2,I),COM06(I),COM06(I),I=1,22)
      PRINT 601
      PRINT 603 , (NCOM06(1,I),NCOM06(2,I),COM06(I),COM06(I),I=23,33)
      PRINT 601
      PRINT 603 , (NCOM06(1,I),NCOM06(2,I),COM06(I),COM06(I),I=34,NUM)
      PRINT 606
      NUM = NOATTB(7)
      PRINT 602 , IC(1,07),IC(2,07)
      PRINT 603 , (NCOM07(1,I),NCOM07(2,I),COM07(I),COM07(I),I=1,NUM)
      NUM = NOATTB(8)
```

```
      PRINT 602 , IC(1,08),IC(2,08)
      PRINT 603 , (NCOM08(1,I),NCOM08(2,I),COM08(I),COM08(I),I=1,12)
      PRINT 601
      PRINT 603 , (NCOM08(1,I),NCOM08(2,I),COM08(I),COM08(I),I=13,NUM)
C-----PRINT THE CURRENT VALUE OF THE VARIABLES IN SELECTED COMMON BLOCKS
      PRINT 606
      PRINT 602 , IC(1,09),IC(2,09)
      PRINT 604 , (NCOM09(1,I),NCOM09(2,I),COM09(I),COM09(I),I=1,17)
C     PRINT 602 , IC(1,10),IC(2,10)
C     PRINT 603 , (NCOM10(1,I),NCOM10(2,I),COM10(I),COM10(I),I=1,5)
      PRINT 602 , IC(1,11),IC(2,11)
      PRINT 603 , (NCOM11(1,I),NCOM11(2,I),COM11(I),COM11(I),I=1,14)
      PRINT 605 , (NCOM11(1,I),NCOM11(2,I),COM11(I),COM11(I),I=15,17)
      PRINT 602 , IC(1,12),IC(2,12)
      PRINT 604 , (NCOM12(1,I),NCOM12(2,I),COM12(I),COM12(I),I=1,8)
      PRINT 603 , (NCOM12(1,I),NCOM12(2,I),COM12(I),COM12(I),I=9,13)
C     PRINT 602 , IC(1,13),IC(2,13)
C     PRINT 604 , (NCOM13(1,I),NCOM13(2,I),COM13(I),COM13(I),I=1,25)
C     PRINT 603 , (NCOM13(1,I),NCOM13(2,I),COM13(I),COM13(I),I=26,300)
      PRINT 606
      PRINT 602 , IC(1,14),IC(2,14)
      PRINT 603 , (NCOM14(1,I),NCOM14(2,I),COM14(I),COM14(I),I=1,3)
      PRINT 603 , NCOM14(1,4),NCOM14(2,4),ICPHAS,ICPHAS
      PRINT 604 , NCOM14(1,5),NCOM14(2,5),TP,TP
      PRINT 604 , NCOM14(1,6),NCOM14(2,6),TR,TR
C     PRINT 602 , IC(1,15),IC(2,15)
C     PRINT 603 , NCOM15(1,1),NCOM15(2,1),COM15(1),COM15(1)
C     PRINT 604 , (NCOM15(1,I),NCOM15(2,I),COM15(I),COM15(I),I=2,3)
C     PRINT 603 , NCOM15(1,4),NCOM15(2,4),NVSYA,NVSYA
C     PRINT 603 , NCOM15(1,5),NCOM15(2,5),NBANG,NBANG
      PRINT 602 , IC(1,16),IC(2,16)
      PRINT 604 , (NCOM16(1,I),NCOM16(2,I),COM16(I),COM16(I),I=1,16)
      PRINT 603 , (NCOM16(1,I),NCOM16(2,I),COM16(I),COM16(I),I=17,19)
      PRINT 605 , (NCOM16(1,I),NCOM16(2,I),COM16(I),COM16(I),I=20,23)
C-----PRINT THE CULEASE STORAGE MANAGEMENT COMMON BLOCKS
C     PRINT 602 , IC(1,17),IC(2,17)
C     PRINT 607 , (I,IAT(1,I),IAT(2,I),IAT(3,I),I=1,310)
C     PRINT 602 , IC(1,18),IC(2,18)
C     PRINT 608 , (I,(IEN(J,I),J=1,9),I=1,8)
C     PRINT 602 , IC(1,19),IC(2,19)
C     PRINT 609 , IFU
C-----PRINT THE NAMES OF ALL ROUTINES CALLED
      PRINT 601
      PRINT 610 , (IRNAME(1,IRN),IRNAME(2,IRN),IRN=1,NRNAME)
C-----PRINT THE TIME INTO THE SIMULATION IN THE USERS DAYFILE
C¬101 CONTINUE
C¬    CALL  XMIT    ( JRECAD )
C¬          ENCODE  ( 10,701,ITIME )              TIME
C¬    ITIME = ITIME . AND . 8L;;;;;;;;
C¬    CALL  REMARK  ( ITIME )
C-----PRINT THE SUMMARY STATISTICS
                IF ( TIME . LE . STRTIM )    GO TO 102
      CALL  SUMARY
  102 CONTINUE
C-----PRINT THE ERROR MESSAGE IN THE USERS DAYFILE
C¬    ICHS = NWDS*4
C¬          ENCODE  ( ICHS,601,MSGPP )            (MSG(I),I=1,NWDS)
C¬    I = (NCHS+9)/10 + 1
C¬    MSGPP(I) = 0
C¬    CALL  XMIT    ( 0 )
C¬    CALL  REMARK  ( MSGPP )
      RETURN
C¬103 GO TO IRECAD                                                       *DEBUG*
C¬104 GO TO JRECAD                                                       *DEBUG*
      END                                                                ABORTR
```

```
      SUBROUTINE SMEP ( IR,IY,IN,IV,IE,ISNAME )                          COLEASE
      DIMENSION        ISNAME(2),IERROR(8)                               COLEASE
      DATA      IERROR / 4H FAT,4HAL E,4HRROR,4H IN ,4HCOLE,4HASE /      COLEASE
  901 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,13H OUT OF RANGE)  COLEASE
  902 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,11H ATTRIBUTE ,I3, COLEASE
     *        13H OUT OF RANGE)                                          COLEASE
  903 FORMAT(5H0----,A4,A2,8H ENTITY ,I2,7H ENTRY ,I3,11H ATTRIBUTE ,I3, COLEASE
C¬   *        3H = ,020,4HR = ,I10,9H OVERFLOW)                          COLEASE
C;   *        3H = ,Z8,4HZ = ,I10,9H OVERFLOW)                           COLEASE
      GO TO ( 9010,9020,9030 ) , IE                                      COLEASE
 9010 CONTINUE                                                           COLEASE
      PRINT 901 , ISNAME,IY,IN                                           COLEASE
      GO TO 9040                                                         COLEASE
 9020 CONTINUE                                                           COLEASE
      PRINT 902 , ISNAME,IY,IN,IV                                        COLEASE
      GO TO 9040                                                         COLEASE
 9030 CONTINUE                                                           COLEASE
      PRINT 903 , ISNAME,IY,IN,IV,IR,IR                                  COLEASE
 9040 CONTINUE                                                           COLEASE
      IERROR(7) = ISNAME(1)                                              COLEASE
      IERROR(8) = ISNAME(2)                                              COLEASE
      CALL  ABORTR  ( IERROR,30 )                                        COLEASE
      STOP                                                               COLEASE
      END                                                                COLEASE
```

```
      SUBROUTINE EXTRAC ( IY,IN )                                       COLEASE
      COMMON / APPRO  / ID(1)                                           COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION         ISNAME(2)                                       COLEASE
      DATA      ISNAME / 4HEXTR,4HACT  /                                COLEASE
C¬    DATA      NBITS  / 60 /                                           COLEASE
C                                                                       COLEASE
C-----SUBROUTINE EXTRAC EXTRACTS THE ATTRIBUTES FOR ENTRY IN OF ENTITY   COLEASE
C-----IY FROM THE STORAGE STACK AND PUTS THEM IN THE COMMON BLOCK FOR    COLEASE
C-----ENTITY IY                                                         COLEASE
C                                                                       COLEASE
C-----CHECK BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY                 COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                     IF ( IN . LT . 1 )             GO TO 9010          COLEASE
                     IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010            COLEASE
C-----SET UP THE PARAMETERS FOR EXTRAC                                  COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      IEA = IBA + IEN(IIEN+2) - 1                                       COLEASE
C-----EXTRACT EACH ATTRIBUTE FROM THE STORAGE STACK FOR ENTRY IN OF     COLEASE
C-----ENTITY IY                                                         COLEASE
      IIAT = (IBA-1)*3                                                  COLEASE
      DO 1010  I = IBA , IEA                                            COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C¬    ID(I) = LSHIFT((IS(IWD).AND.IAT(IIAT+3)),NBITS-IAT(IIAT+2))       COLEASE
C;    ID(I) = LSHIFT(IAND(IS(IWD),IAT(IIAT+3)),-IAT(IIAT+2))            COLEASE
      IIAT = IIAT + 3                                                   COLEASE
 1010 CONTINUE                                                          COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERROR AND STOP                              COLEASE
 9010 CONTINUE                                                          COLEASE
      CALL  SMEP    ( 0,IY,IN,0,1,ISNAME )                              COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE FIND   ( IR,IY,IN,IV )                                 COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION         ISNAME(2)                                       COLEASE
      DATA      ISNAME / 4HFIND,4H     /                                COLEASE
C¬    DATA      NBITS  / 60 /                                           COLEASE
C                                                                       COLEASE
C-----SUBROUTINE FIND FINDS THE VALUE OF ATTRIBUTE IV OF ENTRY IN OF    COLEASE
C-----ENTITY IY AND PUTS IT INTO LOCAL INTEGER IR                       COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                     IF ( IN . LT . 1 )             GO TO 9010          COLEASE
                     IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010            COLEASE
C-----CHECK THE BOUNDARIES FOR ATTRIBUTE NUMBER IV                      COLEASE
                     IF ( IV . LT . 1 )             GO TO 9020          COLEASE
                     IF ( IV . GT . IEN(IIEN+2) ) GO TO 9020            COLEASE
C-----SET UP THE PARAMETERS FOR FIND                                    COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      I = IBA + IV - 1                                                  COLEASE
      IIAT = (I-1)*3                                                    COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C-----FIND LOCAL INTEGER IR FROM ATTRIBUTE IV OF ENTRY IN OF ENTITY IV  COLEASE
C¬    IR = LSHIFT((IS(IWD).AND.IAT(IIAT+3)),NBITS-IAT(IIAT+2))          COLEASE
C;    IR = LSHIFT(IAND(IS(IWD),IAT(IIAT+3)),-IAT(IIAT+2))               COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IE = 1                                                            COLEASE
      GO TO 9030                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IE = 2                                                            COLEASE
 9030 CONTINUE                                                          COLEASE
      CALL  SMEP    ( 0,IY,IN,IV,IE,ISNAME )                            COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE REPACK ( IY,IN )                                       COLEASE
      COMMON / APPRO  / ID(1)                                           COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION          ISNAME(2)                                      COLEASE
      DATA      ISNAME / 4HREPA,4HCK    /                               COLEASE
C                                                                       COLEASE
C-----SUBROUTINE REPACK REPACKS THE VALUES OF THE ATTRIBUTES FROM THE    COLEASE
C-----COMMON BLOCK FOR ENTITY IY INTO ENTRY IN OF ENTITY IY IN THE       COLEASE
C-----STORAGE STACK                                                     COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                       IF ( IN . LT . 1 )             GO TO 9010        COLEASE
                       IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010          COLEASE
C-----SET UP THE PARAMETERS FOR REPACK                                  COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      ILW = IFW + NWE - 1                                               COLEASE
C-----ZERO OUT ENTRY IN OF ENTITY IY IN THE STORAGE STACK               COLEASE
      DO 1010  I = IFW , ILW                                            COLEASE
      IS(I) = 0                                                         COLEASE
 1010 CONTINUE                                                          COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      IEA = IBA + IEN(IIEN+2) - 1                                       COLEASE
      IIAT = (IBA-1)*3                                                  COLEASE
C-----REPACK EACH ATTRIBUTE FOR ENTRY IN OF ENTITY IY                   COLEASE
      DO 1020  I = IBA , IEA                                            COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
      IT = LSHIFT(ID(I),IAT(IIAT+2))                                    COLEASE
C¬    IX = IT . AND . (.NOT.IAT(IIAT+3))                                COLEASE
C;    IX = IAND(IT,INOT(IAT(IIAT+3)))                                   COLEASE
C-----IF ATTRIBUTE I IS OUT OF RANGE THEN ERROR                         COLEASE
                       IF ( IX . NE . 0 )             GO TO 9020        COLEASE
C¬    IS(IWD) = IT . OR . IS(IWD)                                       COLEASE
C;    IS(IWD) = IOR(IT,IS(IWD))                                         COLEASE
      IIAT = IIAT + 3                                                   COLEASE
 1020 CONTINUE                                                          COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IR = 0                                                            COLEASE
      IV = 0                                                            COLEASE
      IE = 1                                                            COLEASE
      GO TO 9030                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IR = ID(I)                                                        COLEASE
      IV = I - IBA + 1                                                  COLEASE
      IE = 3                                                            COLEASE
 9030 CONTINUE                                                          COLEASE
      CALL  SMEP    ( IR,IY,IN,IV,IE,ISNAME )                           COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE STORE  ( IR,IY,IN,IV )                                 COLEASE
      COMMON / ATTB   / IAT(1)                                          COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION          ISNAME(2)                                      COLEASE
      DATA      ISNAME / 4HSTOR,4HE     /                               COLEASE
C                                                                       COLEASE
C-----SUBROUTINE STORE STORES THE VALUE OF LOCAL INTEGER IR INTO        COLEASE
C-----ATTRIBUTE IV OF ENTRY IN OF ENTITY IY IN THE STORAGE STACK        COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY NUMBER IN FOR ENTITY IY            COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                       IF ( IN . LT . 1 )             GO TO 9010        COLEASE
                       IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010          COLEASE
C-----CHECK THE BOUNDARIES FOR ATTRIBUTE NUMBER IV                      COLEASE
                       IF ( IV . LT . 1 )             GO TO 9020        COLEASE
                       IF ( IV . GT . IEN(IIEN+2) ) GO TO 9020          COLEASE
C-----SET UP THE PARAMETERS FOR STORE                                   COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBA = IEN(IIEN+9)                                                 COLEASE
      I = IBA + IV - 1                                                  COLEASE
      IIAT = (I-1)*3                                                    COLEASE
      IWD = IFW + IAT(IIAT+1)                                           COLEASE
C-----STORE LOCAL INTEGER IR INTO ATTRIBUTE IV OF ENTRY IN OF ENTITY IY COLEASE
      IT = LSHIFT(IR,IAT(IIAT+2))                                       COLEASE
C¬    IX = IT . AND . (.NOT.IAT(IIAT+3))                                COLEASE
C;    IX = IAND(IT,INOT(IAT(IIAT+3)))                                   COLEASE
C-----IF LOCAL INTEGER IR IS OUT OF RANGE THEN ERROR                    COLEASE
                       IF ( IX . NE . 0 )             GO TO 9030        COLEASE
C¬    IS(IWD) = IT.OR.(IS(IWD).AND.(.NOT.IAT(IIAT+3)))                  COLEASE
C;    IS(IWD) = IOR(IT,IAND(IS(IWD),INOT(IAT(IIAT+3))))                 COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERRORS AND STOP                             COLEASE
 9010 CONTINUE                                                          COLEASE
      IE = 1                                                            COLEASE
      GO TO 9040                                                        COLEASE
 9020 CONTINUE                                                          COLEASE
      IE = 2                                                            COLEASE
      GO TO 9040                                                        COLEASE
 9030 CONTINUE                                                          COLEASE
      IE = 3                                                            COLEASE
 9040 CONTINUE                                                          COLEASE
      CALL  SMEP    ( IR,IY,IN,IV,IE,ISNAME )                           COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

```
      SUBROUTINE LOGIC  ( IY,IN )                                       COLEASE
      COMMON / ENTITY / IEN(1)                                          COLEASE
      COMMON / FUN    / IFU(1)                                          COLEASE
      COMMON / STACK  / IS(1)                                           COLEASE
      DIMENSION          ISNAME(2)                                      COLEASE
      DATA      ISNAME / 4HLOGI,4HC     /                               COLEASE
C                                                                       COLEASE
C-----SUBROUTINE LOGIC FINDS THE VALUE FOR THE LOGICAL DEPENDENT        COLEASE
C-----ATTRIBUTES FOR ENTRY IN OF ENTITY IY BASED ON THE VALUE OF THE    COLEASE
C-----LOGICAL INDEPENDENT ATTRIBUTES FOR ENTRY IN OF ENTITY IY IN THE   COLEASE
C-----STORAGE STACK AND STORES THEIR VALUES IN THE STORAGE STACK        COLEASE
C                                                                       COLEASE
C-----CHECK THE BOUNDARIES FOR ENTRY IN FOR ENTITY IY                   COLEASE
      IIEN = (IY-1)*9                                                   COLEASE
                     IF ( IN . LT . 1 )             GO TO 9010          COLEASE
                     IF ( IN . GT . IEN(IIEN+1) ) GO TO 9010            COLEASE
C-----SET UP THE PARAMETERS FOR LOGIC                                   COLEASE
      NWE = IEN(IIEN+3)                                                 COLEASE
      IFW = IEN(IIEN+4) + NWE*(IN-1)                                    COLEASE
      IBF = IEN(IIEN+8)                                                 COLEASE
      IEF = IBF + IEN(IIEN+7) - 1                                       COLEASE
      IIW = IFW + IEN(IIEN+6) - 1                                       COLEASE
      IDW = IIW + 1                                                     COLEASE
C-----ZERO OUT THE DEPENDENT ATTRIBUTES FOR ENTRY IN OF ENTITY IY IN THE COLEASE
C-----STORAGE STACK                                                     COLEASE
      ISIIW = IS(IIW)                                                   COLEASE
      ISIDW = 0                                                         COLEASE
C-----DETERMINE THE LOGICAL VALUE FOR EACH PATH IN THE BINARY NETWORK   COLEASE
C-----LEADING TO THE DEPENDENT ATTRIBUTES                               COLEASE
      IIFU = (IBF-1)*2                                                  COLEASE
      DO 1010  I = IBF , IEF                                            COLEASE
      LTF = 2                                                           COLEASE
C¬    IIV = ISIIW . AND . IFU(IIFU+1)                                   COLEASE
C!    IIV = IAND(ISIIW,IFU(IIFU+1))                                     COLEASE
C-----IF EACH INDEPENDENT ATTRIBUTE IS THE APPROPRIATE VALUE TO MAKE    COLEASE
C-----THE PATH TRUE THEN SET THE DEPENDENT ATTRIBUTE TRUE               COLEASE
                     IF ( IFU(IIFU+1) . EQ . IIV )LTF = 1               COLEASE
C-----STORE THE VALUE FOR THE DEPENDENT ATTRIBUTE IN THE STORAGE STACK  COLEASE
C¬    ISIDW = ISIDW . OR . LSHIFT(LTF,IFU(IIFU+2))                      COLEASE
C!    ISIDW = IOR(ISIDW,LSHIFT(LTF,IFU(IIFU+2)))                        COLEASE
      IIFU = IIFU + 2                                                   COLEASE
 1010 CONTINUE                                                          COLEASE
      IS(IDW) = ISIDW                                                   COLEASE
      RETURN                                                            COLEASE
C-----PROCESS THE EXECUTION ERROR AND STOP                              COLEASE
 9010 CONTINUE                                                          COLEASE
      CALL  SHEP    ( 0,IY,IN,0,1,ISNAME )                              COLEASE
      STOP                                                              COLEASE
      END                                                               COLEASE
```

PROGRAMMERS DOCUMENTATION

SIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKAGE

LATEST UPDATE: 27 AUG 77

THIS DOCUMENTATION IS DIVIDED INTO THE FOLLOWING SECTIONS:

1. SIMULATION PROCESSOR LIMITATIONS
2. EXPLANATION OF THE INPUT ERRORS
3. EXPLANATION OF THE EXECUTION ERRORS
4. DEFINITION OF THE ATTRIBUTES IN EACH ENTITY AND THE ROUTINES IN WHICH EACH ENTITY IS USED
5. DEFINITION OF THE VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED
6. DEFINITION OF THE LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL
7. ALPHABETICAL LISTING OF ALL THE ROUTINES AND THE ROUTINES WHICH CAN CALL THEM
8. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED
9. GENERALIZED CALLING SEQUENCE DIAGRAM

1. SIMULATION PROCESSOR LIMITATIONS

MAXIMUM NUMBER OF INBOUND APPROACHES	6
MAXIMUM NUMBER OF OUTBOUND APPROACHES	6
RANGE OF APPROACH NUMBERS	1-12
MAXIMUM SPEED LIMIT FOR APPROACHES	118 FT/SEC (80 MPH)
MAXIMUM NUMBER OF LANES PER APPROACH	6
MAXIMUM SIGHT DISTANCE RESTRICTIONS PER APPROACH	5
MAXIMUM NUMBER OF INBOUND LANES	25
MAXIMUM NUMBER OF OUTBOUND LANES	25
RANGE OF LANE NUMBERS	1-50
MAXIMUM LENGTH OF LANES	1000 FEET
MAXIMUM WIDTH OF LANES	15 FEET
MAXIMUM NUMBER OF DETECTORS PER LANE	5
MAXIMUM NUMBER OF INTERSECTION PATHS PER LANE	7
MAXIMUM NUMBER OF INTERSECTION PATHS	125
MAXIMUM LENGTH OF INTERSECTION PATHS	250 FEET
MAXIMUM SPEED LIMIT FOR INTERSECTION PATHS	118 FT/SEC (80 MPH)
MAXIMUM NUMBER OF CONFLICTS PER PATH	60
MAXIMUM NUMBER OF DRIVER CLASSES	5
MAXIMUM NUMBER OF VEHICLE CLASSES	15
MAXIMUM PERCEPTION-REACTION TIME	15*DT SECONDS
MAXIMUM POS SLOPE OF ACCEL/DECEL	8 FT/SEC/SEC/SEC
MAXIMUM NEG SLOPE OF ACCEL/DECEL	-12 FT/SEC/SEC/SEC
MAXIMUM ACCELERATION RATE	19.2 FT/SEC/SEC
MAXIMUM DECELERATION RATE	-32 FT/SEC/SEC
MAXIMUM SPEED OF VEHICLE	161 FT/SEC (110 MPH)
MAXIMUM AVERAGE DESIRED SPEED FOR 2000*DT SECONDS	129 FT/SEC (88 MPH)
MAXIMUM TIME IN SYSTEM	2000*DT SECONDS
MAXIMUM NUMBER OF VEHICLES IN THE SYSTEM	200
MAXIMUM NUMBER OF SIGHT DISTANCE RESTRICTIONS	20
MAXIMUM NUMBER OF INTERSECTION CONFLICTS	1000
MAXIMUM NUMBER OF CAM STACK ENTRIES	72
MAXIMUM NUMBER OF SIGNAL PHASES	8
MAXIMUM NUMBER OF DETECTORS	20
MAXIMUM NUMBER OF DETECTORS PER SIGNAL PHASE	10

2. EXPLANATION OF THE INPUT ERRORS

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE INITAL:

STOP 801 = END-OF-FILE ON FIRST READ OF GEOPRO INPUT ON TAPE <IGEOP>
(GEOMETRY PROCESSOR FILE EMPTY)
STOP 802 = END-OF-FILE ON FIRST READ OF DVPRO INPUT ON TAPE <IVEHP>
(DRIVER-VEHICLE PROCESSOR FILE EMPTY)
STOP 803 = END-OF-FILE ON FIRST READ OF SIMPRO INPUT ON TAPE <INPUT>
(TRAFFIC SIMULATION PROCESSOR INPUT EMPTY)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE RUSERD:

STOP 804 = START-UP TIME = <STRTIM> IS LT 2.0 OR GT 5.0
(START-UP TIME IS OUT OF RANGE 2.0-5.0)
STOP 805 = SIMULATION TIME = <SIMTIM> IS LT 10.0 OR GT 60.0
(SIMULATION TIME IS OUT OF RANGE 10.0-60.0)
STOP 806 = STEP INCREMENT FOR SIMULATION TIME = <DT> IS LT 0.5 OR GT 1.5
(STEP INCREMENT FOR SIMULATION TIME IS OUT OF RANGE 0.5-1.5)
STOP 807 = SPEED FOR DELAY BELOW XX MPH = <XMPH> IS LT 0.0 OR GT 40.0
(SPEED FOR DELAY BELOW XX MPH IS OUT OF RANGE 0.0-40.0)
STOP 808 = MAXIMUM CLEAR DISTANCE FOR BEING IN A QUEUE = <XQDIST> IS LT 4.0 OR GT 40.0
(MAXIMUM CLEAR DISTANCE FOR BEING IN A QUEUE IS OUT OF RANGE 4.0-44.0)
STOP 809 = CAR FOLLOWING EQUATION LAMBDA = <CAREQL> IS LT 0.0 OR GT 4.0
(CAR FOLLOWING EQUATION LAMBDA IS OUT OF RANGE 0.0-4.0)
STOP 810 = CAR FOLLOWING EQUATION MU = <CAREQM> IS LT 0.0 OR GT 4.0
(CAR FOLLOWING EQUATION MU IS OUT OF RANGE 0.0-4.0)
STOP 811 = CAR FOLLOWING EQUATION ALPHA = <CAREQA> IS LT 0.0 OR GT 9999.9
(CAR FOLLOWING EQUATION ALPHA IS OUT OF RANGE 0.0-9999.9)
STOP 812 = INTERSECTION TRAFFIC CONTROL = <ICONTR> IS LT 1 OR GT 7
(INTERSECTION TRAFFIC CONTROL IS OUT OF RANGE 1-7)
STOP 813 = SUMMARY STATISTICS PRINTED BY TURNING MOVEMENTS = (<IPTC>) IS NOT (YES) OR (NO)
(ILLEGAL CHARACTERS FOR SUMMARY STATISTICS PRINTED BY TURNING MOVEMENTS)
STOP 814 = SUMMARY STATISTICS PRINTED BY INBOUND APPROACH = (<IPAP>) IS NOT (YES) OR (NO)
(ILLEGAL CHARACTERS FOR SUMMARY STATISTICS PRINTED BY INBOUND APPROACH)
STOP 815 = LEAD TIME GAP FOR CONFLICT CHECKING = <TLEAD> IS LT 1.0 OR GT 3.0
(LEAD TIME GAP FOR CONFLICT CHECKING IS IS OUT OF RANGE 1.0-3.0)
STOP 816 = LAG TIME GAP FOR CONFLICT CHECKING = <TLAG> IS LT 1.0 OR GT 3.0
(LAG TIME GAP FOR CONFLICT CHECKING IS IS OUT OF RANGE 1.0-3.0)
STOP 817 = PUNCHED OUTPUT OF STATISTICS = <IPUNCH> IS NOT (YES) OR (NO)
(ILLEGAL CHARACTERS FOR PUNCHED OUTPUT OF STATISTICS)
STOP 818 = WRITE TAPE FOR POLLUTION DISPERSION MODEL = <IPOLL> IS NOT (YES) OR (NO)
(ILLEGAL CHARACTERS FOR WRITE TAPE FOR POLLUTION DISPERSION MODEL)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE RGEOPD:

STOP 819 = LANE CONTROL SPECIFIED FOR MORE THAN <NRLAN> LANES
(THERE WAS AT LEAST ONE EXTRA LANE CONTROL SPECIFIED THAT WAS NOT REQUIRED)
STOP 820 = LANE <I> LANE CONTROL = <LCONTR> IS LT 1 OR GT 7
(LANE CONTROL IS OUT OF RANGE 1-7)
STOP 821 = LANE <I> LANE CONTROL = <LCONTR> IS EQ 1 FOR INBOUND LANE
(OUTBOUND LANE CONTROL SPECIFIED FOR INBOUND LANE)
STOP 822 = LANE <I> LANE CONTROL = <LCONTR> IS NE 1 FOR OUTBOUND LANE
(INBOUND LANE CONTROL SPECIFIED FOR OUTBOUND LANE)
STOP 823 = LANE <I> LANE CONTROL = <LCONTR> IS GT 2 FOR INTERSECTION TRAFFIC CONTROL = 1
(UNCONTROLLED INTERSECTIONS MAY HAVE ONLY UNCONTROLLED LANES)
STOP 824 = LANE <I> LANE CONTROL = <LCONTR> IS GT 3 FOR INTERSECTION TRAFFIC CONTROL = 2
(YIELD SIGN CONTROLLED INTERSECTIONS MAY HAVE ONLY UNCONTROLLED AND YIELD SIGN CONTROLLED LANES)
STOP 825 = LANE <I> LANE CONTROL = <LCONTR> IS GT 4 FOR INTERSECTION TRAFFIC CONTROL = 3
(LESS-THAN-ALL-WAY STOP SIGN CONTROLLED INTERSECTIONS MAY HAVE ONLY UNCONTROLLED, YIELD SIGN CONTROLLED, AND STOP SIGN CONTROLLED LANES)
STOP 826 = LANE <I> LANE CONTROL = <LCONTR> IS LT 3 OR GT 4 FOR INTERSECTION TRAFFIC CONTROL = 4
(ALL-WAY STOP SIGN CONTROLLED INTERSECTIONS MAY HAVE ONLY YIELD SIGN CONTROLLED AND STOP SIGN CONTROLLED LANES)
STOP 827 = LANE <I> LANE CONTROL = <LCONTR> IS LT 3 OR EQ 4 FOR INTERSECTION TRAFFIC CONTROL GE 5
(SIGNAL CONTROLLED INTERSECTIONS MAY HAVE ONLY YIELD SIGN CONTROLLED, SIGNAL CONTROLLED, SIGNAL CONTROLLED WITH LEFT TURN ON RED, AND SIGNAL CONTROLLED WITH RIGHT TURN ON RED LANES)
STOP 828 = LANE <I> SIGNAL WITH LEFT TURN ON RED SPECIFIED FOR OTHER THAN MEDIAN LANE
(ONLY MEDIAN LANE MAY BE SPECIFIED FOR LEFT TURN ON RED)
STOP 829 = LANE <I> SIGNAL WITH RIGHT TURN ON RED SPECIFIED FOR OTHER THAN CURB LANE
(ONLY CURB LANE MAY BE SPECIFIED FOR RIGHT TURN ON RED)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE RCAMSD:

STOP 830 = NUMBER OF CAM STACK ENTRIES = <NCAMSP> IS LT 4 OR GT 72
(NUMBER OF CAM STACK ENTRIES IS OUT OF RANGE 4-72)
STOP 831 = CAM STACK <I> SIGNAL PHASE NUMBER = <ICAMPH(I)> IS LT 1 OR GT 8
(SIGNAL PHASE NUMBER IS OUT OF RANGE 1-8)
STOP 832 = CAM STACK <I> PHASE TIME = <IPHTIM> IS LT 1
(PRE-TIMED SIGNAL PHASE TIME IS OUT OF RANGE 1-999)
STOP 833 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = (<LANESS(K)>) IS NOT (L) (S) (R) (A) (U) OR ( )
(ILLEGAL FIRST CHARACTER FOR SIGNAL INDICATION FOR LANE)
STOP 834 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> SECOND CHARACTER = (<LANESS(K+1)>) IS NOT (G) (A) (R) (P) (N) OR ( )
(ILLEGAL SECOND CHARACTER FOR SIGNAL INDICATION FOR LANE)
STOP 835 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> THIRD CHARACTER = (<LANESS(K+2)>) IS NOT (G) (A) (R) (S) OR ( )
(ILLEGAL THIRD CHARACTER FOR SIGNAL INDICATION FOR LANE)
STOP 836 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = (<LANESS(K)>) SECOND CHARACTER = (<LANESS(K+1)>) THIRD CHARACTER = (<LANESS(K+2)>) IS AN ILLEGAL COMBINATION
(SIGNAL INDICATIONS SPG RPG SPA RPA SPR RPR ARE NOT ALLOWED)
STOP 837 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> SECOND CHARACTER = (<LANESS(K+1)>) IS NOT (G) (A) (R) OR (P) WHEN FIRST CHARACTER = (A)
(ILLEGAL SECOND CHARACTER FOR SIGNAL INDICATION FOR LANE WHEN THE FIRST CHARACTER IS (A) [ALL])
STOP 838 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = (A) AND SECOND CHARACTER = (<LANESS(K+1)>) BUT THIRD CHARACTER = (LANESS(K+2)>) IS NOT ( )
(ILLEGAL THIRD CHARACTER FOR SIGNAL INDICATION FOR LANE WHEN THE FIRST CHARACTER IS (A) [ALL] AND THE SECOND CHARACTER IS OK)
STOP 839 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = ( ) BUT SECOND CHARACTER = (<LANESS(K+1)>) IS NOT ( ) ALSO
(ILLEGAL SECOND CHARACTER FOR SIGNAL INDICATION FOR LANE WHEN THE FIRST CHARACTER IS ( ) INDICATING THREE BLANKS)
STOP 840 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = ( ) AND SECOND CHARACTER = ( ) BUT THIRD CHARACTER = (<LANESS(K+2)>) IS NOT ( ) ALSO
(ILLEGAL THIRD CHARACTER FOR SIGNAL INDICATION FOR LANE WHEN THE FIRST CHARACTER IS ( ) AND THE SECOND CHARACTER IS ( ) INDICATING THREE BLANKS)
STOP 841 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = ( ) AND SECOND CHARACTER = ( ) AND THIRD CHARACTER = ( ) FOR CAM STACK 1
(FIRST CAM STACK POSITION SIGNAL INDICATION MUST BE SPECIFIED)
STOP 842 = CAM STACK <I> LANE <J> INBOUND LANE <IBLN> FIRST CHARACTER = (<LANESS(K)>) SECOND CHARACTER = (<LANESS(K+1)>) THIRD CHARACTER = (<LANESS(K+2)>) IS ILLEGAL FOR UNSIGNALIZED LANE
(UNSIGNALIZED LANES MUST HAVE SIGNAL INDICATION CHARACTERS (UNS))

```
THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE RPHASD:

STOP 843 = NUMBER OF SIGNAL PHASES = <NPHASE> IS LT 2 OR GT 8
           (NUMBER OF SIGNAL PHASES IS OUT OF RANGE 2→8)
STOP 844 = SIGNAL PHASE NUMBER = <JP> IS LT 1 OR GT 8
           (SIGNAL PHASE NUMBER IS OUT OF RANGE 1→8)
STOP 845 = MORE THAN 1 SET OF DATA FOR SIGNAL PHASE <JP>
           (SIGNAL PHASE DATA DECLARED MORE THAN ONCE)
STOP 846 = SIGNAL PHASE = <JP> IS NOT IN THE CAM STACK
           (CAM STACK INFORMATION NOT ENTERED FOR SIGNAL PHASE)
STOP 847 = SIGNAL PHASE <JP> AMBER CLEARANCE INTERVAL = <TCI(JP)> IS LT 0.0
           (AMBER CLEARANCE INTERVAL TIME IS OUT OF RANGE 0.0→999.9)
STOP 848 = SIGNAL PHASE <JP> ALL-RED CLEARANCE INTERVAL = <TAR(JP)> IS LT 0.0
           (ALL-RED CLEARANCE INTERVAL TIME IS OUT OF RANGE 0.0→999.9)
STOP 849 = SIGNAL PHASE <JP> MAXIMUM EXTENSION AFTER DEMAND ON RED = <TMX(JP)>
           IS LT 0.0
           (MAXIMUM EXTENSION AFTER DEMAND ON RED TIME IS OUT OF RANGE
           0.0→999.9)
STOP 850 = SIGNAL PHASE <JP> SKIP PHASE SWITCH = <ISKP(JP)> IS NOT (ON )
           (OFF) OR (    )
           (ILLEGAL CHARACTERS FOR SKIP PHASE SWITCH OPTION)
STOP 851 = SIGNAL PHASE <JP> AUTO-RECALL SWITCH = <IREC(JP)> IS NOT (ON ) (OFF)
           OR (    )
           (ILLEGAL CHARACTERS FOR AUTO-RECALL SWITCH OPTION)
STOP 852 = SIGNAL PHASE <JP> PARENT/MINOR OPTION = <IMINOR(JP)> IS NOT (YES)
           (NO ) OR (    )
           (ILLEGAL CHARACTERS FOR PARENT/MINOR OPTION)
STOP 853 = SIGNAL PHASE <JP> DUAL LEFT OPTION = <IDUALL(JP)> IS NOT (YES) (NO )
           OR (    )
           (ILLEGAL CHARACTERS FOR DUAL LEFT OPTION)
STOP 854 = SIGNAL PHASE <JP> DETECTOR CONNECTION TYPE = <IANDOR(JP)> IS NOT
           (AND) (OR ) OR (    )
           (ILLEGAL CHARACTERS FOR DETECTOR CONNECTION TYPE)
STOP 855 = SIGNAL PHASE <JP> NUMBER OF DETECTORS FOR PHASE = <N> IS LT 0
           OR GT 10
           (NUMBER OF DETECTORS FOR PHASE IS OUT OF RANGE 0→10)
STOP 856 = SIGNAL PHASE <JP> IS ACTUATED BUT HAS NO DETECTORS AND THE
           AUTO-RECALL SWITCH = (OFF)
           (ILLEGAL ACTUATION CONFIGURATION FOR ACTUATED SIGNAL PHASE)
STOP 857 = SIGNAL PHASE <JP> AUTO-RECALL SWITCH = (ON ) BUT NUMBER OF
           DETECTORS = <N> IS NE 0
           (ILLEGAL ACTUATION CONFIGURATION FOR ACTUATED SIGNAL PHASE)
STOP 858 = SIGNAL PHASE <JP> DETECTOR NUMBER <N> = 0
           (DETECTOR NUMBER IS OUT OF RANGE 1→20)
STOP 859 = SIGNAL PHASE <JP> POSITIVE CONNECTED DETECTOR IS NOT FIRST ON LIST
           (ONLY ALL-RED REST PHASE MAY HAVE A NEGATIVELY CONNECTED DETECTOR
           AS THE FIRST ON ITS LIST)
STOP 860 = SIGNAL PHASE <JP> NUMBER OF SIGNAL PHASES CLEARED TO = <NN
           IS LT 1 OR GT 7
           (NUMBER OF PHASES CLEARED TO IS OUT OF RANGE 1→7)
STOP 861 = SIGNAL PHASE <JP> DUAL LEFT OPTION = (YES) BUT THE NUMBER OF
           PHASES CLEARED TO = <NN> IS LT 3
           (DUAL LEFT PHASE MUST HAVE AT LEAST 3 PHASES TO CLEAR TO)
STOP 862 = SIGNAL PHASE <JP> CAN NOT CLEAR TO ITSELF
           (PHASE NUMBER CAN NOT BE ON LIST OF PHASES THAT CAN BE CLEARED TO)
STOP 863 = SIGNAL PHASE <JP> PHASE CLEARED TO = <LPHNXT(J,JP)> IS NOT IN
           THE CAM STACK
           (CAM STACK INFORMATION HAS NOT BEEN ENTERED FOR SIGNAL PHASE THAT
           CAN BE CLEARED TO)
STOP 864 = SIGNAL PHASE <JP> NUMBER OF ENTRIES IN THE CAM STACK = <NCAM> IS
           NE 1+(NUMBER OF SIGNAL PHASES CLEARED TO)+(ALL-RED) = <MCAM>
           (INCOMPATIBLE NUMBER OF ENTRIES IN THE CAM STACK BASED ON SIGNAL
           PHASE TIMING)
STOP 865 = SIGNAL PHASE <JP> DUAL LEFT OPTION = (YES) BUT THE FIRST
           PHASE CLEARED TO = <LPHNXT(1,JP)> IS NOT <JPP1>
           (FIRST PHASE CLEARED TO MUST BE THE DUAL LEFT PHASE NUMBER PLUS 1
           FOR THE DUAL LEFT PHASE)
STOP 866 = SIGNAL PHASE <JP> DUAL LEFT OPTION = (YES) BUT THE SECOND
           PHASE CLEARED TO = <LPHNXT(2,JP)> IS NOT <JPP2>
           (SECOND PHASE CLEARED TO MUST BE THE DUAL LEFT PHASE NUMBER PLUS 2
           FOR THE DUAL LEFT PHASE)
STOP 867 = SIGNAL PHASE <JP> IS IN THE CAM STACK FOR THE SIGNAL BUT NO OTHER
           DATA WAS ENTERED
           (SIGNAL PHASE TIMING DATA NOT ENTERED FOR A PHASE IN THE CAM STACK)
STOP 868 = SIGNAL PHASE <I> DID NOT HAVE THE ALL-RED REST PHASE AS THE LAST
           PHASE ON ITS LIST OF PHASES TO CLEAR TO
           (WHEN AN ALL-RED REST PHASE EXISTS, EVERY OTHER PHASE MUST HAVE THE
           ALL-RED REST PHASE AS THE LAST ON ITS LIST OF PHASES TO CLEAR TO)

THE FOLLOWING INPUT ERRORS ARE DETECTED IN SUBROUTINE RLOOPD:

STOP 869 = NUMBER OF DETECTORS = <NLOOPS> IS LT 1 OR GT 20
           (NUMBER OF DETECTORS IS OUT OF RANGE 1→20)
STOP 870 = DETECTOR NUMBER = <JL> IS LT 1 OR GT 20
           (DETECTOR NUMBER IS OUT OF RANGE 1→20)
STOP 871 = MORE THAN 1 SET OF DATA FOR DETECTOR <JL>
           (DETECTOR DATA DECLARED MORE THAN ONCE)
STOP 872 = DETECTOR <JL> DETECTOR TYPE = (<ITYPLD(JL)><IT1>) IS NOT (PULSE    )
           (PRESENCE) OR (          )
           (ILLEGAL CHARACTERS FOR DETECTOR TYPE)
STOP 873 = DETECTOR <JL> STARTING POSITION = <LDSTRT> IS LT 0
           (DETECTOR STARTING POSITION IS OUT OF RANGE 0→<END OF LANE>)
STOP 874 = DETECTOR <JL> STOPPING POSITION = <LDSTOP> IS LT STARTING POSITION =
           <LDSTRT>
           (DETECTOR STOPPING POSITION IS OUT OF RANGE <DETECTOR STARTING
           POSITION>→<END OF LANE>)
STOP 875 = DETECTOR <JL> APPROACH NUMBER = <LDA> IS NOT ON LIST OF INBOUND
           APPROACHES
           (ILLEGAL INBOUND APPROACH NUMBER SPECIFIED)
STOP 876 = DETECTOR <JL> NUMBER OF LANE NUMBERS = <NLDLN> IS LT 1 OR GT 6
           (NUMBER OF LANE NUMBERS IS OUT OF RANGE 1→6)
STOP 877 = DETECTOR <JL> LANE NUMBER = <ILDLN> IS LT 1 OR GT NUMBER OF LANES
           FOR APPROACH <LDA> = <NLANES>
           (DETECTOR LANE NUMBER IS OUT OF RANGE 1→<NUMBER OF LANES FOR INBOUND
           APPROACH>)
STOP 878 = APPROACH <LDA> NUMBER OF DETECTORS FOR LANE <ILDLN> = <NLDL> IS GT 5
           (NUMBER OF DETECTORS FOR INBOUND LANE IS OUT OF RANGE 0→5)
STOP 879 = DETECTOR <JL> APPROACH <LDA> LANE <ILDLN> IS NOT AVAILABLE AT THE
           INTERSECTION
           (LANE FOR DETECTOR HAS LGEOM(3) = LGEOM(4) THUS NO VEHICLES MAY
           ENTER THE INTERSECTION FROM THE LANE THUS CREATING AN ILLEGAL
           DETECTOR CONFIGURATION)
STOP 880 = DETECTOR <JL> STOPPING POSITION = <LDSTOP> IS GT END OF LANE FOR
           APPROACH <LDA> LANE <ILDLN> = <LGEOM4>
           (DETECTOR STOPPING POSITION IS OUT OF RANGE <DETECTOR STARTING
           POSITION>→<END OF LANE>)
STOP 881 = DETECTOR <JL> IS ON LIST OF DETECTORS FOR PHASE <I> BUT NO
           OTHER DATA WAS ENTERED
           (DETECTOR DATA NOT ENTERED FOR A DETECTOR DECLARED FOR A PHASE)
STOP 882 = DETECTOR <JL> DATA WAS ENTERED BUT DID NOT APPEAR ON THE LIST OF
           DETECTORS FOR ANY SIGNAL PHASE AS POSITIVE
           (DETECTOR MUST BE POSITIVELY CONNECTED TO AT LEAST 1 PHASE)

THE FOLLOWING INPUT ERROR IS DETECTED IN SUBROUTINE RDVPRD:

STOP 883 = AVERAGE PIJR = <APIJR> IS LT MINIMUM PIJR = <PIJRMI>
           (OLD STYLE DRIVER-VEHICLE PROCESSOR TAPE READ BY SIMPRO)
```

3. EXPLANATION OF THE EXECUTION ERRORS

```
STOP 901 IN SSINTR = LIBL NOT ON LLANES FOR JSNA
                     (CAN NOT GET HERE HALT)
STOP 902 IN LOGIOB = LNEXT IS NOT ON LLANES LIST
                     (CAN NOT GET HERE HALT)
STOP 903 IN LCHDES = LEGAL NOT CHECKED
                     (CAN NOT GET HERE HALT)
STOP 904 IN LCHDES = ILLEGAL TURN CODE
                     (CAN NOT GET HERE HALT)
STOP 905 IN LCHDES = TRYING TO CHANGE LANES WHEN NO LANE ALTERNATIVE EXISTS
                     (CAN NOT GET HERE HALT)
STOP 906 IN ACDCP  = NO VEHD DEPENDENT ATTRIBUTE TRUE
                     (CAN NOT GET HERE HALT)
STOP 907 IN ACDCP  = STOPPED VEHICLES NOT PROGRAMMED YET
                     (CURRENTLY A CAN NOT GET HERE HALT)
STOP 908 IN ADLVAI = IV ALREADY ON LVATIN
                     (CAN NOT GET HERE HALT)
STOP 909 IN ADLVAI = NVATIN GT 25
                     (CAN NOT GET HERE HALT)
STOP 910 IN INTLOG = NO LANE CONTROL SET
                     (CAN NOT GET HERE HALT)
STOP 911 IN INTLOG = NO VEHIL DEPENDENT ATTRIBUTE TRUE
                     (CAN NOT GET HERE HALT)
STOP 912 IN SIGRES = JSISET LE 0 OR GT 25
                     (CAN NOT GET HERE HALT)
STOP 913 IN CHKCON = INFINITE LOOP
                     (CAN NOT GET HERE HALT)
STOP 914 IN SETCON = LNEXT EQ 0
                     (CAN NOT GET HERE HALT)
STOP 915 IN INFLZN = LCONTR EQ 1
                     (CAN NOT GET HERE HALT)
STOP 916 IN PATHF  = NO INTERSECTION PATHS FROM LANE FOR FORCED PATH
                     (CAN NOT GET HERE HALT)
STOP 917 IN CHKMLN = LANE DOES NOT EXIST AT POSNEW
                     (CAN NOT GET HERE HALT)
STOP 918 IN CHKMLN = NO LANE ALTERNATIVE FOR BLOCKED LANE
                     (NO LANES AVAILABLE AT THE INTERSECTION FOR APPROACH)
STOP 919 IN BANGS  = NO LANE ON LIST MATCHES MPRES
                     (CAN NOT GET HERE HALT)
STOP 920 IN LOGIN  = MORE THAN 200 VEHICLES IN SYSTEM
                     (CHECK EVERYTHING - IF OK THEN CONTACT AGENCY SUPPLYING
                     PROGRAM AND REQUEST MODIFICATION OF PROGRAM TO
                     ACCOMMODATE MORE THAN 200 VEHICLES IN THE SYSTEM)
STOP 921 IN ACTSIG = NO DEMAND FOR ANY PHASES ON LPHNXT LIST
                     (CAN NOT GET HERE HALT)
```

4. DEFINITION OF THE ATTRIBUTES IN EACH ENTITY AND THE ROUTINES IN WHICH EACH ENTITY IS USED

```
APPRO              ENTITY FOR APPROACHES (12 ENTRIES)
                       SIMPRO RGEOPD OBAP   LOGOUT LOGIOB IBAP   CHKDSP SSIBAP
                       LOGIBI CHGMLN ADLVAI CHKSDR SETPTV PATHF  LOGIN  ABORTR
                       EXTRAC REPACK

   IALEFT          ENTRY NUMBER OF APPROACH TO THE LEFT [1→12]
   ISDRA(5)        LIST OF ENTRY NUMBERS OF APPROACH FOR SIGHT DISTANCE
                   RESTRICTION [1→12]
   ISDRN(5)        LIST OF ENTRY NUMBERS FOR SDR ENTITY OF SIGHT DISTANCE
                   RESTRICTION [1→30]
   ISLIM           THE LEGAL SPEED LIMIT (FT/SEC) [0→118]
   LLANES(6)       LIST OF ENTRY NUMBERS FOR LANE ENTITY OF LANES IN THE
                   APPROACH, SUBSCRIPTED BY LANE NUMBER COUNTED FROM MEDIAN
                   TO CURB [1→50]
   NLANES          NUMBER OF LANES [1→6]
   NSDR            NUMBER OF SIGHT DISTANCE RESTRICTIONS [0→5]
   NVIL(6)         NUMBER OF VEHICLES IN EACH LANE, SUBSCRIPTED BY LANE
                   NUMBER [0→63]

CONFLT             ENTITY FOR INTERSECTION CONFLICTS (1000 ENTRIES)
                       SIMPRO RGEOPD CLRCON CHKSDR CHKCON SETCON UNSETC ABORTR

   ICONA(2)        ENTRY NUMBER FOR APPRO ENTITY OF LINKING INBOUND APPROACH
                   FOR INTERSECTION PATH ICONP() INVOLVED IN THE INTERSECTION
                   CONFLICT
   ICONAN          CONFLICT ANGLE MEASURED FROM FIRST INTERSECTION PATH
                   CLOCKWISE [0→360]
   ICOND(2)        DISTANCE DOWN INTERSECTION PATH FROM START OF INTERSECTION
                   PATH TO CONFLICT [0→250]
   ICONI(2)        INDEX NUMBER FOR IGEOCP AND ICPSET ARRAYS IN PATH ENTITY
                   FOR ENTRY ICONP() [1→60]
   ICONP(2)        ENTRY NUMBER FOR PATH ENTITY OF INTERSECTION PATHS INVOLVED
                   IN INTERSECTION CONFLICT [1→125]
   ICONV(2)        ENTRY NUMBER FOR VEH ENTITIES OF NEXT VEHICLE ON INTERSECTION
                   PATH ICONP() THAT HAS NOT CLEARED THE INTERSECTION CONFLICT
                   [1→200]
   IDUMCO          DUMMY VARIABLE FOR CONFLT ENTITY TO MAKE NUMBER OF
                   ATTRIBUTES EVEN

LANE               ENTITY FOR THE LANES IN THE APPROACHES  (50 ENTRIES)
                       SIMPRO RGEOPD OBAP   LOGOUT LOGIOB IBAP   LOKIBI CHKLDT
                       LOGIBI PRESTI LCHDES SVEHU  DELAY  CKLALT CHGMLN ACDCP
                       CHKSDR CHKCON SETPTV INFLZN PATHF  CHKMLN BANGS  LOGIN
                       ABORTR

   IBLN            INBOUND LANE NUMBER (FOR INDEXING ARRAY ISISET IN /SIGCAM/)
   IDUMLA          DUMMY VARIABLE FOR LANE ENTITY TO MAKE NUMBER OF
                   ATTRIBUTES EVEN
   IFVL            ENTRY NUMBER FOR VEH ENTITIES OF FIRST VEHICLE IN LANE
                   [0→200]
   ILVL            ENTRY NUMBER FOR VEH ENTITIES OF LAST VEHICLE IN LANE
                   [0→200]
   ISNA            ENTRY NUMBER FOR APPRO ENTITY OF APPROACH CONTAINING LANE
                   [1→12]
   LCONTR          TRAFFIC CONTROL INDICATOR FOR THIS LANE: [1→7]
                       1=OUTBOUND LANE
                       2=NO CONTROL
                       3=YIELD SIGN CONTROL
                       4=STOP SIGN CONTROL
                       5=SIGNAL CONTROL
                       6=SIGNAL WITH LEFT TURN ON RED
                       7=SIGNAL WITH RIGHT TURN ON RED
   LGEOM(4)        BEGINNING AND END POINTS OF LANE [0→1000]
                       (1)=FIRST BEGINNING POINT
                       (2)=FIRST END POINT
```

```
                (3)=SECOND BEGINNING POINT
                (4)=SECOND END POINT
LINTP(7)     LIST OF ENTRY NUMBERS FOR PATH ENTITY OF INTERSECTION PATHS
             INTO THE INTERSECTION [1,125]
LLDL(5)      LIST OF INDEX NUMBERS FOR /LOOPS/ OF THE DETECTOR
             FOR LANE [1,20]
LTURN        TURN CODE OF THE LANE: [0,15]
                  0=OUTBOUND OR BLOCKED INBOUND
                  1=                  RIGHT
                  2=          STRAIGHT
                  3=          STRAIGHT RIGHT
                  4=     LEFT
                  5=     LEFT         RIGHT
                  6=     LEFT STRAIGHT
                  7=     LEFT STRAIGHT RIGHT
                  8=U-TURN
                  9=U-TURN            RIGHT
                 10=U-TURN    STRAIGHT
                 11=U-TURN    STRAIGHT RIGHT
                 12=U-TURN LEFT
                 13=U-TURN LEFT       RIGHT
                 14=U-TURN LEFT STRAIGHT
                 15=U-TURN LEFT STRAIGHT RIGHT
LWID         WIDTH OF LANE (FEET) [8,15]
NLDL         NUMBER OF DETECTORS IN LANE [0,5]
NLL          ENTRY NUMBER OF LANE TO LEFT [1,50]
NLR          ENTRY NUMBER OF LANE TO RIGHT [1,50]
NPINT        NUMBER OF INTERSECTION PATHS INTO THE INTERSECTION [0,7]

PATH         ENTITY FOR INTERSECTION PATHS THROUGH THE INTERSECTION
             (125 ENTRIES)
                 SIMPRO RGEOPD INTERP LOKIOB SSINTR CLRCON LOGIOB LOGIBI
                 LSTOP  CHKSDR CHKCON SETPTV SETCON UNSETC ABORTR

ICPSET(60)   IS THERE IS A VEHICLE WHICH HAS THE RIGHT TO ENTER THE
             INTERSECTION ON THE INTERSECTION PATH WHICH CONFLICTS
             WITH ME AT MY IGEOCP() CONFLICT AND WHICH HAS NOT PASSED
             THE POINT OF INTERSECTION CONFLICT [0,1]
                 0=NO
                 1=YES
IFVP         ENTRY NUMBER FOR VEH ENTITIES OF FIRST VEHICLE IN THE
             INTERSECTION PATH [0,200]
IGEOCP(60)   LIST OF ENTRY NUMBERS FOR CONFLT ENTITY FOR THE GEOMETRIC
             INTERSECTION CONFLICT POINTS [1,1000]
ILCH         LANE CHANGE WITHIN THE INTERSECTION FLAG
                 0=NO
                 1=YES
ILVP         ENTRY NUMBER FOR VEH ENTITIES OF LAST VEHICLE IN THE
             INTERSECTION PATH [0,200]
IOPT         INTERSECTION PATH OPTION  [0,1]
                 0=PRIMARY
                 1=OPTION1
IPT          INTERSECTION PATH TURN CODE: [1,8]
                 1=                RIGHT
                 2=        STRAIGHT
                 4=   LEFT
                 8=U-TURN
LENP         THE LENGTH OF THE INTERSECTION PATH (FEET) [1,250]
LIBL         ENTRY NUMBER FOR LANE ENTITY OF LINKING INBOUND LANE
             [1,50]
LIMP         THE MINIMUM OF THE PHYSICAL SPEED LIMIT OF THE INTERSECTION
             PATH AND THE LEGAL SPEED LIMIT OF THE LINKING APPROACHES
             (FT/SEC) [0,118]
LOBAP        ENTRY NUMBER FOR APPRO ENTITY OF LINKING OUTBOUND
             APPROACH [1,12]
LOBL         ENTRY NUMBER FOR LANE ENTITY OF LINKING OUTBOUND LANE
             [1,50]
NCPSET       NUMBER OF INTERSECTION CONFLICT POINTS SET; SUM OF ICPSET
             ARRAY [0,60]
NGEOCP       NUMBER OF GEOMETRIC CONFLICT POINTS [0,60]
```

```
SDR          ENTITY FOR SIGHT DISTANCE RESTRICTIONS (30 ENTRIES)
                 SIMPRO RGEOPD ABORTR

ICANSE(40)   DISTANCE DOWN THE CENTER OF AN INBOUND APPROACH WHICH IS
             JUST VISIBLE BY THE APPROACH THE VEHICLE IS ON (INDEXED
             BY THE POSITION OF THE VEHICLE DIVIDED BY 25 FEET PLUS 1)

VEHD         ENTITY FOR DYNAMIC VEHICLE ATTRIBUTES (200 ENTRIES)
                 SIMPRO OBAP   SSOBAP LOGOUT INTERP LOKIOB SSINTR CLRCON
                 LOGIOB IBAP   LOKIBI CHKDSP SSIBAP LOGIBI PRESTI PREST2
                 UNBIAS NEWVEL LCHGEO ENDLCH LCHDES SVEHU  DELAY  CKLALT
                 GAPACC CHGMLN ACDCP  CARFOL ACCEL  CRIDIS ADLVAI INTLOG
                 SIGRES LSTOP  CHKSDR CHKCON SETPTV SETCON UNSETC INFLZN
                 PATHF  CHKMLN BANGS  BIAS   LOGIN  ABORTR

IACC         ACCELERATION/DECELERATION (BIASED FT/SEC/SEC) [0,16000]
IDTS         DISTANCE TRAVELED FOR STATISTICS (BIASED FEET) [0,56250]
IDVS         DELAY BELOW XX MPH FOR STATISTICS (IN DT,S) [0,2000]
IPOS         POSITION OF FRONT BUMPER OF VEHICLE (BIASED FEET)
             [0,25000]
IPRTM        PERCEPTION-REACTION TIME COUNTER FOR ACCEL/DECEL LOGIC
             (IN DT,S) [0,7]
IQDS         QUEUE DELAY FOR STATISTICS (IN DT,S) [0,2000]
ISDS         STOP DELAY FOR STATISTICS (IN DT,S) [0,2000]
ISET         LANE CHANGE DECISION FLAG: [1,7]
                 1=GAP IS ACCEPTED, CHANGE LANE
                 2=SLOW DOWN, POSSIBLE ACCEPTANCE NEXT TIME
                 3=SPEED UP, POSSIBLE ACCEPTANCE NEXT TIME
                 4=REJECT GAP, SLOW DOWN AND LOOK AT NEXT GAP
                 5=REJECT GAP, CONTINUE AS BEFORE
                 6=DO NOT CHECK FOR LANE CHANGE
                 7=VEHICLE IN ADJACENT LANE IS MOVING INTO THE
                   INTERSECTION, IN THE 4-WAY-STOP CASE
ISLP         ACCELERATION/DECELERATION SLOPE (BIASED FT/SEC/SEC/SEC)
             [0,8000]
ISPDP        0/1 FOR NO/YES IF VEHICLE HAS SET DESIRED SPEED FOR
             INTERSECTION PATH [0,1]
ISPDS        SUM OF DESIRED SPEED OF VEHICLE FOR EACH DT FOR
             STATISTICS [0,258134]
ISTCON       INDEX NUMBER FOR ICPSET/IGEOCP ARRAY IN PATH ENTITY OF
             NEXT INTERSECTION CONFLICT THAT REAR BUMPER HAS NOT CLEARED
             (EQUIVALENCED TO LALT) [0,61]
ITIMV        TIME OF VEHICLE IN SYSTEM (IN DT,S) [0,2000]
             (EQUIVALENCED TO LALT)
IVEL         VELOCITY (BIASED FT/SEC) [0,4034]
IVMAXA       VEHICLE MAXIMUM ACCELERATION FOR STATISTICS
             (BIASED FT/SEC/SEC) [0,150]
IVMAXD       VEHICLE MAXIMUM DECELERATION FOR STATISTICS
             (BIASED FT/SEC/SEC) [0,250]
LALT         LANE ALTERNATIVES (EQUIVALENCED TO ISTCON) [1,5]
                 1=THERE ARE NO ALTERNATIVES; THE PRESENT LANE IS
                   THE ONLY POSSIBLE ONE
                 2=THE RIGHT LANE IS THE ONLY ALTERNATIVE
                 3=THE LEFT LANE IS THE ONLY ALTERNATIVE
                 4=BOTH RIGHT AND LEFT LANES ARE ALTERNATIVES
                 5=LANE ALTERNATIVES HAVE NOT BEEN CHECKED
LATPOS       LATERAL POSITION OF THE VEHICLE DURING A LANE
             CHANGE; NUMBER OF FEET REMAINING TO MOVE LATERALLY
             TO BE AT CENTER OF NEW LANE (BIASED FEET) [0,250]
             (AMBER GO WHEN STOPPED)
LCHGE        LANE CHANGE INFORMATION FLAG: [1,3]
                 1=NO LANE CHANGE
                 2=VEHICLE IS CHANGING LANE
                 3=A VEHICLE AHEAD IS CHANGING LANE
LEGAL        TOTAL LATERAL DISTANCE FOR LANE CHANGE (FEET) [0,30]
             (WHEN LCHGE = 2)
LEGAL        LANE CHANGE DESIRABILITY FLAG: [1,5]
                 1=TURN IS LEGAL FROM APPROACH, BUT NOT FROM LANE,
```

```
                       THEREFORE CHANGE LEFT
                     2=TURN REQUESTED IS LEGAL FROM PRESENT LANE
                     3=TURN IS LEGAL FROM APPROACH, BUT NOT FROM LANE,
                       THEREFORE CHANGE RIGHT
                     4=DESIRABILITY OF LANE CHANGE HAS NOT BEEN CHECKED
                     5=TURN REQUESTED IS ILLEGAL FROM APPROACH
   LOGFLG        FLAG TO CONTROL THE CALLING OF GENERAL INTERSECTION
                 LOGIC  (SEE ALSO LOGTMP IN /INDEX/): [0-7]
                     0=DO NOT CALL LOGIC, DO NOT EXTRACT VEHIL, AND
                       DO NOT CALL INTLOG
                     1=CALL LOGIC, EXTRACT VEHIL, CALL INTLOG,
                       AND POSSIBLY CALL CONFLT
                     2-7=DO NOT CALL LOGIC, EXTRACT VEHIL, CALL INTLOG,
                         AND DO NOT CALL CONFLT
   NORC          ENTRY NUMBER OF NEAREST VEHICLE TO THE REAR FOR INTERSECTION
                 CONFLICT CHECKING (=201 FOR INTERSECTION CONFLICTS NOT SET)
                 [0-201]

 VEHD LOGICAL INDEPENDENT ATTRIBUTES (ASK QUESTIONS)

   MBLOCK        DOES LANE END BEFORE END OF APPROACH [T/F]
   MFINL         IS VEHICLE FIRST IN LANE [T/F]
   MININT        HAS THE VEHICLE ENTERED THE INTERSECTION [T/F]
   MLAG          SHOULD VEHICLE YIELD TO A VEHICLE TRYING TO CHANGE LANES
                 [T/F]
   MOASF         IS VEHICLE AHEAD STOPPED (IF NO VEHICLE, ALWAYS TRUE)
                 [T/F]
   MPOBS         IS VEHICLE PARKED OR IS BUS STOPPED [T/F]
   MPRO          IS VEHICLE PROCEED=INTO=INTERSECTION FLAG SET [T/F]
   MSAOR         IS VEHICLE STOPPED AT OBJECT REQUIRING THE STOP
                 (WITHIN 10 FEET OF PREVIOUS VEHICLE OR STOP LINE) [T/F]
   MSFLG         IS VEHICLE STOPPING FLAG SET [T/F]
   MSTPF         IS VEHICLE STOPPED [T/F]
   MTCARS        DOES TRAFFIC CONTROL AHEAD REQUIRE VEHICLE TO STOP [T/F]

 VEHD LOGICAL DEPENDENT ATTRIBUTES (ANSWER QUESTIONS) (ONLY 1 SET TRUE)

   IACDS         ACCELERATE ACCORDING TO DESIRED SPEED [T/F]
   IACLDS        ACCELERATE ACCORDING TO LEAD VEHICLE SPEED [T/F]
   ICDFS         CONTINUE DECELERATION FOR STOP [T/F]
   IFVA          FOLLOW VEHICLE AHEAD IF WITHIN CAR FOLLOWING DISTANCE,
                 OTHERWISE ACCELERATE [T/F]
   IRSTOP        REMAIN STOPPED [T/F]
   ISDEC         INITIATE DECELERATION FOR STOP IF CRITICAL STOPPING
                 DISTANCE VIOLATED, OTHERWISE ACCELERATE ACCORDING TO
                 DESIRED SPEED [T/F]
   ISTMO         CHECK IF PARKED VEHICLE (OR STOPPED BUS) SHOULD START
                 TO MOVE [T/F]


 VEHF            ENTITY FOR FIXED VEHICLE ATTRIBUTES (200 ENTRIES)
                     SIMPRO OBAP   SSOBAP LOGOUT FLGNOR INTERP LOKIOB SSINTR
                     CLRCON LOGIOB IBAP   LOKIBI CHKDSP CHKLDT SSIBAP LOGIBI
                     PREST1 UNBIAS LCHGEO ENDLCH LCHDES DELAY  CKLALT GAPACC
                     CHGMLN ACDCP  CARFOL ACCEL  CRIDIS ADLVAI INTLOG SIGRES
                     LSTOP  CHKSDR CHKCON SETPTV SETCON UNSETC INFLZN PATHF
                     BANGS  BIAS   LOGIN  ABORTR

   IBAPS         INDEX NUMBER FOR LIBA ARRAY OF /INTER/ FOR INBOUND
                 APPROACH NUMBER FOR STATISTICS [1-6]
   IDRICL        DRIVER CLASS NUMBER [1-5]
   IEXTIM        EXTRA TIME AT LOGIN (PORTION OF DT) [0-25]
   IPRTLO        0/1 FOR NO/YES FOR PRINTING INDIVIDUAL VEHICLE STATISTICS
                 AT LOGOUT [0-1]
   ISPD          VEHICLE DESIRED SPEED (FT/SEC) [0-161]
   ITURN         TURN CODE OF VEHICLE FOR STATISTICS: [0-3]
                     0=VEHICLE NOT DEDICATED TO AN INTERSECTION PATH  YET
                     1=U=TURN AND LEFT TURN
                     2=STRAIGHT
                     3=RIGHT TURN
   IVEHCL        VEHICLE CLASS NUMBER [1-15]
```

```
   LNEXT         ENTRY NUMBER FOR LANE OR PATH ENTITIES OF NEXT LINK
                 [0-125]
   LPRES         ENTRY NUMBER FOR LANE OR PATH ENTITIES OF PRESENT LINK
                 [0-125]
   NOBAPD        ENTRY NUMBER FOR APPRO ENTITY OF DESIRED OUTBOUND
                 APPROACH [1-12]
   NOF           ENTRY NUMBER OF NEAREST VEHICLE TO FRONT [0-200]
   NOR           ENTRY NUMBER OF NEAREST VEHICLE TO REAR [0-200]


 VEHIL           ENTITY FOR VEHICLE INTERSECTION LOGIC (200 ENTITIES)
                     SIMPRO IBAP   CHGMLN ACDCP  INTLOG SIGRES LSTOP  CHKSDR
                     CHKCON INFLZN LOGIN  ABORTR

   MDUMIL        DUMMY VARIABLE FOR VEHIL ENTITY TO MAKE THE NUMBER OF
                 ATTRIBUTES EVEN

 VEHIL LOGICAL INDEPENDENT ATTRIBUTES (ASK QUESTIONS)

   MATSTL        IS VEHICLE STOPPED AT THE STOP LINE [T/F]
   MCHKCF        MUST VEHICLE CHECK INTERSECTION CONFLICTS [T/F]
   MDEDIC        IS VEHICLE DEDICATED TO AN INTERSECTION PATH [T/F]
   MINFLZ        IS VEHICLE WITHIN THE INFLUENCE ZONE OF THE INTERSECTION
                 CONTROL [T/F]
   MIUNC         IS THE INTERSECTION UNCONTROLED [T/F]
   MLRTOR        MAY VEHICLE MAKE A LEFT=TURN=ON=RED OR RIGHT=TURN=
                 ON=RED [T/F]
   MLSTOP        IS THIS LANE CONTROLLED BY A STOP SIGN [T/F]
   MLUNC         IS THE LANE UNCONTROLLED [T/F]
   MLYELD        IS THIS LANE CONTROLLED BY A YIELD SIGN [T/F]
   MSSGRN        IS SIGNAL SETTING FOR THIS LANE SHOWING GREEN [T/F]
   MSSRED        IS SIGNAL SETTING FOR THIS LANE SHOWING RED [T/F]

 VEHIL LOGICAL DEPENDENT ATTRIBUTES (ANSWER QUESTIONS) (ONLY 1 SET TRUE)

   ICHKCF        CHECK INTERSECTION CONFLICTS [T/F]
   ICONTN        CONTINUE AS FAR AS INTERSECTION LOGIC IS CONCERNED
                 [T/F]
   IDEDIC        CHECK IF IT IS TIME FOR VEHICLE TO DEDICATE HIMSELF TO
                 AN INTERSECTION PATH [T/F]
   IERROR        VEHIL LOGIC ERROR [T/F]
   ILSTOP        FOLLOW STOP SIGN CONTROLLED LANE LOGIC [T/F]
   ILUNC         FOLLOW UNCONTROLLED LANE LOGIC [T/F]
   ILYELD        FOLLOW YIELD SIGN CONTROLLED LOGIC [T/F]
   INFLZ         CHECK IF IT IS TIME FOR VEHICLE TO BE WITHIN THE INFLUENCE
                 ZONE OF THE TRAFFIC CONTROL [T/F]
```

5. DEFINITION OF THE VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED

COMMON BLOCKS <APPRO>, <PATH>, <LANE>, <VEHF>, <VEHD>, AND <VEHIL> ARE ENTITIES AND ARE EXPLAINED IN SECTION 4

COMMON / ABIAS / BIASED VEHICLE ATTRIBUTES

BLKDAT OBAP SSOBAP LOGOUT INTERP LOKIOB SSINTR CLRCON LOGIOB IBAP LOKIBI CHKDSP CHKLDT SSIBAP LOGIBI PREST1 PREST2 UNBIAS NEWVEL LCHGEO LCHDES CHKLSI GAPACC CHGMLN ACDCP CARFOL ACCEL CRIDIS HOLDSP PVAPRT INTLOG SIGRES CHKSDR CHKCON SETPTV SETCON CHKMLN BANGS BIAS LOGIN ABORTR

ACCNEW	PRESENT ACCELERATION (FT/SEC/SEC)
ACCOLD	ACCELERATION OLD (AT START OF THIS DT) (FT/SEC/SEC)
DESVEL	DESIRED SPEED (FT/SEC)
ENDLN	POSITION OF THE END OF THE LANE FOR THE VEHICLE (FT)
OLDDTS	OLD DISTANCE TRAVELED FOR STATISTICS (AT START OF THIS DT) (BIASED FEET)
POSNEW	PRESENT POSITION (FROM START OF LINK TO FRONT BUMPER) (FEET)
POSOLD	POSITION OLD (AT START OF THIS DT) (FEET)
PVACC	PREVIOUS VEHICLE ACCELERATION/DECELERATION (FT/SEC/SEC)
PVPOS	PREVIOUS VEHICLE POSITION (FROM START OF LINK TO 4 FEET BEHIND REAR BUMPER), OR IF NO VEHICLE, POSITION OF THE END OF THE LANE (FEET)
PVVEL	PREVIOUS VEHICLE VELOCITY (FT/SEC)
RELEND	RELATIVE DISTANCE BETWEEN THE VEHICLE AND THE END OF HIS LANE (FT) (ENDLN MINUS POS)
RELPOS	RELATIVE DISTANCE BETWEEN VEHICLE AND PREVIOUS VEHICLE (PVPOS MINUS POS) (FEET)
RELVEL	RELATIVE VELOCITY BETWEEN VEHICLE AND PREVIOUS VEHICLE (PVEL MINUS VEL) (FT/SEC)
SLPNEW	PRESENT SLOPE OF ACCEL/DECEL (FT/SEC/SEC/SEC)
SLPOLD	SLOPE OLD OF ACC/DEC (AT START OF THIS DT) (FT/SEC/SEC/SEC)
VELNEW	PRESENT VELOCITY (FT/SEC)
VELOLD	VELOCITY OLD (AT START OF THIS DT) (FT/SEC)

COMMON / ATTB / COLEASE GENERATED DATA TO DESCRIBE THE ATTRIBUTES IN EACH ENTITY

SIMPRO BLKDAT EXTRAC FIND REPACK STORE

IAT(3,310) DESCRIBES THE LOCATION AND SIZE OF THE ATTRIBUTES
- (1,I)=WORD NUMBER FOR EACH ATTRIBUTE (STARTS AT 0)
- (2,I)=STARTING BIT POSITION FOR EACH ATTRIBUTE (0 IS BOTTOM)
- (3,I)=NUMBER OF BITS FOR EACH ATTRIBUTE (AFTER THE DO 1010 LOOP IN SIMPRO IT IS THE MASK FOR EACH ATTRIBUTE POSITIONED PROPERLY)

COMMON / CLASS / DRIVER AND VEHICLE PERFORMANCE VALUES

BLKDAT RDVPRD OBAP LOKIOB CLRCON IBAP LOKIBI CHKDSP CHKLDT PREST1 LCHGEO LCHDES DELAY GAPACC CHGMLN ACDCP CARFOL ACCEL CRIDIS ADLVAI INTLOG SIGRES LSTOP CHKSDR CHKCON PREDTV BIAS LOGIN ABORTR

AMAX(15)	MAXIMUM ACCELERATION RATE FOR EACH VEHICLE CLASS (FT/SEC/SEC)
DCHAR(5)	DRIVER CHARACTERISTIC FOR EACH DRIVER CLASS (AVERAGE DRIVER=1,0, AGGRESSIVE DRIVER=>1,0, SLOW DRIVER=<1,0) (IDCHAR()/100,0)
DCHARM	MAXIMUM DRIVER CHARACTERISTIC FOR ALL DRIVER CLASSES
DMAX(15)	MAXIMUM LINEAR DECELERATION RATE FOR EACH VEHICLE CLASS (FT/SEC/SEC) (=IDMAX()*8/3)
IPIJR(5)	PERCEPTION-REACTION TIME FOR EACH DRIVER CLASS (IN DT/S)
IRMIN(15)	MINIMUM TURNING RADIUS FOR EACH VEHICLE CLASS (FEET)
LENV(15)	LENGTH OF VEHICLES FOR EACH VEHICLE CLASS (FEET)
PIJR(5)	PERCEPTION-REACTION TIME FOR EACH DRIVER CLASS (SECONDS)
VCHAR(15)	VEHICLE CHARACTERISTIC FOR EACH VEHICLE CLASS (AVERAGE VEHICLE=1,0, RESPONSIVE VEHICLE=>1,0, SLUGGISH VEHICLE=<1,0) (IVCHAR()/100,0)
VMAX(15)	MAXIMUM VELOCITY FOR EACH VEHICLE CLASS (FT/SEC)

COMMON / ENTITY / COLEASE GENERATED DATA TO DESCRIBE THE ENTITIES

SIMPRO BLKDAT EXTRAC FIND REPACK STORE LOGIC

IEN(9,8) DATA TO DESCRIBE THE ENTITIES
- (1,I)=NUMBER OF ENTRIES FOR ENTITY I
- (2,I)=NUMBER OF ATTRIBUTES FOR ENTITY I
- (3,I)=NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR AN ENTRY FOR ENTITY I
- (4,I)=LOCATION OF THE FIRST ENTRY IN THE STORAGE STACK FOR ENTITY I
- (5,I)=NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR THE LOGICAL INDEPENDENT ATTRIBUTES FOR ENTITY I
- (6,I)=LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK (RELATIVE TO THE FIRST) FOR THE LOGICAL INDEPENDENT ATTRIBUTES FOR ENTITY I
- (7,I)=NUMBER OF FUNCTION MASKS FOR THE LOGICAL ATTRIBUTES FOR ENTITY I
- (8,I)=LOCATION OF THE FIRST FUNCTION MASK IN THE IFU IN /FUN/ FOR ENTITY I
- (9,I)=LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY ARRAY OF /ATTB/ FOR ENTITY I

COMMON / FUN / COLEASE GENERATED DATA DESCRIBING THE LOGICAL BINARY NETWORK FOR THE ENTITIES

SIMPRO BLKDAT LOGIC

IFU(2,31) DATA TO DESCRIBE THE LOGICAL BINARY NETWORK
- (1,I)=FUNCTION MASK
- (2,I)=STARTING BIT POSITION FOR DEPENDENT ATTRIBUTE (IAT(2,J) FOR DEPENDENT ATTRIBUTE J)

COMMON / INDEX / INDEX NUMBERS FOR CURRENT ENTITIES BEING PROCESSED

BLKDAT RGEOPD OBAP LOGOUT INTERP LOKIOB CLRCON LOGIOB IBAP LOKIBI CHKDSP CHKLDT SSIBAP LOGIBI PREST1 PREST2 NEWVEL DELAY GAPACC CHGMLN ACDCP CARFOL ACCEL CRIDIS ADLVAI PVAPRT INTLOG SIGRES LSTOP CHKSDR CHKCON SETCON UNSETC INFLZN PATHF BANGS LOGIN ABORTR

IA	ENTRY NUMBER FOR APPRO ENTITY OF APPROACH BEING PROCESSED [1,12]
IAN	INDEX NUMBER FOR LIBA/LOBA ARRAYS OF /INTER/ OF APPROACH BEING PROCESSED [1,6]
ICONUP	ENTRY NUMBER FOR CONFLICT ENTITY CURRENTLY EXTRACTED [1,1000]
IL	ENTRY NUMBER FOR LANE ENTITY OF LANE BEING PROCESSED [1,50]
ILN	INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF LANE BEING PROCESSED [1,6]
IP	ENTRY NUMBER FOR PATH ENTITY OF INTERSECTION PATH BEING PROCESSED [1,125]
IPFLAG	DEBUG PRINTING FLAG
IPTHUP	ENTRY NUMBER FOR PATH ENTITY CURRENTLY EXTRACTED [1,125]
IREPFX	FLAG TO INDICATE IF VEHF ATTRIBUTES HAVE BEEN CHANGED IN THIS DT SO THAT THEY MUST BE REPACKED [T/F]
IREPIL	FLAG TO INDICATE IF VEHIL ATTRIBUTES HAVE BEN CHANGED IN THIS DT SO THAT THEY MUST BE REPACKED [T/F]
IV	ENTRY NUMBER FOR VEH ENTITIES OF VEHICLE BEING PROCESSED [1,200]
IVN	INDEX NUMBER OF VEHICLE WITHIN THIS LINK, STARTING WITH THE FIRST VEHICLE IN LINK
IVPV	ENTRY NUMBER FOR THE VEH ENTITIES OF THE PREVIOUS VEHICLE

JPFLAG	DEBUG PRINTING FLAG
JPRTM	PERCEPTION/REACTION TIMER FOR NEXT DT (IN NUMBER OF DT,S) (SEE IPRTM IN VEHD ENTITY)
KPFLAG	DEBUG PRINTING FLAG
LOGTMP	TEMPORARY VARIABLE TO STORE THE VALUE THAT LOGFLG FOR VEHD ENTITY WILL HAVE FOR NEXT DT

COMMON / INTER / DATA ABOUT INTERSECTION
BLKDAT EXEC INITAL RUSERD RGEOPD RCAMSD RPHASD RLOOPD QUEUE OBAP LOGOUT INTERP SSINTR LOGIOB IBAP LOGIBI GAPACC ADLVAI LSTOP CHKSDR CHKCON INFLZN LOGIN INTSTA SUMARY ACTSTA TIMSTA

ICONTR	INTERSECTION TRAFFIC CONTROL INDICATOR: [1,7] 1=UNCONTROLLED 2=YIELD SIGN ON ONE OR MORE APPROACHES 3=STOP SIGN ON LESS THAN ALL APPROACHES 4=STOP SIGN ON ALL APPROACHES 5=PRETIMED SIGNAL 6=SEMI-ACTUATED SIGNAL 7=FULL-ACTUATED SIGNAL
LIBA(6)	LIST OF ENTRY NUMBERS FOR APPRO ENTITY OF INBOUND APPROACHES [1,12]
LIBAR(12)	LIST OF INBOUND APPROACH NUMBERS GIVING ASSOCIATED ENTRY NUMBER (REVERSE OF LIBA) [1,6]
LOBA(6)	LIST OF ENTRY NUMBERS FOR APPRO ENTITY OF OUTBOUND APPROACHES [1,12]
LOBAR(12)	LIST OF OUTBOUND APPROACH NUMBERS GIVING ASSOCIATED ENTRY NUMBER (REVERSE OF LOBA) [1,6]
LVATIN(25)	LIST OF ENTRY NUMBERS FOR VEH ENTITIES OF VEHICLES AT THE INTERSECTION [1,200]
NIBA	NUMBER OF INBOUND APPROACHES [1,6]
NIBL	NUMBER OF INBOUND LANES
NOBA	NUMBER OF OUTBOUND APPROACHES [1,6]
NOCONF	NUMBER OF ENTRIES FOR CONFLT ENTITY [1,2000]
NPATHS	NUMBER OF ENTRIES FOR PATH ENTITY [1,125]
NRLAN	TOTAL NUMBER OF LANES (INBOUND PLUS OUTBOUND)
NUMSDR	NUMBER OF SIGHT DISTANCE RESTRICTIONS
NVATIN	NUMBER OF VEHICLES AT THE INTERSECTION [1,25]
NVIA(12)	NUMBER OF VEHICLES ON EACH APPROACH [0,378]
NVIBA	NUMBER OF VEHICLES ON INBOUND APPROACHES [0,200]
NVIN	NUMBER OF VEHICLES IN THE INTERSECTION [0,200]
NVIP(125)	NUMBER OF VEHICLES IN EACH INTERSECTION PATH [0,15]
NVOBA	NUMBER OF VEHICLES IN OUTBOUND APPROACHES [0,200]
NVSY	NUMBER OF VEHICLES CURRENTLY IN THE SYSTEM [0,200]
TVATIN(25)	TIME INTO THE SIMULATION THAT THE VEHICLE ARRIVED AT THE INTERSECTION (STOPPED AT THE STOP LINE)

COMMON / LANECH / DATA FOR LANE CHANGE PROCESSING
BLKDAT UNBIAS NEWVEL LCHDES SVEHU DELAY GAPACC CHGMLN CARFOL CRIDIS INTLOG LOGIN ABORTR

AVSF	ACCELERATION/DECELERATION OF THE VEHICLE ON THE SIDE OF INTEREST TO THE FRONT
AVSR	ACCELERATION/DECELERATION OF THE VEHICLE ON THE SIDE OF INTEREST TO THE REAR
FACTOR	FACTOR WHICH IS DIVIDED INTO CAR-FOLLOWING EQUATION IN ORDER TO COMPUTE GAP ACCEPTANCE LIMIT FOR LANE CHANGE
ISIDE	LANE CHANGE SIDE INDICATION FLAG: [1,3] 1=WANT TO CHANGE LEFT 2=PRESENT LANE IS ADEQUATE NOW 3=WANT TO CHANGE RIGHT
LAGSPD	SPEED OF VEHICLE TO THE REAR IN ADJACENT DESIRED LANE LAG VEHICLE) (FT/SEC)
LEADSP	SPEED OF VEHICLE TO THE FRONT IN ADJACENT DESIRED LANE (LEAD VEHICLE) (FT/SEC)
NOSF	ENTRY NUMBER FOR VEH ENTITIES OF VEHICLE TO THE FRONT IN ADJACENT DESIRED LANE (LEAD VEHICLE) [1,200]
NOSR	ENTRY NUMBER FOR VEH ENTITIES OF VEHICLE TO THE REAR IN ADJACENT DESIRED LANE (LAG VEHICLE) [1,200]
PVSF	POSITION OF VEHICLE TO THE FRONT IN ADJACENT DESIRED LANE (LEAD VEHICLE) (FEET) [0,1000]
PVSR	POSITION OF VEHICLE TO THE REAR IN ADJACENT DESIRED LANE (LAG VEHICLE) (FEET) [0,1000]
SLPLCH	DESIRED SLOPE OF ACC/DEC FOR LANE CHANGE PROCESSOR (FT/SEC/SEC/SEC)
VVSF	VELOCITY OF THE VEHICLE ON THE SIDE OF INTEREST TO THE FRONT
VVSR	VELOCITY OF THE VEHICLE ON THE SIDE OF INTEREST TO THE REAR

COMMON / LOGICV / VALUES FOR LOGICAL TRUE AND FALSE FOR COLEASE
BLKDAT RDVPRD OBAP LOGOUT FLGNOR INTERP LOKIOB LOGIOB IBAP LOKIBI LOGIBI PRESTI PRESTZ NEWVEL LCHDES GAPACC CHGMLN ACDCP CARFOL ACCEL CRIDIS ADLVAI INTLOG SIGRES LSTOP CHKSDR CHKCON SETCON INFLZN PATHF CHKMLN BANGS BIAS LOGIN ACTSIG

LFALSE	VALUE FOR LOGICAL FALSE FOR COLEASE LOGICAL ATTRIBUTES
LTRUE	VALUE FOR LOGICAL TRUE FOR COLEASE LOGICAL ATTRIBUTES

COMMON / LOOPS / DATA FOR DETECTORS FOR SIGNAL CONTROLLERS
BLKDAT INITAL RPHASD RLOOPD CHKLDT ACTSIG CHKDFP SETLDF

ITYPLD(20)	TYPE OF DETECTOR PULS=PULSE DETECTOR PRES=PRESSURE DETECTOR
LDTRIP(20)	FLAG TO INDICATE IF A VEHICLE TRIPPED EACH DETECTOR SINCE LAST SET FALSE [T/F]
LLOOPS(20)	LIST OF INDEX NUMBERS FOR DETECTORS [1,20]
NLOOPS	NUMBER OF DETECTORS [0,20]
STOPLD(20)	LOCATION ON LANE OF END OF DETECTOR (FEET) [0,1000]
STRTLD(20)	LOCATION ON LANE OF START OF DETECTOR (FEET) [0,1000]

COMMON / NOATTB / NUMBER OF ATTRIBUTES IN EACH COLEASE ENTITY
SIMPRO BLKDAT RGEOPD OBAP INTERP CLRCON IBAP LOGIN ABORTR

NOATTB(8)	NUMBER OF ATTRIBUTES IN EACH OF 8 ENTITIES (SET IN DATA STATEMENT IN BLOCK DATA ROUTINE) (1)=ENTITY APPRO (2)=ENTITY CONFLT (3)=ENTITY LANE (4)=ENTITY PATH (5)=ENTITY SDR (6)=ENTITY VEHD (7)=ENTITY VEHF (8)=ENTITY VEHIL

COMMON / PHASES / DATA FOR SIGNAL CONTROLLER PHASES
BLKDAT INITAL RPHASD RLOOPD ACTSIG CHKDFP SETLDF SUMARY ACTSTA

IANDOR(8)	DETECTOR CONNECTION FOR SIGNAL PHASE AND=SERIES OR=PARALLEL
ICAMPS(8)	STARTING CAM STACK POSITION FOR SIGNAL PHASE
IDUALL(8)	DUAL LEFT OPTION [YES/NO]
IMINOR(8)	PARENT/MINOR OPTION [YES/NO]
IREC(8)	SETTING FOR AUTO-RECALL SWITCH FOR EACH SIGNAL PHASE [ON/OFF]
ISKP(8)	SETTING FOR SKIP-PHASE SWITCH FOR EACH SIGNAL PHASE [ON/OFF]
LLD(10,8)	LIST OF INDEX NUMBERS FOR /LOOPS/ OF DETECTORS CONNECTED TO EACH SIGNAL PHASE (I,PHASE) [1,20]
LPHASE(8)	LIST OF INDEX NUMBERS FOR SIGNAL PHASES
LPHNXT(7,8)	LIST OF INDEX NUMBERS FOR SIGNAL PHASES THAT THE SIGNAL PHASE MAY CLEAR TO
NGAPO(8)	NUMBER OF GAP-OUTS FOR EACH SIGNAL PHASE

```
  NLD(8)          NUMBER OF DETECTORS CONNECTED TO EACH SIGNAL PHASE [0,5]
  NMAXO(8)        NUMBER OF MAX-OUTS FOR EACH SIGNAL PHASE
  NPHASE          NUMBER OF SIGNAL PHASES IN FULL-ACTUATED CONTROLLER [1,8]
  NPHNXT(8)       NUMBER OF SIGNAL PHASES THAT THE SIGNAL PHASE MAY CLEAR TO
  TAR(8)          TIME FOR ALL-RED SIGNAL INTERVAL FOR EACH SIGNAL PHASE
                  (SECONDS)
  TCI(8)          TIME FOR CLEARANCE INTERVAL FOR EACH SIGNAL PHASE (SECONDS)
  TGAPO(8)        SUM OF TIME INTO SIGNAL PHASE FOR GAP-OUTS
  TII(8)          TIME FOR INITIAL INTERVAL FOR EACH SIGNAL PHASE (SECONDS)
  TMAXO(8)        SUM OF TIME INTO SIGNAL PHASE FOR MAX-OUTS
  TMX(8)          TIME FOR MAXIMUM GREEN EXTENSION FOR MAX-OUT AFTER DEMAND
                  ON RED FOR EACH SIGNAL PHASE (SECONDS)
  TVI(8)          TIME FOR VEHICLE INCREMENT FOR EACH SIGNAL PHASE (SECONDS)


COMMON / PRTPVA / DATA FOR PRINTING POSITION/VELOCITY/ACCELERATION PLOTS
                      BLKDAT LOGIOB LOGIBI PVAPRT LOGIN

  DISTAD(200)     SUMMATION OF THE LENGTHS OF THE PREVIOUS LINKS TRAVELED

COMMON / QUE /    DATA FOR VEHICLES WAITING TO BE QUEUED INTO THE SYSTEM
                      BLKDAT EXEC   RDVPRD QUEUE OBAP   LOGOUT INTERP IBAP
                      LOGIBI PVAPRT BANGS  LOGIN  ABORTR

  IBUF(25,8)      HOLDING BUFFER FOR THE NEXT 25 (OR LESS) VEHICLES TO
                  ENTER THE SYSTEM WITH THE FOLLOWING INDEXES :
                      (N,1) = IVEHCL FOR VEHF ENTITY
                      (N,2) = IDRICL FOR VEHF ENTITY
                      (N,3) = ISPD FOR VEHF ENTITY
                      (N,4) = NOBAPD FOR VEHF ENTITY
                      (N,5) = IA FOR /INDEX/ FOR INBOUND APPROACH
                      (N,6) = ILN FOR /INDEX/ FOR LANE ON APPROACH
                      (N,7) = IPRTLO FOR VEHF ENTITY
                      (N,8) = NUMV FOR VEHICLE FOR IQ ARRAY
  IEF             0/1 FOR <MORE>/<NO MORE> VEHICLES ON INPUT FILE WHICH
                  HAVE BEEN READ INTO HOLDING BUFFER IBUF.
  IQ(200)         LIST OF VEHICLE NUMBERS (NUMV) INDEXED BY ENTRY NUMBER
                  FOR VEH ENTITIES (IQ(N)=0 MEANS THAT THE NTH ENTRY OF THE
                  VEH ENTITIES IS UNUSED)
  IQF             COUNTER OF VEHICLES IN QUEUE-IN-BUFFER; WHEN ZERO , THERE
                  ARE NO MORE VEHICLES TO BE LOGGED INTO THE SYSTEM
                  (SEE IBUF AND QTIME IN /QUE/) [0,25]
  LQ(6,6)         LIST OF INDEX NUMBERS FOR IBUF ARRAY FOR THE VEHICLE
                  TO BE QUEUED INTO THE SYSTEM DURING THIS DT FOR THEIR
                  INBOUND APPROACH LANE, INDEXED BY (IAN,ILN) FROM /INDEX/
  NUMV            SEQUENTIAL VEHICLE NUMBER (VALUE FOR NEXT VEHICLE READ
                  IN)
  QTIME(25)       THE QUEUE-IN TIME FOR EACH OF THE 25 (OR LESS) VEHICLES
                  IN IBUF (SECONDS)


COMMON / RUTINE / ROUTINE NAMES CALLED TO PROCESS VEHICLE
                      EXEC   INITAL RUSERD RGEOPD RCAMSD RPHASD RLOOPD RDVPRD
                      QUEUE  OBAP   SSOBAP LOGOUT FLGNOR INTERP LOKIOB SSINTR
                      CLRCON LOGIOB IBAP   LOKIBI CHKDSP CHKLDT SSIBAP LOGIBI
                      PREST1 PREST2 UNBIAS NEWVEL LCHGEO ENDLCH LCHDES CHKLSI
                      SVEHU  DELAY  CKLALT GAPACC CHGMLN ACDCP  CARFOL ACCEL
                      CRIDIS ADLVAI HOLDSP PVAPRT INTLOG SIGRES LSTOP  CHKSDR
                      CHKCON SETPTV PREDTV SETCON UNSETC INFLZN PATHF  CHKMLN
                      BANGS  BIAS   LOGIN  PRESIG ACTSIG CHKDFP SETLDF INTSTA
                      EXTIME ABORTR

  IRNAME(2,36)    ROUTINE NAMES CALLED TO PROCESS VEHICLE
                      (1,I)=FIRST 4 CHARACTERS OF ROUTINE NAME
                      (2,I)=LAST 2 CHARACTERS OF ROUTINE NAME
  MSGR(4)         ERROR MESSAGE IF MORE THAN 35 ROUTINES CALLED FOR A VEHICLE
  NR              NUMBER OF CHARACTERS (13) IN ERROR MESSAGE
  NRNAME          NUMBER OF ROUTINE NAMES ALREADY STORED IN IRNAME ARRAY
  NRNAMM          MAXIMUM NUMBER (35) OF ROUTINES TO BE CALLED BY A VEHICLE
```

```
COMMON / SIGCAM / DATA FOR SIGNAL INDICATIONS FOR LANES
                      BLKDAT INITAL RCAMSD RPHASD OBAP   INTERP IBAP   GAPACC
                      CHGMLN ACDCP  CRIDIS SIGRES INFLZN BANGS  LOGIN  PRESIG
                      ACTSIG SETLDF ABORTR

  IARRPH          ALL-RED REST PHASE NUMBER (0 IF NONE)
  ICAMPC          CURRENT CAM STACK POSITION
  ICAMPH(72)      SIGNAL PHASE FOR CAM STACK POSITION
  ICAMPO          OLD CAM STACK POSITION
  ICPHAS          CURRENT SIGNAL PHASE
  IGO             FLAG INDICATING PROPER RESPONSE IF SIGNAL JUST TURNED
                  AMBER
                      0=SIGNAL IS NOT AMBER
                      1=AMBER-GO
                      2=AMBER-STOP
                      3=FOLLOW AMBER-STOP
  ISISET(72,25)   SIGNAL INDICATION, SUBSCRIPTED BY CAM STACK POSITION
                  AND INBOUND LANE NUMBER (ICAMPC,IBLN) [1,25]
                      1=SIGNAL FOR MOVEMENT IS GREEN AND                AG
                        INTERSECTION CONFLICTS ARE
                        CHECKED FOR U-TURN AND LEFT TURN
                      2=SIGNAL FOR MOVEMENT IS AMBER AND DECISION IS    AA
                        MADE TO GO OR STOP
                      3=SIGNAL FOR MOVEMENT IS RED AND VEHICLE IS       AR
                        STOPPED AT STOP LINE
                      4=SIGNAL FOR MOVEMENT IS PROTECTED GREEN AND      AP
                        INTERSECTION CONFLICTS ARE NOT CHECKED
                      5=LEFT     =GREEN(1)            OTHERS=AMBER(2)  LGA
                      6=LEFT     =GREEN(1)            OTHERS=RED(3)    LGR
                      7=LEFT     =AMBER(2)            OTHERS=GREEN(1)  LAG
                      8=LEFT     =AMBER(2)            OTHERS=RED(3)    LAR
                      9=LEFT     =RED(3)              OTHERS=GREEN(1)  LRG
                     10=LEFT     =RED(3)              OTHERS=AMBER(2)  LRA
                     11=STRAIGHT=GREEN(1)             OTHERS=AMBER(2)  SGA
                     12=STRAIGHT=GREEN(1)             OTHERS=RED(3)    SGR
                     13=STRAIGHT=AMBER(2)             OTHERS=GREEN(1)  SAG
                     14=STRAIGHT=AMBER(2)             OTHERS=RED(3)    SAR
                     15=STRAIGHT=RED(3)               OTHERS=GREEN(1)  SRG
                     16=STRAIGHT=RED(3)               OTHERS=AMBER(2)  SRA
                     17=RIGHT    =GREEN(1)            OTHERS=AMBER(2)  RGA
                     18=RIGHT    =GREEN(1)            OTHERS=RED(3)    RGR
                     19=RIGHT    =AMBER(2)            OTHERS=GREEN(1)  RAG
                     20=RIGHT    =AMBER(2)            OTHERS=RED(3)    RAR
                     21=RIGHT    =RED(3)              OTHERS=GREEN(1)  RRG
                     22=RIGHT    =RED(3)              OTHERS=AMBER(2)  RRA
                     23=LEFT     =PROTECTED GREEN(4)  OTHERS=GREEN(1)  LPG
                     24=LEFT     =PROTECTED GREEN(4)  OTHERS=AMBER(2)  LPA
                     25=LEFT     =PROTECTED GREEN(4)  OTHERS=RED(3)    LPR
  NCAMSP          NUMBER OF CAM STACK POSITIONS
  TCAMSP(72)      TIME INTERVAL FOR CAM STACK POSITION (PRETIMED SIGNAL ONLY)
  TP              TIME INTO SIGNAL PHASE
  TR              TIME REMAINING FOR SIGNAL PHASE INTERVAL


COMMON / STACK /  COLEASE GENERATED STORAGE STACK
                      SIMPRO EXTRAC FIND   REPACK STORE  LOGIC

  IS(5821)        COLEASE STORAGE STACK FOR CDC
  IS(9380)        COLEASE STORAGE STACK FOR IBM


COMMON / SUMSTA / DATA FOR SUMMARY STATISTICS FOR VEHICLES
                  (ARRAYS DIMENSIONED TO (6,3) ARE INDEXED BY (IAN,ITURN))
                      BLKDAT EXEC   INITAL RUSERD SSOBAP LOGOUT SSINTR SSIBAP
                      BANGS  LOGIN  INTSTA SUMARY PSTATS ADDSTA TIMSTA EXTIME
                      ABORTR

  ADESPD(6,3)     SUMMATION FOR AVERAGE DESIRED SPEED (MPH)
  ASPEED(6,3)     SUMMATION FOR TIME MEAN SPEED (MPH)
  DXPH(6,3)       DELAY BELOW XX MPH (VEHICLE-SECONDS)
```

LQUEUE(6,6)	SUMMATION FOR AVERAGE LENGTH OF THE QUEUE (NUMBER OF VEHICLES) INDEXED BY (IAN,ILN)
MNVSY	MAXIMUM NUMBER OF VEHICLES IN THE SYSTEM
MQUEUE(6,6)	MAXIMUM QUEUE LENGTH FOR INBOUND APPROACHES INDEXED BY (IAN,ILN)
NBANG(6)	NUMBER OF COLLISIONS INDEXED BY (IAN)
NDMPH(6,3)	NUMBER OF VEHICLES EXPERIENCING DELAY BELOW XX MPH
NLVDV(6)	NUMBER OF VEHICLES ADDED TO PLVDV ARRAY INDEXED BY (IAN)
NQD(6,3)	NUMBER OF VEHICLES EXPERIENCING QUEUE DELAY
NSD(6,3)	NUMBER OF VEHICLES EXPERIENCING STOPPED DELAY
NTD(6,3)	NUMBER OF VEHICLES EXPERIENCING TOTAL DELAY
NUMPRO(6,3)	NUMBER OF VEHICLES PROCESSED DURING SIMULATION TIME
NUMPSU	NUMBER OF VEHICLES PROCESSED DURING START-UP TIME
NELIM(6)	NUMBER OF VEHICLES ELIMINATED INDEXED BY (IAN)
NVSYA	AVERAGE NUMBER OF VEHICLES IN THE SYSTEM DURING SIMULATION TIME
PLVDV(6)	PERCENT LOGIN VELOCITY TO DESIRED VELOCITY INDEXED BY (IAN)
QD(6,3)	QUEUE DELAY, INCLUDING MOVE UP TIME (VEHICLE-SECONDS)
SD(6,3)	STOPPED DELAY (VEHICLE-SECONDS)
STIME(6,3)	TRAVEL TIME (SECONDS)
TD(6,3)	TOTAL DELAY (VEHICLE-SECONDS)
TMTIME(5)	TOTAL COMPUTER TIME (TM SECONDS)
VMAXA(6,3)	SUMMATION FOR AVERAGE VEHICLE MAXIMUM ACCELERATION (FT/SEC/SEC)
VMAXD(6,3)	SUMMATION FOR AVERAGE VEHICLE MAXIMUM DECELERATION (FT/SEC/SEC)
VMT(6,3)	VEHICLE MILES OF TRAVEL (MILES)
XFPS	XX ASSOCIATED WITH DELAY BELOW XX MPH (FT/SEC)
XQDIST	MAXIMUM RELATIVE POSITION FOR MAINTAINING QUEUE (FEET)

COMMON / TITLE / TITLE FOR RUN

```
BLKDAT EXEC   INITAL RUSERD RGEOPD RCAMSD RPHASD RLUOPD
INTSTA SUMARY ACTSTA TIMSTA
```

ITITLE(20)	80 CHARACTER TITLE FROM SIMPRO INPUT FOR THIS RUN

COMMON / USER / DATA FOR USER DEFINED VALUES

```
BLKDAT EXEC   INITAL RUSERD RGEUPD RCAMSD RPHASD RLOOPD
RDVPRD QUEUE  OBAP   LOGOUT INTERP CLRCON IBAP   CHKDSP
CHKLDT SSIBAP LOGIBI UNBIAS NEWVEL LCHGEO LCHDES SVEHU
GAPACC CHGMLN ACDCP  CARFOL ACCEL  CRIDIS ADLVAI HOLDSP
INTLOG SIGRES LSTOP  CHKSDR CHKCON PREDTV SETCON UNSETC
PATHF  BANGS  LOGIN  PRESIG ACTSIG CHKDFP SETLDF INTSTA
SUMARY PSTATS ACTSTA TIMSTA ABORTR
```

APIJR	AVERAGE PIJR TIME
AUTOL	FACTOR FOR CONVERTING ACCELERATION RATE FROM UNIFORM TO LINEAR
CAREQA	TRADITIONAL CAR FOLLOWING EQUATION ALPHA
CAREQL	TRADITIONAL CAR FOLLOWING EQUATION LAMBDA
CAREQM	TRADITIONAL CAR FOLLOWING EQUATION MU
DT	TIME INCREMENT FOR SIMULATION (SECONDS)
DTCU	DT CUBED
DTSQ	DT SQUARED
DUTOL	FACTOR FOR CONVERTING DECELERATION RATE FROM UNIFORM TO LINEAR
IGEOP	TAPE NUMBER FOR INPUT FILE FROM GEOMETRY PROCESSOR
INPUT	TAPE NUMBER FOR INPUT TO SIMPRO
IPAP	YES/NO FOR SUMMARY STATISTICS PRINTED BY INBOUND APPROACH
IPOLL	YES/NO FOR WRITING POLLUTION TAPE
IPTC	YES/NO FOR SUMMARY STATISTICS PRINTED BY TURN CODE (U-TURN AND LEFT TURN, STRAIGHT, AND RIGHT TURN)
IPUNCH	YES/NO FOR PUNCHING OUTPUT STATISTICS
IVEHP	TAPE NUMBER FOR INPUT FILE FROM DRIVER/VEHICLE PROCESSOR
SIMTIM	TOTAL TIME THAT IS TO BE SIMULATED (SECONDS)
STRTIM	TIME THAT IS TO BE SIMULATED BEFORE STATISTICS ARE GATHERED FROM INDIVIDUAL VEHICLES AT LOGOUT (SECONDS)
TIME	TIME THAT HAS BEEN SIMULATED (COUNTER TO CHECK AGAINST SIMTIM (SECONDS)
TLAG	TIME FOR LAG ZONE FOR INTERSECTION CONFLICT CHECKING (SECONDS)
TLEAD	TIME FOR LEAD ZONE FOR INTERSECTION CONFLICT CHECKING (SECONDS)
TPRINT	TIME INTO THE SIMULATION TO START DEBUG PRINTING
TSTATS	TIME INTERVAL FOR INTERMEDIATE STATISTICS

COMMON / ZTEMPD / TEMPORARY STORAGE OF DATA FOR VARIOUS ROUTINES (THE ONLY ROUTINES THAT USE /ZTEMPD/ FOR TRANSFER OF DATA BETWEEN THEM ARE CHKCON CHKSDR PREDTV SETPTV AND THOSE VARIABLES ARE DEFINED BELOW - ALL OTHER USES ARE LOCAL) (THE MAXIMUM LENGTH OF /ZTEMPD/ IS 110)

```
BLKDAT INITAL RUSERD RGEOPD RCAMSD RPHASD RLOOPD RDVPRD
QUEUE  LOGOUT LOKIOB SSINTR CLRCON LOGIOB LOKIBI CHKDSP
CHKLDT SSIBAP LOGIBI PRESTI LCHGEO LCHDES CHKLSI SVEHU
DELAY  CKLALT GAPACC CHGMLN ACDCP  CARFOL ACCEL  CRIDIS
ADLVAI HOLDSP PVAPRT CHKSDR CHKCON SETPTV PREDTV BANGS
PRESIG ACTSIG CHKDFP SETLDF INTSTA SUMARY PSTATS ADDSTA
ACTSTA TIMSTA ABORTR
```

AO	ACCELERATION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
JD	DRIVER CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSLIM	SPEED LIMIT FOR APPROACH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPD	DESIRED SPEED FOR VEHICLE FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPDP	0/1 FOR NO/YES IF VEHICLE HAS SET DESIRED SPEED FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JV	VEHICLE CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
LGEOM4	LGEOM(4) FOR LANE FOR PREDICTING TIME TO INTERSECTION CONFLICT
MIMP	SPEED LIMIT FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
P	POSITION OF INTERSECTION CONFLICT (LGEOM4+ICOND(J)) FOR PREDICTING TIME TO INTERSECTION CONFLICT
PO	POSITION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
SO	ACC/DEC SLOPE OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
VO	VELOCITY OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT

6. DEFINITION OF THE LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL

VARIABLES THAT ARE LOCAL WITHIN SUBROUTINES ARE LISTED BELOW, EXCEPT FOR MOST DO-LOOP INDICES

SUBROUTINE ABORTR PROCESSES SYSTEM AND USER ERRORS
(CALLED FROM EXEC INITAL RUSERD RGEOPD RCAMSD RPHASD RLOOPD RDVPRD QUEUE SSOBAP LOGOUT FLGNOR LOKIOB SSINTR CLRCON LOGIOB LOKIBI CHKDSP CHKLDT SSIBAP LOGIBI PREST1 PREST2 UNBIAS NEWVEL LCHGEO ENDLCH LCHDES CHKLSI SVEHU DELAY CKLALT GAPACC CHGMLN ACDCP CARFOL ACCEL CRIDIS ADLVAI HOLDSP PVAPRT INTLUG SIGRES LSTOP CHKSDR CHKCON SETPTV PREDTV SETCON UNSETC INFLZN PATHF CHKMLN BANGS BIAS LOGIN PRESIG ACTSIG CHKDFP SETLDF INTSTA EXTIME SHEP)
(CALLS SUMARY XMIT)

COM01	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY APPRO
COM02	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY CONFLT
COM03	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY LANE
COM04	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY PATH
COM05	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY SDR
COM06	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHD
COM07	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
COM08	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK VEHIL
COM09	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK ABIAS
COM10	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK CLASS
COM11	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK INDEX
COM12	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK LANECH
COM13	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK QUE
COM14	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK SIGFAS
COM15	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK SUMSTA
COM16	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN COMMON BLOCK USER
IC(2,19)	COMMON BLOCK NAMES
ICHS	NUMBER OF CHARACTERS TO ENCODE FOR REMARK (CDC ONLY)
IRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
ITIME	DAYFILE MESSAGE FOR TIME IN THE SIMULATION AT ABORT (CDC ONLY)
JRECAD	RECOVERY ADDRESS IF SYSTEM ERROR DETECTED (CDC ONLY)
MSG(NWDS)	ERROR MESSAGE PRINTED
MSGPP(9)	ERROR MESSAGE FOR REMARK (CDC ONLY)
NCHS	NUMBER OF CHARACTERS IN ERROR MESSAGE
NCOM01	VARIABLE NAMES FOR PRINTING ENTITY APPRO
NCOM02	VARIABLE NAMES FOR PRINTING ENTITY CONFLT
NCOM03	VARIABLE NAMES FOR PRINTING ENTITY LANE
NCOM04	VARIABLE NAMES FOR PRINTING ENTITY PATH
NCOM05	VARIABLE NAMES FOR PRINTING ENTITY SDR
NCOM06	VARIABLE NAMES FOR PRINTING ENTITY VEHD
NCOM07	VARIABLE NAMES FOR PRINTING ENTITY VEHF
NCOM08	VARIABLE NAMES FOR PRINTING ENTITY VEHIL
NCOM09	VARIABLE NAMES FOR PRINTING COMMON BLOCK ABIAS
NCOM10	VARIABLE NAMES FOR PRINTING COMMON BLOCK CLASS
NCOM11	VARIABLE NAMES FOR PRINTING COMMON BLOCK INDEX
NCOM12	VARIABLE NAMES FOR PRINTING COMMON BLOCK LANECH
NCOM13	VARIABLE NAMES FOR PRINTING COMMON BLOCK QUE
NCOM14	VARIABLE NAMES FOR PRINTING COMMON BLOCK SIGFAS
NCOM15	VARIABLE NAMES FOR PRINTING COMMON BLOCK SUMSTA
NCOM16	VARIABLE NAMES FOR PRINTING COMMON BLOCK USER
NUM	NUMBER OF ATTRIBUTES FOR ENTITY BEING PRINTED
NWDS	NUMBER OF WORDS FOR ERROR MESSAGE MSG

SUBROUTINE ACCEL ACCELERATES ACCORDING TO THE DESIRED SPEED FOR THIS VEHICLE
(CALLED FROM ACDCP CARFOL CRIDIS)
(CALLS ABORTR)

A	COEFFICIENT FOR T SQUARED FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
ACC	NEW ACCELERATION FOR THE DT
ACCMAX	ACCELERATION MAXIMUM FOR DRIVER
ACCVEH	ACCELERATION MAXIMUM FOR VEHICLE
B	COEFFICIENT FOR T FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
C	CONSTANT FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
CRISLP	CRITICAL SLOPE OF ACC/DEC FOR DRIVER
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
RADICL	RADICAL FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
RELPN	RELATIVE POSITION NEW AFTER DT SECONDS USING ACC/DEC OF ACC
SLOPE	ACC/DEC SLOPE REQUIRED FOR THE DESIRED ACTION
T	TIME REQUIRED FOR THE DESIRED ACTION
VT	VELOCITY OF THE VEHICLE AFTER T SECONDS AND TAKING DESIRED ACTION

SUBROUTINE ACDCP CHECKS THE ACC/DEC LOGICAL DEPENDENT ATTRIBUTES, CALLS THE APPROPRIATE ACC/DEC ROUTINES, AND COMPUTES THE VEHICLES NEW POS/VEL/ACC
(CALLED FROM OBAP INTERP IBAP)
(CALLS ABORTR CARFOL ACCEL CRIDIS NEWVEL ADLVAI HOLDSP)

K	FLAG TO INDICATE IF CRITICAL STOPPING DISTANCE IS VIOLATED: 1=CSD IS VIOLATED, START DECELERATION FOR STOP 2=CSD IS NOT VIOLATED AND WILL NOT BE WITHIN PIJR TIME 3=CSD WILL BE VIOLATED WITHIN PIJR TIME, REDUCE ACCELERATION FOR UPCOMING DECELERATION FOR STOP
MSG906(11)	ERROR MESSAGE
MSG907(11)	ERROR MESSAGE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
RADICL	VALUE FOR SQRT
T	TIME TO BRING VEHICLE TO STOP DURING THIS DT

SUBROUTINE ACTSIG SIMULATES THE SEMI-ACTUATED OR FULL-ACTUATED SIGNAL CONTROLLER
(CALLED FROM EXEC)
(CALLS ABORTR CHKDFP SETLDF)

DTIME	TIME THE SIGNAL CHANGES TO GREEN (FOR MARCH OUT HEADWAYS)
EOM	END OF MAX (SECONDS)
IDFP	T/F FOR DEMAND FOR SIGNAL PHASE
IDOG	T/F FOR DEMAND ON GREEN; TRUE IF RECALL SWITCH IS ON FOR THIS SIGNAL PHASE; TRUE IF A DETECTOR CONNECTED TO THIS SIGNAL PHASE HAS BEEN TRIPPED DURING THIS DT

IDOR T/F FOR DEMAND ON RED; TRUE IF RECALL SWITCH IS ON FOR ANY OTHER SIGNAL PHASE; TRUE IF DEMAND FOR SIGNAL PHASE FOR ANY OTHER SIGNAL PHASE IS TRUE; TRUE IF ANY DETECTOR NOT CONNECTED TO THIS SIGNAL PHASE HAS BEEN TRIPPED DURING THIS DT
INO CHARACTERS (NO)
INTER POSITION OF SIGNAL PHASE THAT SIGNAL IS CURRENTLY IN
1=GREEN
2=AMBER
3=ALL-RED
IOFF CHARACTERS (OFF)
IOPHAS OLD SIGNAL PHASE NUMBER
IPCLTO INDEX NUMBER FOR LPHNXT ARRAY OF THE FIRST SIGNAL PHASE TO CHECK TO SEE IF THIS SIGNAL PHASE SHOULD CLEAR TO IT
IYES CHARACTERS (YES)
MAGSAT T/F FOR MINIMUM ASSURED GREEN SATISFIED WHEN GAP-OUT FROM DUAL-LEFT SIGNAL PHASE
MSG921(13) ERROR MESSAGE
NEXTPH NEXT SIGNAL PHASE FOR THE SIGNAL TO ENTER AFTER AMBER CLEARANCE AND ALL-RED CLEARANCE INTERVALS
NPCLTO NUMBER OF SIGNAL PHASES THAT THIS SIGNAL PHASE MAY CLEAR TO
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME
TBIG VALUE OF A VERY LARGE TIME (SECONDS)
TMAG1 MINIMUM ASSURED GREEN FOR THE FIRST SINGLE LEFT TURN SIGNAL PHASE AFTER THE DUAL LEFT TURN SIGNAL PHASE
TMAG2 MINIMUM ASSURED GREEN FOR THE SECOND SINGLE LEFT TURN SIGNAL PHASE AFTER THE DUAL LEFT TURN SIGNAL PHASE

SUBROUTINE ACTSTA PRINTS THE ACTUATED SIGNAL CONTROLLER STATISTICS AND OPTIONALLY WRITES THE ACTUATED SIGNAL CONTROLLER STATISTICS ELSE PRINTS THE MAIN STREET SEMI-ACTUATED SIGNAL CONTROLLER STATISTICS
(CALLED FROM SUMARY)

ATGAPO AVERAGE TIME INTO SIGNAL PHASE FOR GAP-OUT
ATMAXO AVERAGE TIME INTO SIGNAL PHASE FOR MAX-OUT
IST STARTING SIGNAL PHASE NUMBER
1=FULL ACTUATED SIGNAL CONTROLLER
2=SEMI-ACTUATED SIGNAL CONTROLLER
N NUMBER OF DETECTORS CONNECTED TO SIGNAL PHASE I
NN NUMBER OF SIGNAL PHASES THAT THIS SIGNAL PHASE CLEAR TO
NYES CHARACTERS (YES)

SUBROUTINE ADDSTA ADDS THE SUMMARY STATISTICS FROM (J,K) TO (I,1)
(CALLED FROM SUMARY)

I INBOUND APPROACH NUMBER TO ADD STATISTICS INTO
INDEX SINGLE INTEGER SUBSCRIPT FOR DOUBLE SUBSCRIPTED ARRAYS DIMENSIONED TO (6,3) IN /SUMSTA/ (INDEX) = (I,J)
J INBOUND APPROACH NUMBER FOR ADDING STATISTICS
K TURN CODE NUMBER FOR ADDING STATISTICS

SUBROUTINE ADLVAI ADDS THE STOPPED VEHICLE TO THE LIST OF VEHICLES AT THE INTERSECTION
(CALLED FROM ACDCP)
(CALLS ABORTR ENDLCH PATHF FIND)

IVATIN INDEX NUMBER FOR LVATIN AND TVATIN ARRAYS IN /INTER/ FOR LOCATION OF THIS VEHICLE
J INDEX NUMBER FOR LVATIN AND TVATIN ARRAYS IN /INTER/ FOR MOVING LIST DOWN FROM IVATIN TO END
JSNA ISNA FOR VEHICLE JV
JV ENTRY NUMER FOR VEH ENTITIES OF VEHICLE BEING CHECKED AGAINST
MPRES LPRES FOR VEHICLE JV
MSG908(8) ERROR MESSAGE
MSG909(6) ERROR MESSAGE
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE BANGS PRINTS THE COLLISION INFORMATION AND RESETS THE VEHICLES POS/VEL/ACC
(CALLED FROM OBAP INTERP IBAP)
(CALLS ABORTR FIND)

IAFORM CHARACTERS FOR PRINTING APPROACH HEADER
IDESPD DESIRED SPEED FOR THE VEHICLE FOR THIS DT
IPFORM CHARACTERS FOR PRINTING INTERSECTION PATH HEADER
ISAME FLAG FOR BOTH VEHICLES ON THE SAME LIST (T/F)
ISIG SIGNAL SETTING FOR VEHICLE BEING PRINTED
IWHERE TYPE OF LINK WHICH REAR VEHICLE IN THE COLLISION WAS ON
JA IA FOR FRONT VEHICLE IN THE COLLISION
JBAPS IBAPS FOR FRONT VEHICLE IN THE COLLISION
JDRICL IDRICL FOR FRONT VEHICLE IN THE COLLISION
JL IL FOR FRONT VEHICLE IN THE COLLISION
JLN ILN FOR FRONT VEHICLE IN THE COLLISION
JP IP FOR FRONT VEHICLE IN THE COLLISION
JPOS IPOS FOR FRONT VEHICLE IN THE COLLISION
JSET ISET FOR FRONT VEHICLE IN THE COLLISION
JSIG ISIG FOR FRONT VEHICLE IN THE COLLISION
JSLP ISLP FOR FRONT VEHICLE IN THE COLLISION
JSPD ISPD FOR FRONT VEHICLE IN THE COLLISION
JSTCON ISTCON FOR FRONT VEHICLE IN THE COLLISION
JTURN ITURN FOR FRONT VEHICLE IN THE COLLISION
JVEHCL IVEHCL FOR FRONT VEHICLE IN THE COLLISION
KPRTM IPRTM FOR FRONT VEHICLE IN THE COLLISION
MATPOS LATPOS FOR FRONT VEHICLE IN THE COLLISION
MCHGE LCHGE FOR FRONT VEHICLE IN THE COLLISION
MEGAL LEGAL FOR FRONT VEHICLE IN THE COLLISION
MLANES NLANES FOR APPROACH JA
MNEXT LNEXT FOR FRONT VEHICLE IN THE COLLISION
MOBAPD NOBAPD FOR FRONT VEHICLE IN THE COLLISION
MOF NOF FOR FRONT VEHICLE IN THE COLLISION
MOGFLG LOGFLG FOR FRONT VEHICLE IN THE COLLISION
MOR NOR FOR FRONT VEHICLE IN THE COLLISION
MORC NORC FOR FRONT VEHICLE IN THE COLLISION
MPRES LPRES FOR FRONT VEHICLE IN THE COLLISION
MSG919(10) ERROR MESSAGE
NININT MININT FOR FRONT VEHICLE IN THE COLLISION
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME
POS POSNEW FOR FRONT VEHICLE IN THE COLLISION
POSL LATERAL POSITION OF THE FRONT VEHICLE IN THE COLLISION IN HIS LANE (IF MCHGE=2)
-=LEFT OF CENTER OF LANE
+=RIGHT OF CENTER OF LANE
POSLAT LATERAL POSITION OF THE REAR VEHICLE IN THE COLLISION IN HIS LANE (IF LCHGE=2)
-=LEFT OF CENTER OF LANE
+=RIGHT OF CENTER OF LANE
SLP ACC/DEC SLOPE FOR THE FRONT VEHICLE IN THE COLLISION

SUBROUTINE BIAS BIASES THE VEHICLE ATTRIBUTES, SETS THE PREVIOUS VEHICLE PARAMETERS, AND UPDATES THE MAXIMUM ACC/DEC FOR THE VEHICLE
(CALLED FROM OBAP INTERP IBAP LOGIN)
(CALLS ABORTR)

N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE BLKDAT INITIALIZES DATA IN LABELED COMMON BLOCKS (BLOCK DATA)

SUBROUTINE CARFOL CALCULATES THE ACC/DEC SLOPE REQUIRED TO FOLLOW THE VEHICLE AHEAD
(CALLED FROM ACDCP LOGIN)

(CALLS ABORTR FIND ACCEL)

A	COEFFICIENT FOR T SQUARED FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
ACC	ACCELERATION TO BRING VEHICLE BACK UP TO SPEED
ACCMAX	MAXIMUM ACCELERATION FOR DRIVER
B	COEFFICIENT FOR T FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
C	CONSTANT FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
CARDEC	ACC/DEC VALUE AS DEFINED BY TRADITIONAL CAR FOLLOWING EQUATION
CARDIS	DESIRED CAR FOLLOWING DISTANCE
CRISLP	CRITICAL SLOPE OF ACC/DEC FOR DRIVER
DECVEH	MAXIMUM DECELERATION FOR VEHICLE
DIST	DISTANCE TRAVELED DURING T SECONDS
FACT	(3010-4010) FACTOR FOR MULTIPLYING RELPOS TO TRANSITION A LANE CHANGING VEHICLE INTO CAR FOLLOWING
FACT	(4020-7010) FACTOR FOR MULTIPLYING DCHAR FOR CALLING ACCEL
LATNOW	BIASED LATERAL POSITION FOR A LANE CHANGE (POSITION NOW)
LAT2GO	BIASED TOTAL LATERAL POSITION FOR A LANE CHANGE (TO GO)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
RADICL	RADICAL FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
SLOPE	SLOPE OF ACC/DEC FOR DECELERATING TO DESIRED SPEED
SLOPEU	ACC/DEC SLOPE REQUIRED TO REDUCE THE VEHICLES ACC/DEC TO ZERO BY THE TIME THE VEHICLE REACHES HIS DESIRED SPEED
SPD	MAXIMUM OF DESIRED SPEED AND PREVIOUS VEHICLE VELOCITY
T	(5010-5020) TIME TO BRING DECELERATION UP TO ACCNEW
T	(8010-8020) TIME TO REDUCE DECELERATION TO 0 AT MINUS FIVE-SIXTHS CRISLP
T1	TIME TO BRING DECELERATION UP TO ZERO AT A SLOPE OF ONE-HALF CRISLP
VT1	VELOCITY AFTER T1 SECONDS

SUBROUTINE CHGMLN LOGS THE VEHICLE OUT OF HIS PRESENT LANE AND INTO THE NEW LANE
(CALLED FROM LCHDES)
(CALLS ABORTR STORE FIND FLGNOR UNSETC PATHF INFLZN)

DECMAX	MAXIMUM DECELERATION DRIVER-VEHICLE UNIT WILL USE TO DECELERATE TO A STOP
F3	VALUE FOR MINUS FOUR THIRDS
IENT6	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VEHD LOGICAL DEPENDENT ATTRIBUTES
IENT7	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
JBLN	SAVED IBLN FOR CALLING INFLZN
JGO	SAVED IGO FOR CALLING INFLZN
JLN	INDEX NUMBER FOR LLANES ARRAY FOR APPRO ENTITY OF LANE BEING CHANGED INTO
JSET	ISET FOR NEW NOR VEHICLE
JVEL	IVEL FOR NEW NOF VEHICLE
LGEOM2	LGEOM(2) FOR NEW LANE
LGEOM4	LGEOM(4) FOR NEW LANE
LTF	T/F FOR MFINL AND MOASF FOR OLD NOR VEHICLE
MCONTR	SAVED LCONTR FOR CALLING INFLZN
MEGAL	LEGAL FOR NEW NOR VEHICLE
MWID	LWID FOR NEW LANE
NOASF	NEW MOASF FOR OLD NOR VEHICLE
NVILL	NUMBER OF VEHICLES IN LANE FOR NEXT DT
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
POSLAT	LATERAL POSITION IN LANE FOR LANE CHANGE (TOTAL DISTANCE TO CHANGE) -=LEFT OF CENTER OF NEW LANE +=RIGHT OF CENTER OF NEW LANE
XCRIT	CRITICAL STOPPING DISTANCE

SUBROUTINE CHKCON CHECKS INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED INTO THE INTERSECTION
(CALLED FROM CHKSDR)
(CALLS ABORTR EXTRAC SETPTV PREDTV FIND STORE SETCON)

ACH	ACC/DEC AT THE INTERSECTION CONFLICT FOR HIM
ACM	ACC/DEC AT THE INTERSECTION CONFLICT FOR ME
AO	ACCELERATION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
DCH	DISTANCE TO THE INTERSECTION CONFLICT FOR HIM
DCM	DISTANCE TO THE INTERSECTION CONFLICT FOR ME
DVH	DESIRED VELOCITY ON THE INTERSECTION PATH FOR HIM
DVM	DESIRED VELOCITY ON THE INTERSECTION PATH FOR ME
ERRJUD	ERROR IN JUDGEMENT IN PREDICTING TCH
IVCONF	ENTRY NUMBER FOR VEH ENTITIES OF HIM VEHICLE
J	INDEX NUMBER FOR CONFLT ENTITY ARRAYS FOR OTHER INTERSECTION PATH INVOLVED IN INTERSECTION CONFLICT
JACC	IACC FOR VEHICLE IVCONF
JD	DRIVER CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
JFVA	IFVA FOR VEHICLE IVCONF
JL	ENTRY NUMBER FOR LANE ENTITY OF LINKING INBOUND LANE FOR INTERSECTION PATH JP
JNDEX	ENTRY NUMBER FOR CONFLT ENTITY OF INTERSECTION CONFLICT BEING CHECKED
JP	ENTRY NUMBER FOR PATH ENTITY OF OTHER INTERSECTION PATH AT INTERSECTION CONFLICT
JPOS	IPOS FOR VEHICLE IVCONF
JSLIM	SPEED LIMIT FOR APPROACH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSLP	ISLP FOR VEHICLE IVCONF
JSNA	ISNA FOR VEHICLE IVCONF
JSPD	DESIRED SPEED FOR VEHICLE FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPDP	0/1 FOR NO/YES IF VEHICLE HAS SET DESIRED SPEED FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JV	VEHICLE CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
JVEL	IVEL FOR VEHICLE IVCONF
KOUNT	COUNT FOR NUMBER OF TIMES GOING THROUGH 1090 TO 1100 CODE
KPRTM	IPRTM FOR VEHICLE IVCONF
KSPD	ISPD FOR VEHICLE IVPV
LGEOM4	LGEOM(4) FOR LANE FOR PREDICTING TIME TO INTERSECTION CONFLICT
MGEOM4	LGEOM(4) FOR LANE JL
MIMP	SPEED LIMIT FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
MOR	NOR FOR VEHICLE NOFC
MORC	NORC FOR VEHICLE NOFC
MSG913(6)	ERROR MESSAGE
NININT	MININT FOR VEHICLE NOFC
NOFC	ENTRY NUMBER FOR VEH ENTITIES OF THE VEHICLES BETWEEN THE INTERSECTION CONFLICT BEING CHECKED AND THEIR VEHICLE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
P	POSITION OF INTERSECTION CONFLICT (LGEOM4+ICOND(J)) FOR PREDICTING TIME TO INTERSECTION CONFLICT
PO	POSITION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
RADIAN	VALUE FOR ONE RADIAN
SLOPE	75 PERCENT OF THE NORMAL CRITICAL SLOPE FOR THE DRIVER
SO	ACC/DEC SLOPE OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
TCH	TIME TO INTERSECTION CONFLICT FOR HIM
TCM	TIME TO INTERSECTION CONFLICT FOR ME
TCRASH	TIME REQUIRED TO REDUCE THE RELATIVE SPEEDS SO THERE WOULD BE NO COLLISION
TFZ	TIME FOR FRONT OF ZONE TO REACH INTERSECTION CONFLICT

TIM	MAXIMUM TIME FROM THE END OF THE LANE THAT THIS VEHICLE MAY DEDICATE HIMSELF TO AN INTERSECTION PATH WITH NO INTERSECTION CONFLICTS BEING MET
TPASSH	TIME FOR HIS VEHICLE TO PASS INTERSECTION CONFLICT
TPASSM	TIME FOR MY VEHICLE TO PASS INTERSECTION CONFLICT
TRZ	TIME FOR REAR OF ZONE TO REACH INTERSECTION CONFLICT
VCH	VELOCITY AT INTERSECTION CONFLICT FOR HIM
VCM	VELOCITY AT INTERSECTION CONFLICT FOR ME
VO	VELOCITY OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT

SUBROUTINE CHKDFP CHECKS THE DEMAND FOR THE IP SIGNAL PHASE (WHEN ITYPE IS EQ 1 THEN ONLY THE POSITIVE DETECTOR CONNECTIONS ARE CHECKED AND WHEN ITYPE EQ 2 THEN BOTH THE POSITIVE AND NEGATIVE CONNECTIONS ARE CHECKED)
(CALLED FROM ACTSIG)
(CALLS ABORTR)

IDFP	T/F FOR DEMAND FOR SIGNAL PHASE IP
ION	CHARACTERS (ON)
IP	INDEX NUMBER FOR /PHASES/ OF SIGNAL PHASE BEING CHECKED
ITYPE	FLAG FOR CHECKING NEGATIVE DETECTOR CONNECTIONS 1=POSITIVE DETECTOR CONNECTIONS ONLY 2=NEGATIVE AND POSITIVE DETECTOR CONNECTIONS
JAND	CHARACTERS (AND)
JLD	INDEX NUMBER FOR /LOOPS/ FOR DETECTOR BEING CHECKED
NUMLD	NUMBER OF DETECTORS CONNECTED TO SIGNAL PHASE IP
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE CHKDSP CHECKS TO SEE IF THE VEHICLE SHOULD RESET HIS DESIRED SPEED TO THE DESIRED SPEED OF HIS INTERSECTION PATH SO THAT HE CAN GRADUALLY DECELERATE TO HIS NEW DESIRED SPEED BEFORE HE ENTERS THE INTERSECTION
(CALLED FROM IBAP)
(CALLS ABORTR FIND STORE)

MIMP	LIMP FOR LINKING INTERSECTION PATH FOR VEHICLE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
SLOPE	ACC/DEC SLOPE REQUIRED FOR VELOCITY CHANGE TO SPD
SPD	DESIRED SPEED FOR THE INTERSECTION PATH (FT/SEC)
T	TIME REQUIRED FOR VELOCITY CHANGE TO SPD
XCRIT	DISTANCE REQUIRED FOR VELOCITY CHANGE TO SPD (MINIMUM OF 4*SPD)

SUBROUTINE CHKLDT CHECKS EACH DETECTOR FOR THIS LANE TO SEE IF THIS VEHICLE TRIPPED ANY OF THEM THIS DT
(CALLED FROM IBAP)
(CALLS ABORTR)

IPULS	CHARACTERS (PULS)
JLDL	INDEX NUMBER FOR /LOOPS/ FOR DETECTOR
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
POSNRB	NEW POSITION OF REAR BUMPER FOR VEHICLE
POSORB	OLD POSITION OF REAR BUMPER FOR VEHICLE
STOP	STOP POSITION FOR DETECTOR BEING CHECKED
STRT	START POSITION FOR DETECTOR BEING CHECKED

SUBROUTINE CHKLSI CHECKS THE LANE ON THE SIDE OF INTEREST TO SEE IF THE LANE IS AVAILABLE AT THE CURRENT POSITION OF THE VEHICLE
(CALLED FROM LCHGES DELAY)
(CALLS ABORTR FIND)

LANSI	ENTRY NUMBER FOR LANE ENTITY OF LANE TO BE CHECKED ON THE SIDE OF INTEREST
LB	LANE BEGINNING
LE	LANE ENDING
LGEOM1	LGEOM(1) FOR LANE ON SIDE OF INTEREST
LGEOM2	LGEOM(2) FOR LANE ON SIDE OF INTEREST
LGEOM3	LGEOM(3) FOR LANE ON SIDE OF INTEREST
LGEOM4	LGEOM(4) FOR LANE ON SIDE OF INTEREST
LOK	FLAG INDICATING WHETHER OR NOT AN ADJACENT LANE IS AVAILABLE AT THIS POINT (AT POSNEW) 0=LANE IS AVAILABLE AND NOT BLOCKED 1=LANE IS NOT AVAILABLE AT POSNEW 2=VEHICLE PAST END OF LANE AT POSNEW
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE CHKMLN CHECKS MY LANE AND IF BLOCKED THEN SETS PARAMETERS FOR BLOCKED LANE
(CALLED FROM LOGIOB PATHF LOGIN)
(CALLS ABORTR)

MSG917(10)	ERROR MESSAGE
MSG918(12)	ERROR MESSAGE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE CHKSDR CHECKS SIGHT DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECKS INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED INTO THE INTERSECTION
(CALLED FROM INTLOG LSTOP)
(CALLS ABORTR EXTRAC FIND SETPTV PREDTV CHKCON)

ACM	ACC/DEC AT THE INTERSECTION CONFLICT FOR ME
DCH	DISTANCE TO THE INTERSECTION CONFLICT FOR HIM
DCM	DISTANCE TO THE INTERSECTION CONFLICT FOR ME
DVM	DESIRED VELOCITY ON THE INTERSECTION PATH FOR ME
ERRJUD	ERROR IN JUDGMENT IN PREDICTING TCH
INDEX	INDEX NUMBER FOR IGEOCP ARRAY OF PATH ENTITY FROM LAST TO FIRST
IPNDEX	INDEX NUMBER OF ICANSE ARRAY OF SDR ENTITY BASED ON THE NEW POSITION OF THIS VEHICLE
J	INDEX NUMBER FOR ARRAYS IN CONFLT ENTITY FOR OTHER INTERSECTION PATH INVOLVED IN THE INTERSECTION CONFLICT
JA	ENTRY NUMBER FOR APPRO ENTITY OF INBOUND APPROACH FOR OTHER INTERSECTION PATH INVOLVED IN THE INTERSECTION CONFLICT
JCANSE	THE DISTANCE DOWN INBOUND APPROACH JA THAT CAN FIRST BE SEEN BY THIS VEHICLE
JL	LIBL FOR INTERSECTION PATH JP
JNDEX	INDEX NUMBER FOR CONFLT ENTITY OF INTERSECTION CONFLICT BEING CHECKED
JP	ENTRY NUMBER FOR PATH ENTITY OF OTHER INTERSECTION PATH INVOLVED IN THE INTERSECTION CONFLICT
JSDRA	T/F FOR INBOUND APPROACH CHECKED FOR SIGHT DISTANCE RESTRICTION (PARALLELS ARRAY ISDRA OF APPRO ENTITY)
JSLIM	SPEED LIMIT FOR APPROACH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPD	DESIRED SPEED FOR VEHICLE FOR PREDICTING TIME TO INTERSECTION CONFLICT
JVEL	IVEL FOR THE LAST VEHICLE ON THIS VEHICLE#S INTERSECTION PATH
KCANSE	THE DISTANCE DOWN THE INBOUND APPROACH JA THAT CAN FIRST BE SEEN BY THIS VEHICLE AT POSCHK
KSPD	DESIRED SPEED OF THE PREVIOUS VEHICLE
LGEOM4	LGEOM(4) FOR LANE FOR PREDICTING TIME TO INTERSECTION CONFLICT
MAXLOG	MAXIMUM LOGFLG/LOGTMP
MSDR	NUMBER OF INBOUND APPROACHES CHECKED THAT HAVE A SIGHT DISTANCE RESTRICTION
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

P — POSITION OF INTERSECTION CONFLICT (LGEOM4+ICOND(J)) FOR PREDICTING TIME TO INTERSECTION CONFLICT
PO — POSITION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
POSCHK — PREDICTED POSITION IN THE FUTURE BASED ON CONSTANT SPEED
TCH — TIME TO THE INTERSECTION CONFLICT FOR HIM
TCM — TIME TO THE INTERSECTION CONFLICT FOR ME
TFZ — TIME FOR FRONT ZONE OF OTHER FAKE VEHICLE TO REACH THE INTERSECTION CONFLICT
TIM — MAXIMUM TIME FROM THE END OF THE LANE THAT THIS VEHICLE MAY DECIDE TO PROCEED IF SIGHT DISTANCE RESTRICTIONS ARE CLEAR
TIMEND — MAXIMUM TIME FROM THE INTERSECTION CONFLICT
TPASSM — TIME REQUIRED FOR MY VEHICLE TO PASS THE INTERSECTION CONFLICT AT THE VELOCITY AT THE INTERSECTION CONFLICT
VCM — VELOCITY AT THE INTERSECTION CONFLICT FOR ME
VO — VELOCITY OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT

SUBROUTINE CKLALT CHECKS THE LANE ALTERNATIVES FOR THIS LANE
(CALLED FROM DELAY)
(CALLS ABORTR FIND)

IPATH — ENTRY NUMBER FOR PATH ENTITY OF INTERSECTION PATH BEING CHECKED FOR LANE ALTERNATIVES
JLCH — ILCH FOR INTERSECTION PATH BEING CHECKED
MOBAP — LOBAP FOR INTERSECTION PATH IPATH
MPINT — NPINT FOR LANE BEING CHECKED
N1 — FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 — LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE CLRCON CLEARS THE INTERSECTION CONFLICTS AS THE REAR BUMBER PASSES THEM
(CALLED FROM INTERP)
(CALLS ABORTR EXTRAC STORE FIND)

IENT2 — SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY CONFLT
IPOSRR — POSITION OF REAR BUMPER FOR CLEARING INTERSECTION CONFLICTS
JCONI — ICONI FOR OTHER INTERSECTION PATH INVOLVED IN INTERSECTION CONFLICT
JGEOCP — IGEOCP FOR INTERSECTION CONFLICT IK
JP — ENTRY NUMBER FOR PATH ENTITY FOR OTHER INTERSECTION PATH INVOLVED IN INTERSECTION CONFLICT
MCPSET — NCPSET FOR OTHER INTERSECTION PATH INVOLVED IN INTERSECTION CONFLICT
NUM — NUMBER OF ATTRIBUTES IN ENTITY
N1 — FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 — LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE CRIDIS CHECKS CRITICAL STOPPING DISTANCE FOR A DECELERATION TO A STOP AND IF VIOLATED THEN INITIATES A DECELERATION TO A STOP
(CALLED FROM ACDCP)
(CALLS ABORTR NEWVEL HOLDSP ACCEL)

CRISLP — CRITICAL ACC/DEC SLOPE FOR DRIVER
DECMAX — MAXIMUM DECELERATION FOR DRIVER FOR NORMAL DECELERATION TO STOP
DENOM — 6 TIMES REMAINING DISTANCE TO NEAREST OBJECT FORWARD
F3 — VALUE FOR MINUS FOUR-THIRDS
K — FLAG TO INDICATE IF CRITICAL STOPPING DISTANCE IS VIOLATED:
  1=CSD IS VIOLATED, START DECELERATION FOR STOP
  2=CSD IS NOT VIOLATED AND WILL NOT BE WITHIN PIJR TIME
  3=CSD WILL BE VIOLATED WITHIN PIJR TIME, REDUCE ACCELERATION FOR UP COMING DECELERATION FOR STOP
N1 — FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 — LAST 2 CHARACTERS OF THE ROUTINE NAME

OLDACC — OLD ACC/DEC FOR DECELERATION TO STOP
RADICL — VALUE FOR SQRT
REACTT — PERCEPTION/REACTION TIME FOR DECELERATION TO STOP
RELNEW — (1030*2010) RELATIVE POSITION NEW AFTER T SECONDS
(3010*3020) RELATIVE POSITION NEW AFTER REACTT SECONDS
RELOLD — RELATIVE POSITION USING OLD POSITION
T — (1030*2010) TIME TO REDUCE ACCELERATION TO 0.01
T — (2010*3010) TIME INTO FUTURE FOR REDUCING ACCELERATION TO 0.01
V — VELOCITY AT END OF T SECONDS
VSQT4 — VELOLD SQUARED TIMES 4
VT2 — VELOLD TIMES 2
X — CHANGE IN POSITION AT END OF T SECONDS
XCRIT — CRITICAL STOPPING DISTANCE FOR DECELERATION TO STOP

SUBROUTINE DELAY FINDS THE LEGAL LANE FOR THE VEHICLE WITH THE MINIMUM EXPECTED DELAY
(CALLED FROM LCHDES)
(CALLS ABORTR FIND CKLALT CHKLSI SVEHU)

IPENTC(3,3) — PENALTIES TO BE ADDED TO THE NUMBER OF VEHICLES IN LANE, INDEXED BY (ITURN FOR ME, ITURN FOR VEHICLE AHEAD)
JLCH — ILCH FOR LINKING INTERSECTION PATH FOR VEHICLE
JTURN — ITURN FOR NOF/NOSF/NOSR VEHICLE
LAGR — LAGSPD FOR RIGHT LANE (SEE /LANECH/)
LANSI — ENTRY NUMBER FOR LANE ENTITY OF LANE TO BE CHECKED ON THE SIDE OF INTEREST
LEADR — LEADSP FOR RIGHT LANE (SEE /LANECH/)
LOK — FLAG INDICATING WHETHER OR NOT AN ADJACENT LANE IS AVAILABLE AT THIS POINT (AT POSNEW)
  0=LANE IS AVAILABLE AND NOT BLOCKED
  1=LANE IS NOT AVAILABLE AT POSNEW
  2=VEHICLE PAST END OF LANE AT POSNEW
NOQ — NUMBER OF VEHICLES AHEAD OF PRESENT VEHICLE IN ADJACENT LANE
NORF — NOSF FOR RIGHT LANE (SEE /LANECH/)
NORR — NOSR FOR RIGHT LANE (SEE /LANECH/)
N1 — FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 — LAST 2 CHARACTERS OF THE ROUTINE NAME
PVRF — PVSF FOR RIGHT LANE (SEE /LANECH/)
PVRR — PVSR FOR RIGHT LANE (SEE /LANECH/)
QUEL — EQUIVALENT NUMBER OF VEHICLES AHEAD OF VEHICLE IN LEFT LANE
QUER — EQUIVALENT NUMBER OF VEHICLES AHEAD OF VEHICLE IN RIGHT LANE
QUES — EQUIVALENT NUMBER OF VEHICLES AHEAD OF VEHICLE IN SAME LANE

SUBROUTINE ENDLCH ENDS THE LANE CHANGE AND RESETS THE LANE CHANGE FLAGS
(CALLED FROM LOGIBI LCHGEO ADLVAI)
(CALLS ABORTR FIND STORE)

MCHGE — LCHGE FOR NOR VEHICLE
N1 — FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 — LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE EXEC IS THE MAIN DRIVER FOR SIMPRO AND CONTROLS THE CALLING OF THE VARIOUS OTHER ROUTINES
(CALLED FROM SIMPRO)
(CALLS EXTIME ISLCPF ABORT INITAL XMIT QUEUE OBAP INTERP IBAP PRESIG ACTSIG INTSTA SUMARY ABORTR)

IBUF1 — BUFFER FOR TAPE1 FOR POSITION VS TIME PLOT (CDC ONLY)
IBUF2 — BUFFER FOR TAPE2 FOR VELOCITY VS TIME PLOT (CDC ONLY)
IBUF3 — BUFFER FOR TAPE3 FOR ACCELERATION VS TIME PLOT (CDC ONLY)
IBUF4 — BUFFER FOR TAPE4 FOR PAGE PLOT OF POSITION (CDC ONLY)
IFET1 — FILE ENVIRONMENT TABLE FOR TAPE1 FOR POSITION VS TIME PLOT (CDC ONLY)

```
IFET2          FILE ENVIRONMENT TABLE FOR TAPE2 FOR VELOCITY VS TIME
               PLOT (CDC ONLY)
IFET3          FILE ENVIRONMENT TABLE FOR TAPE3 FOR ACCELERATION VS TIME
               PLOT (CDC ONLY)
IFET4          FILE ENVIRONMENT TABLE FOR TAPE4 FOR PAGE PLOT OF POSITION
               (CDC ONLY)
IPAGE          CONTROLS CARRIAGE CONTROL
                  1=SKIP TO TOP OF NEW PAGE
                  2=SKIP TO BOTTOM OF CURRENT PAGE
IRET           RETURN FLAG FROM ISLCPF (CDC ONLY)
                  0=OK
                  1=FILE ALREADY ASSIGNED
ITIM           NUMBER OF DTS BETWEEN INTERMEDIATE STATISTICS
ITNOW          NUMBER OF DTS INTO SIMULATION TIME (FOR INTERMEDIATE
               STATISTICS)
MSG            ERROR MESSAGE FOR FATAL EXECUTION ERROR (CDC ONLY)
MSG1           ERROR MESSAGE FOR SETTING UP TAPE1 FOR POSITION
               VS TIME PLOT (CDC ONLY)
MSG2           ERROR MESSAGE FOR SETTING UP TAPE2 FOR VELOCITY
               VS TIME PLOT (CDC ONLY)
MSG3           ERROR MESSAGE FOR SETTING UP TAPE3 FOR ACCELERATION
               VS TIME PLOT (CDC ONLY)
MSG4           ERROR MESSAGE FOR SETTING UP TAPE4 FOR PAGE PLOT
               OF POSITION (CDC ONLY)
NRECAD         FATAL EXECUTION ERROR RECOVERY ADDRESS (CDC ONLY)
N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2             LAST  2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE EXTIME GETS THE TM TIME FOR THIS JOB
               (CALLED FROM EXEC   INTSTA SUMARY TIMSTA)
               (CALLS ABORTR)

I              INDEX NUMBER FOR TMTIME ARRAYS IN /SUMSTA/
                  1=START OF JOB
                  2=END OF INITIALIZATION
                  3=END OF START-UP TIME
                  4=END OF SIMULATION
                  5=END OF SUMMARY STATISTICS
ITM            TM TIME USED SO FAR (MILLI-SECONDS)
N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2             LAST  2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE EXTRAC EXTRACTS THE ATTRIBUTES FOR ENTRY IN OF ENTITY IY FROM THE
               STORAGE STACK AND PUTS THEM IN THE COMMON BLOCK FOR ENTITY IY
               (CALLED FROM OBAP   INTERP CLRCON LOGIOB IBAP   LOGIBI PREST1
                        PREST2 LSTOP  CHKSDR CHKCON SETCON UNSETC PATHF)
               (CALLS LSHIFT IAND   SMEP)

IBA            LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
               FOR ENTITY IY
ID             SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES
               IN ALL THE ENTITIES
IEA            LOCATION OF THE LAST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
               FOR ENTITY IY
IFW            LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
               ENTRY IN OF ENTITY IY
IIAT           SINGLE INDEX FOR IAT ARRAY OF /ATTB/
IIEN           SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
IN             ENTRY NUMBER FOR ENTITY IY
ISNAME(2)      SUBROUTINE NAME FOR PRINTING (EXTRAC)
IWD            LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
               ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
               OF ENTITY IY
IY             ENTITY NUMBER
                  1=APPRO
                  2=CONFLT
                  3=LANE
                  4=PATH
                  5=SDR
                  6=VEHD
                  7=VEHF
                  8=VEHIL
NBITS          NUMBER OF BITS PER COMPUTER WORD
NWE            NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE FIND   FINDS THE VALUE OF ATTRIBUTE IV OF ENTRY IN OF ENTITY IY IN
               THE STORAGE STACK AND PUTS IT INTO LOCAL INTEGER IR
               (CALLED FROM RCAMSD RLOOPD FLGNOR LOKIOB SSINTR CLRCON LOGIOB
                        IBAP   LOKIBI CHKDSP SSIBAP LOGIBI PREST1 PREST2
                        ENDLCH LCHDES CHKLSI SVEHU  DELAY  CKLALT GAPACC
                        CHGMLN CARFOL ADLVAI INTLOG SIGRES LSTOP  CHKSDR
                        CHKCON SETCON UNSETC INFLZN PATHF  BANGS)
               (CALLS LSHIFT IAND   SMEP)

I              ABSOLUTE ATTRIBUTE NUMBER
IBA            LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
               FOR ENTITY IY
IE             SMEP ERROR NUMBER
IFW            LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
               ENTRY IN OF ENTITY IY
IIAT           SINGLE INDEX FOR IAT ARRAY OF /ATTB/
IIEN           SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
IN             ENTRY NUMBER FOR ENTITY IY
IR             LOCAL INTEGER TO BE SET TO THE VALUE OF ATTRIBUTE IV OF
               ENTRY IN OF ENTITY IY
ISNAME(2)      SUBROUTINE NAME FOR PRINTING (FIND)
IV             ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)
IWD            LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
               ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
               OF ENTITY IY
IY             ENTITY NUMBER
                  1=APPRO
                  2=CONFLT
                  3=LANE
                  4=PATH
                  5=SDR
                  6=VEHD
                  7=VEHF
                  8=VEHIL
NWE            NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE FLGNOR SETS MFINL AND MOASF TO LTF, RESETS IACC TO SLIGHTLY
               DECELERATING IF MSFLG EQ LTRUE AND THE VEHICLE IS NOT
               DECELERATING, SETS MSFLG TO LFALSE, AND FINALLY STORES NEWNOF
               FOR NOF FOR THE NOR VEHICLE
               (CALLED FROM LOGOUT LOGIOB LOGIBI CHGMLN)
               (CALLS ABORTR STORE  FIND)

JACC           IACC OF THE NOR VEHICLE
LTF            LTRUE OR LFALSE: AM I THE NEW FIRST VEHICLE IN
               INTERSECTION PATH
NEWNOF         NEW NOF OF THE NOR VEHICLE
NSFLG          MSFLG OF THE NOR VEHICLE
N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2             LAST  2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE GAPACC CHECKS IF THERE IS AN ACCEPTABLE GAP TO LANE CHANGE INTO AND
               IF NOT THEN DETERMINES THE APPROPRIATE DRIVER RESPONSE
               FOR LANE CHANGING
               (CALLED FROM LCHDES)
               (CALLS ABORTR FIND   STORE)

ACCVEH         75 PERCENT OF THE MAXIMUM ACCELERATION OF THE VEHICLE
               BASED UPON CURRENT SPEED AND VEHICLE CHARACTERISTICS
ALAGAP         ACCEPTABLE LAG GAP
ALEGAP         ACCEPTABLE LEAD GAP
CRISLP         CRITICAL ACC/DEC SLOPE FOR DRIVER
```

DECMAX	MAXIMUM DECELERATION FOR DECELERATION TO A STOP
DENOM	SIX TIMES RELDIS
FACT	FACTOR USED IN COMPUTING ACCEPTABLE GAPS FOR LANE CHANGE
GAPLA	ACTUAL LAG GAP
GAPLE	ACTUAL LEAD GAP
GAPMIN	MINIMUM VALUE FOR ACCEPTABLE GAP
JACC	IACC FOR NOSF/NOSR VEHICLE
JBLN	IBLN FOR LANE LANSI
JSET	ISET FOR NOSR VEHICLE
JSISET	SIGNAL SETTING FOR LANE LANSI AND CURRENT CAM STACK POSITION
JVEHCL	IVEHCL FOR NOSF/NOSR VEHICLE
LANSI	ENTRY NUMBER OF LANE ENTITY OF LANE TO BE CHECKED ON THE SIDE OF INTEREST
LEGAP	T/F FOR ACCEPTABILITY OF LEAD GAP
MCONTR	LCONTR FOR LANE LANSI
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
OLDACC	ACCOLD FOR COMPUTING DECELERATION TO A STOP
RADICL	RADICAL FOR COMPUTING DECELERATION TO A STOP
RELDIS	HALF THE DISTANCE TO THE END OF THE LANE
RESPLA	RELATIVE SPEED BETWEEN VEHICLE AND LAG VEHICLE IN ADJACENT LANE
RESPLE	RELATIVE SPEED BETWEEN VEHICLE AND LEAD VEHICLE IN ADJACENT LANE
SLOPE	ACC/DEC SLOPE REQUIRED FOR DESIRED ACTION
SLPDEC	ACC/DEC SLOPE REQUIRED FOR DECELERATION TO A STOP
T	TIME FOR CHECKING LEAD/LAG GAP
T1	TIME FOR CHECKING LEAD/LAG GAP
VSQT4	VELOLD SQUARED TIMES FOUR
VT2	VELOLD TIMES TWO
X	GAP REQUIRED TO PREVENT COLLISION

SUBROUTINE HOLDSP HOLDS THE VEHICLE SPEED AT ITS CURRENT VALUE
(CALLED FROM ACDCP CRIDIS)
(CALLS ABORTR NEWVEL)

ACCHLD	SAVED OLDACC
KPRTM	PERCEPTION/REACTION TIME REMAINING (IN DT≠S)
LPRTM	SAVED KPRTM (BECAUSE OF CALL BY REPROCESS, KPRTM MAY BE CHANGED BY NEWVEL)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE IBAP PROCESSES THE VEHICLES ON THE INBOUND APPROACHES AND LOGS NEW VEHICLES INTO THE SYSTEM FROM THE QUEUE BUFFERS AS REQUIRED
(CALLED FROM EXEC)
(CALLS FIND EXTRAC PREST1 LOKIBI PREST2 LOGIC UNBIAS
CHKDSP STORE SIGRES LCHGEO PATHF LCHDES ACDCP
PVAPRT CHKLDT SSIBAP INTLOG LOGIBI BANGS BIAS
REPACK LOGIN)

FLENV	FOUR VEHICLE LENGTHS
IAPRT	T/F FLAG FOR INBOUND APPROACH INFORMATION PRINTED
IDESPD	DESIRED SPEED FOR VEHICLE FOR THIS DT
IENT1	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY APPRO
IENT3	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY LANE
IENT6	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHD
IENT7	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
IENT8	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHIL
IHPRT	T/F FLAG FOR INBOUND APPROACH HEADING PRINTED
ILPRT	T/F FLAG FOR INBOUND LANE INFORMATION PRINTED
INQUE	T/F FLAG FOR VEHICLE IN A QUEUE
IONE	INTEGER 1
JFINL	MFINL BEFORE LOOK AHEAD
JGO	TEMPORARY STORAGE FOR IGO
JSISET	SIGNAL SETTING FOR THIS LANE 0=NO SIGNAL OR NO CHANGE IN SIGNAL INDICATION FOR LANE >0=ISISET(ICAMPC,IBLN) FROM /SIGFAS/
KSISET	TEMPORARY STORAGE FOR JSISET
NQA	NUMBER OF VEHICLES TO ENTER ON THIS APPROACH FOR THIS DT
NUM	NUMBER OF ATTRIBUTES IN ENTITY
NV	NUMBER OF VEHICLES IN LANE TO BE PROCESSED
NXVEH	ENTRY NUMBER FOR VEH ENTITIES OF THE NEXT VEHICLE IN LANE TO BE PROCESSED
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
POSCHK	POSITION FOR CHECKING FOR QUEUE BROKEN (ENDLN FOR FIRST VEHICLE IN LANE AND PVPOS FOR OTHERS)
POSLAT	LATERAL POSITION IN LANE (IF LCHGE=2) -=LEFT OF CENTER OF LANE +=RIGHT OF CENTER OF LANE
TESTLP	LATERAL DISTANCE ALREADY MOVED IN A LANE CHANGE

SUBROUTINE INFLZN INITIALIZES THE VEHICLE INTERSECTION CONTROL LOGICAL ATTRIBUTES BASED ON THE TYPE OF TRAFFIC CONTROL FOR THIS LANE
(CALLED FROM CHGMLN INTLOG)
(CALLS ABORTR FIND SETCON SIGRES)

JLCH	ILCH FOR THE VEHICLE≠S INTERSECTION PATH
JSISET	SIGNAL SETTING FOR THIS LANE (SEE ISISET IN /SIGFAS/)
MSG915(6)	ERROR MESSAGE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE INITAL INITIALIZES THE PARAMETERS FOR THE SIMULATION
(CALLED FROM EXEC)
(CALLS ABORTR RUSERD RGEOPD RCAMSD RPHASD RLOOPD RDVPRD)

ICOM1(212)	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLES IN /INTER/
ICOM2(1951)	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLES IN /SIGCAM/
ICOM3(370)	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL VARIABLES IN /SUMSTA/ EXCEPT TMTIME(5)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE INTERP PROCESSES THE VEHICLES ON THE INTERSECTION PATHS
(CALLED FROM EXEC)
(CALLS EXTRAC PREST1 LOKIOB PREST2 UNBIAS ACDCP PVAPRT
SSINTR CLRCON FIND STORE BANGS LOGIOB BIAS
REPACK ABORTR)

IDESPD	DESIRED SPEED OF VEHICLE FOR THIS DT
IENT4	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY PATH
IENT6	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHD
IENT7	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
IHPRT	T/F FLAG FOR INTERSECTION HEADING PRINTED
IPPRT	T/F FLAG FOR INTERSECTION PATH INFORMATION PRINTED
ITWO	INTEGER 2
IZERO	INTEGER 0
JFINL	MFINL BEFORE LOOK AHEAD
NUM	NUMBER OF ATTRIBUTES IN ENTITY
NV	NUMBER OF VEHICLES IN INTERSECTION PATH TO BE PROCESSED
NXVEH	ENTRY NUMBER FOR VEH ENTITIES OF THE NEXT VEHICLE IN INTERSECTION PATH TO BE PROCESSED
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

```
   POSLAT          LATERAL POSITION IN LANE (IF LCHGE=2)
                       -=LEFT OF CENTER OF LANE
                       +=RIGHT OF CENTER OF LANE

SUBROUTINE INTLOG  CHECKS THE INTERSECTION CONTROL LOGICAL DEPENDENT ATTRIBUTES
                   AND CALLS THE APPROPRIATE INTERSECTION CONTROL ROUTINES
                   (CALLED FROM IBAP)
                   (CALLS ABORTR FIND   LSTOP  CHKSDR INFLZN PATHF)

   DECMAX          MAXIMUM DECELERATION TO BE USED TO DECELERATE TO A STOP
   F3              VALUE FOR MINUS FOUR-THIRDS
   MSG910(8)       ERROR MESSAGE
   MSG911(11)      ERROR MESSAGE
   N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
   N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
   XCRIT           (6010-7010) CRITICAL DISTANCE FOR VEHICLE BEING WITHIN
                   THE INFLUENCE OF THE INTERSECTION
   XCRIT           (7010-9010) CRITICAL DISTANCE FOR VEHICLE DEDICATING TO
                   AN INTERSECTION PATH

SUBROUTINE INTSTA  PRINTS THE INTERMEDIATE STATISTICS
                   (CALLED FROM EXEC)
                   (CALLS ABORTR EXTIME)

   IPAGE           PRINTER CARRIAGE CONTROL
                       1=SKIP TO THE TOP OF THE NEXT PAGE
                       2=SKIP TO THE BOTTOM OF THE CURRENT PAGE
   IPTURN          CHARACTER DESIGNATION FOR TURN CODE
                       (1-3,1)=(U AND LEFT  )
                       (1-3,2)=(STRAIGHT    )
                       (1-3,3)=(RIGHT       )
   MIBA            ENTRY NUMBER FOR APPRO ENTITY OF INBOUND APPROACH
                   BEING PROCESSED
   NUM             NUMBER OF VEHICLES PROCESSED FOR TURN CODE K AND
                   INBOUND APPROACH MIBA
   N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
   N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
   OASD            OVERALL AVERAGE STOPPED DELAY
   PDELAY          PERCENT STOPPED DELAY TO TOTAL STOPPED DELAY FOR
                   INBOUND APPROACH
   PTURN           PERCENT OF VEHICLES MAKING TURNING MOVEMENT
   SUMDEL          TOTAL STOPPED DELAY FOR THE INTERSECTION
   SUMVOL          TOTAL NUMBER OF VEHICLES PROCESSED FOR THE INTERSECTION
   TIMNOW          TIME INTO THE SIMULATION
   TMINT           TM TIME SINCE LAST CALL TO INTSTA
   TMSIM           TM TIME SINCE END OF START-UP TIME
   TOTDEL          TOTAL STOPPED DELAY FOR INBOUND APPROACH
   TOTVOL          TOTAL NUMBER OF VEHICLES PROCESSED FOR INBOUND APPROACH
   VOLUME          EQUIVALENT HOURLY VOLUME OF VEHICLES PROCESSED

FUNCTION    ISLCPF SETS UP THE LOW CORE POINTERS AND FILE ENVIRONMENT TABLE
                   FOR A FILE AT EXECUTION TIME (CDC ONLY)
                   (CALLED FROM EXEC)

SUBROUTINE LCHDES  DETERMINES IF A LANE CHANGE IS DESIRABLE
                   (CALLED FROM OBAP   IBAP)
                   (CALLS ABORTR CHKLSI SVEHU  FIND   DELAY  GAPACC CHGMLN
                          PATHF)

   CARDEC          CAR FOLLOWING DECELERATED (DECELERATION WITH NOSF VEHICLE)
   CARDIS          CAR FOLLOWING DISTANCE FOR NOSF DISTANCE
   CRISLP          CRITICAL ACC/DEC SLOPE FOR VEHICLE
   DECMAX          MAXIMUM DECELERATION FOR DRIVER FOR NORMAL DECELERATION
                   TO STOP
   DENOM           6 TIMES REMAINING DISTANCE TO NEAREST OBJECT FORWARD
   JLCH            ILCH FOR LINKING INTERSECTION PATH FOR VEHICLE
   JSET            TEMPORARY STORAGE FOR ISET

   LANSI           ENTRY NUMBER OF LANE ENTITY OF LANE TO BE CHECKED ON
                   THE SIDE OF INTEREST
   LOK             FLAG INDICATING WHETHER OR NOT AN ADJACENT LANE IS AVAILABLE
                   AT THIS POINT (AT POSNEW)
                       0=LANE IS AVAILABLE AND NOT BLOCKED
                       1=LANE IS NOT AVAILABLE AT POSNEW
                       2=VEHICLE PAST END OF LANE AT POSNEW
   MSG903(7)       ERROR MESSAGE
   MSG904(7)       ERROR MESSAGE
   MSG905(16)      ERROR MESSAGE
   NOQ             NUMBER OF VEHICLES AHEAD OF PRESENT VEHICLE IN
                   ADJACENT LANE
   N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
   N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
   OLDACC          OLD ACC/DEC FOR DECELERATION TO STOP
   RADICL          RADICAL FOR COMPUTING DECELERATION TO A STOP
   RELDIS          RELATIVE DISTANCE WITH NOSF VEHICLE (PVSF-POSOLD)
   RELSPD          RELATIVE SPEED WITH NOSF VEHICLE (VVSF-VELOLD)
   SLPDEC          ACC/DEC SLOPE REQUIRED FOR DECELERATION TO A STOP
   VSQT4           VELOLD SQUARED TIMES 4
   VT2             VELOLD TIMES 2

SUBROUTINE LCHGEO  COMPUTES THE NEW LATERAL POSITION FOR A LANE CHANGE USING A
                   COSINE CURVE AND IF FINISHED THEN ENDS THE LANE CHANGE
                   (CALLED FROM OBAP   IBAP)
                   (CALLS ABORTR ENDLCH)

   DVFACT          DRIVER/VEHICLE FACTOR
   N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
   N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
   PI              VALUE OF PI
   POSLAT          LATERAL DISTANCE REMAINING BEFORE LANE CHANGE IS
                   COMPLETED (FEET)
   TLDIST          TOTAL LATERAL DISTANCE FOR A LANE CHANGE
   XNEW            NEW DISTANCE DOWN XTOT THAT VEHICLE HAS ALREADY TRAVELED
   XOLD            DISTANCE DOWN XTOT THAT VEHICLE HAS ALREADY TRAVELED
   XTOT            LENGTH OF LANE CHANGE MANEUVER ALONG DIRECTION OF
                   TRAVEL (FEET)

SUBROUTINE LOGIBI  LOGS THE VEHICLE OUT OF THE INBOUND APPROACH AND LANE AND
                   INTO THE LINKING INTERSECTION PATH FOR THE VEHICLE
                   (CALLED FROM IBAP)
                   (CALLS ABORTR ENDLCH PATHF  SETCON EXTRAC STORE
                          FLGNOR FIND)

   DTIME           TIME VEHICLE ENTERED THE INTERSECTION
   JVEL            IVEL FOR NOF VEHICLE
   LPREV           ENTRY NUMBER FOR LANE ENTITY OF PREVIOUS LINK
   MOGFLG          LOGFLG FOR NOR VEHICLE
   NSKP            NUMBER OF COLUMNS TO SKIP OVER TO POSITION PRINT OF DTIME
                   UNDER COLUMN FOR APPROACH AND LANE (FOR MARCH OUT HEADWAYS)
   NVILL           NUMBER OF VEHICLES IN LANE FOR NEXT DT
   N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
   N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
   POSTOT          TOTAL DISTANCE TRAVELED THIS DT FOR VEHICLE (FOR MARCH OUT
                   HEADWAYS)

SUBROUTINE LOGIC   FINDS THE VALUE FOR THE LOGICAL DEPENDENT ATTRIBUTES FOR
                   ENTRY IN OF ENTITY IY BASED ON THE VALUE OF THE LOGICAL
                   INDEPENDENT ATTRIBUTES FOR ENTRY IN OF ENTITY IY IN THE
                   STORAGE STACK AND STORES THEIR VALUES IN THE STORAGE STACK
                   (CALLED FROM IBAP   PREST2)
                   (CALLS LSHIFT IAND   IOR    SMEP)

   IBF             LOCATION OF THE FIRST FUNCTION MASK IN THE IFU ARRAY IN
                   /FUN/ FOR ENTITY IY
   IDW             LOCATION OF THE LOGICAL DEPENDENT ATTRIBUTE WORD IN THE
                   STORAGE STACK RELATIVE TO THE FIRST WORD IN THE STORAGE STACK
```

	FOR ENTRY IN OF ENTITY IY
IEF	LOCATION OF THE LAST FUNCTION MASK IN THE IFU ARRAY IN /FUN/ FOR ENTITY IY
IFW	LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
IIEN	SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
IIFU	SINGLE INDEX FOR IFU ARRAY OF /FUN/
IIV	LOGICAL PRODUCT (AND) OF THE LOGICAL INDEPENDENT ATTRIBUTE WORD AND THE FUNCTION MASK
IIW	LOCATION OF THE LOGICAL INDEPENDENT ATTRIBUTE WORD IN THE STORAGE STACK RELATIVE TO THE FIRST WORD IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
IN	ENTRY NUMBER FOR ENTITY IY
ISIDW	LOGICAL DEPENDENT WORD FOR THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
ISIIW	LOGICAL INDEPENDENT WORD FROM THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
ISNAME(2)	SUBROUTINE NAME FOR PRINTING (LOGIC)
IY	ENTITY NUMBER 1=APPRO 2=CONFLT 3=LANE 4=PATH 5=SDR 6=VEHD 7=VEHF 8=VEHIL
LTF	LOGICAL TRUE/FALSE FOR LOGICAL DEPENDENT ATTRIBUTE PATH 1=TRUE 2=FALSE
NWE	NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE LOGIN LOGS THE NEW VEHICLE INTO THE INBOUND APPROACH AND LANE AND INITIALIZES THE VEHICLE ATTRIBUTES
(CALLED FROM IBAP)
(CALLS ABORTR STORE CHKMLN NEWVEL CARFOL BIAS PVAPRT REPACK)

CRISLP	CRITICAL ACC/DEC SLOPE FOR DRIVER
DECMAX	MAXIMUM DECELERATION FOR DRIVER FOR NORMAL DECELERATION TO STOP
DIST	DISTANCE FROM REAR BUMPER OF LEAD VEHICLE AND START OF LANE
FACT	FACTOR TO MULTIPLY DECMAX WHEN CALCULATING MAXIMUM ENTRY VELOCITY
IB	INDEX NUMBER FOR IBUF AND QTIME ARRAYS IN /QUE/ WHICH CONTAINS INFORMATION ABOUT VEHICLE
IDESPD	DESIRED SPEED FOR THE VEHICLE FOR THIS DT
IENT6	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHD
IENT7	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
IENT8	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHIL
IONE	INTEGER 1
IQQ	ENTRY NUMBER FOR VEH ENTITIES FOR VEHICLE BEING LOGGED INTO THE SYSTEM
MCHGE	LCHGE FOR THE NEW VEHICLE
MSG920(11)	ERROR MESSAGE
NUM	NUMBER OF ATTRIBUTES IN ENTITY
NVILL	NUMBER OF VEHICLES IN LANE FOR NEXT DT
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
ONETRD	VALUE OF ONE-THIRD
POSLAT	LATERAL POSITION OF VEHICLE IN LANE -=LEFT OF CENTER OF LANE +=RIGHT OF CENTER OF LANE
SLOPE	SLOPE OF ACC/DEC REQUIRED TO KEEP THE VEHICLE ENTERING AT V SPEED FROM RUNNING INTO LEAD VEHICLE
SLP	MAXIMUM CRITICAL ACC/DEC SLOPE FOR ANY DRIVER
T	TIME FOR ENTERING VEHICLE TO TRAVEL BEFORE HIS VELOCITY MUST BE PVVEL
TSTOP	TIME IT WOULD TAKE THE LEAD VEHICLE TO STOP AT CURRENT VELOCITY AND DECELERATION AND MOST AGGRESSIVE DRIVER ACC/DEC SLOPE
V	MAXIMUM INITIAL VELOCITY UPON ENTRY WHICH WILL ALLOW THE DRIVER TO DECREASE HIS SPEED BEFORE HE RUNS INTO LEAD VEHICLE
XSTOP	DISTANCE FROM REAR BUMPER OF LEAD VEHICLE AFTER TSTOP SECONDS AND START OF LANE
XTIMEL	PORTION OF DT THAT VEHICLES SHOULD BE PROCESSED

SUBROUTINE LOGIOB LOGS THE VEHICLE OUT OF THE INTERSECTION PATH AND INTO THE LINKING OUTBOUND APPROACH AND LANE
(CALLED FROM INTERP)
(CALLS ABORTR EXTRAC STORE FLGNOR FIND CHKMLN)

JPOS	IPOS FOR LAST VEHICLE ON LINKING OUTBOUND APPROACH
JVEL	IVEL FOR LAST VEHICLE ON LINKING OUTBOUND APPROACH
MSG902(10)	ERROR MESSAGE
NVILL	NUMBER OF VEHICLES IN LANE FOR NEXT DT
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE LOGOUT ADDS THE VEHICLES SIMULATION STATISTICS FOR THE INBOUND APPROACH, AND TURN CODE AND LOGS THE VEHICLE OUT OF THE SYSTEM, THE OUTBOUND APPROACH, AND THE OUTBOUND LANE
(CALLED FROM OBAP)
(CALLS ABORTR STORE FLGNOR)

AMAXV	ACCELERATION MAXIMUM FOR VEHICLE (FT/SEC/SEC)
AVGSPD	AVERAGE DESIRED SPEED FOR VEHICLE (FT/SEC)
AVGVEL	TIME MEAN SPEED FOR VEHICLE (MPH)
DESPD	AVERAGE DESIRED SPEED FOR VEHICLE (MPH)
DMAXV	DECELERATION MAXIMUM FOR VEHICLE (FT/SEC/SEC) (EQUIVALENT UNIFORM RATE)
INDEX	SINGLE INTEGER SUBSCRIPT FOR DOUBLE SUBSCRIPTED ARRAYS DIMENSIONED TO (6,3) IN /SUMSTA/ (INDEX) = (I,J)
NVILL	NUMBER OF VEHICLES IN LANE FOR NEXT DT
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
XDISTL	DISTANCE LEFT TO TRAVEL TO END OF LANE FOR VEHICLE LOGGING OUT
XDMPH	DELAY BELOW XX MPH (SECONDS)
XQD	QUEUE DELAY FOR VEHICLE (SECONDS)
XSD	STOPPED DELAY FOR VEHICLE (SECONDS)
XSTIME	TOTAL SIMULATION TIME FOR VEHICLE (SECONDS)
XTD	TOTAL DELAY FOR VEHICLE (SECONDS)
XVMT	VEHICLES MILES OF TRAVEL (MILES)

SUBROUTINE LOKIBI LOOKS AHEAD INTO THE LINKING INTERSECTION PATH FOR THIS VEHICLE AND IF THERE IS A VEHICLE ON THE INTERSECTION PATH THEN RESET THE PREVIOUS VEHICLE PARAMETERS TO THE LAST VEHICLE ON THE INTERSECTION PATH ELSE RESET THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE INTERSECTION PATH
(CALLED FROM IBAP)
(CALLS ABORTR FIND STORE)

JACC	IACC FOR LAST VEHICLE ON LINKING INTERSECTION PATH FOR VEHICLE
JPOS	IPOS FOR LAST VEHICLE ON LINKING INTERSECTION PATH FOR VEHICLE
JVEHCL	IVEHCL FOR LAST VEHICLE ON LINKING INTERSECTION PATH FOR VEHICLE
JVEL	IVEL FOR LAST VEHICLE ON LINKING INTERSECTION PATH FOR VEHICLE
LGEOM1	LGEOM(1) FOR LINKING OUTBOUND LANE FOR LINKING INTERSECTION INTERSECTION PATH LNEXT FOR VEHICLE

HENP	LENP FOR LINKING INTERSECTION PATH FOR VEHICLE
HOBL	LINKING OUTBOUND LANE FOR LINKING INTERSECTION PATH LNEXT FOR VEHICLE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE LOKIOB LOOKS AHEAD INTO THE LINKING OUTBOUND LANE FOR THE INTERSECTION PATH AND IF THERE IS A VEHICLE ON THE LANE THEN RESETS THE PREVIOUS VEHICLE PARAMETERS TO THAT VEHICLE ELSE RESETS THE PREVIOUS VEHICLE PARAMETERS TO THE END OF THE LANE
(CALLED FROM INTERP)
(CALLS ABORTR FIND STORE)

JACC	IACC FOR LAST VEHICLE ON LINKING OUTBOUND LANE FOR VEHICLE
JPOS	IPOS FOR LAST VEHICLE ON LINKING OUTBOUND LANE FOR VEHICLE
JVEHCL	IVEHCL FOR LAST VEHICLE ON LINKING OUTBOUND LANE FOR VEHICLE
JVEL	IVEL FOR LAST VEHICLE ON LINKING OUTBOUND LANE FOR VEHICLE
LGEOM1	LGEOM(1) FOR LNEXT LANE (SEE LANE ENTITY)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE LSTOP CHECKS TO SEE IF THE VEHICLE MAY ENTER THE INTERSECTION WITHOUT BLOCKING ANY VEHICLE STOPPED AT THE INTERSECTION BEFORE THIS VEHICLE AND IF OK THEN CHECKS SIGHT DISTANCE RESTRICTIONS AND IF CLEAR THEN CHECKS INTERSECTION CONFLICTS AND IF CLEAR THEN THE VEHICLE MAY PROCEED INTO THE INTERSECTION
(CALLED FROM INTLOG)
(CALLS ABORTR EXTRAC FIND CHKSDR)

ICONP1	ICONP(1) FOR INTERSECTION CONFLICT JNDEX
ICONP2	ICONP(2) FOR INTERSECTION CONFLICT JNDEX
JNDEX	ENTRY NUMBER FOR CONFLT ENTITY OF INTERSECTION CONFLICT
JV	ENTRY NUMBER FOR VEH ENTITIES OF VEHICLE BEING CHECKED
MCPSET	NCPSET FOR MNEXT INTERSECTION PATH
MNEXT	ENTRY NUMBER FOR PATH ENTITY OF INTERSECTION PATH BEING CHECKED
MOGFLG	LOGFLG FOR VEHICLE JV
NLUNC	MLUNC FOR VEHICLE JV
NPRO	MPRO FOR VEHICLE BEING CHECKED AGAINST (JV)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
THES	TIME FOR HESITATION FOR DRIVER ENTERING THE INTERSECTION

SUBROUTINE NEWVEL CALCULATES THE POS/VEL/ACC FOR THE VEHICLE AFTER T SECONDS
(CALLED FROM UNBIAS ACDCP CRIDIS HOLDSP LOGIN)
(CALLS ABORTR)

DPOS	CHANGE IN POSITION DURING T SECONDS
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
T	TIME INCREMENT FOR CALCULATING CHANGE IN POSITION, VELOCITY, AND ACCELERATION/DECELERATION
TCU	T CUBED
TSQ	T SQUARED

SUBROUTINE OBAP PROCESSES THE VEHICLES ON THE OUTBOUND APPROACHES
(CALLED FROM EXEC)
(CALLS EXTRAC PREST1 PREST2 UNBIAS LCHGEO LCHDES ACDCP PVAPRT SSOBAP BANGS BIAS REPACK LOGOUT)

IAPRT	T/F FLAG FOR APPROACH INFORMATION PRINTED
IDESPD	DESIRED SPEED FOR VEHICLE FOR THIS DT
IENT1	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY APPRO
IENT3	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY LANE
IENT6	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHD
IENT7	SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN ENTITY VEHF
IHPRT	T/F FLAG FOR OUTBOUND APPROACH HEADING PRINTED
ILPRT	T/F FLAG FOR LANE INFORMATION PRINTED
IONE	INTEGER 1
IZERO	INTEGER 0
NUM	NUMBER OF ATTRIBUTES IN ENTITY
NV	NUMBER OF VEHICLES IN LANE TO BE PROCESSED
NXVEH	ENTRY NUMBER FOR VEH ENTITIES OF THE NEXT VEHICLE IN LANE TO BE PROCESSED
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
POSLAT	LATERAL POSITION IN LANE (IF LCHGE=2) -=LEFT OF CENTER OF LANE +=RIGHT OF CENTER OF LANE

SUBROUTINE PATHF FINDS THE INTERSECTION PATH FOR THIS VEHICLE BASED ON THE CURRENT APPROACH, CURRENT LANE, AND THE DESIRED OUTBOUND APPROACH
(CALLED FROM IBAP LOGIBI LCHDES CHGMLN ADLVAI INTLOG)
(CALLS ABORTR EXTRAC FIND CHKMLN STORE)

IFORCE	T/F WHETHER TO FORCE THE VEHICLE TO SET LNEXT; IF THERE IS NO INTERSECTION PATH TO THE VEHICLE DESIRED OUTBOUND APPROACH FROM THE CURRENT LANE, THEN SET TO THE STRAIGHT THROUGH INTERSECTION PATH ELSE SET TO FIRST INTERSECTION PATH FOR THE LANE
ILANE	ENTRY NUMBER FOR LANE ENTITY FOR LANE BEING CHECKED
JOPT	IOPT FOR INTERSECTION PATH LPATH
JPT	IPT FOR INTERSECTION PATH LPATH
LFORCE	ENTRY NUMBER FOR PATH ENTITY OF THE FIRST INTERSECTION PATH IN THE LIST OF INTERSECTION PATHS CONNECTING TO THIS LANE, OR THE STRAIGHT THROUGH INTERSECTION PATH, IF AVAILABLE (THIS IS FOR CONTINGENCY INTERSECTION PATH IF AN INTERSECTION PATH WITH THE DESIRED OUTBOUND APPROACH DOES NOT EXIST)
LPATH	ENTRY NUMBER FOR PATH ENTITY OF INTERSECTION PATH BEING CHECKED
MOBAP	LOBAP FOR INTERSECTION PATH LPATH
MPINT	NPINT FOR LANE ILANE
MSG916(11)	ERROR MESSAGE
NN1	FIRST 4 CHARACTERS OF THE ROUTINE NAME OF CALLING ROUTINE
NN2	LAST 2 CHARACTERS OF THE ROUTINE NAME OF CALLING ROUTINE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE PREDTV PREDICTS THE TIME AND VELOCITY TO AN INTERSECTION CONFLICT
(CALLED FROM CHKSDR CHKCON)
(CALLS ABORTR)

A	COEFFICIENT OF T SQUARED FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
ACC	NEW ACCELERATION FOR THIS DT
ACCM	ACCELERATION MAXIMUM FOR THIS DRIVER
ACCV	ACCELERATION MAXIMUM FOR THIS VEHICLE
AN	ACCELERATION NEW (AT END OF DT)
AO	ACCELERATION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
AX	ACC/DEC AT P
B	COEFFICIENT OF T FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
C	CONSTANT OF T FOR FINDING THE TIME REQUIRED TO

	REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
CRISLP	CRITICAL VALUE OF ACC/DEC SLOPE FOR DRIVER
DV	DESIRED VELOCITY FOR THIS DT
JD	DRIVER CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSLIM	SPEED LIMIT FOR APPROACH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPD	DESIRED SPEED FOR VEHICLE FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPDP	0/1 FOR NO/YES IF VEHICLE HAS SET DESIRED SPEED FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JV	VEHICLE CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
LGEOM4	LGEOM(4) FOR LANE FOR PREDICTING TIME TO INTERSECTION CONFLICT
MIMP	SPEED LIMIT FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
P	POSITION OF INTERSECTION CONFLICT (LGEOM4+ICOND(J)) FOR PREDICTING TIME TO INTERSECTION CONFLICT
PN	POSITION NEW (AT END OF DT)
PO	POSITION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
RADICL	RADICAL FOR FINDING THE TIME REQUIRED TO REDUCE HIS SPEED TO HIS DESIRED SPEED BEFORE HE GETS TO THE END OF HIS LANE
RELDIS	RELATIVE DISTANCE TO THE END OF HIS LANE
SLOPE	ACC/DEC SLOPE REQUIRED FOR VELOCITY CHANGE TO SPD
SN	SLOPE NEW (AT END OF DT)
SO	ACC/DEC SLOPE OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
SPD	DESIRED SPEED FOR INTERSECTION PATH
T	TIME TO INTERSECTION CONFLICT
TT	TIME REQUIRED FOR VELOCITY CHANGE TO SPD
VN	VELOCITY NEW (AT END OF DT)
VO	VELOCITY OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
VTT	VELOCITY AT TT SECONDS
VX	VELOCITY AT INTERSECTION CONFLICT
XCRIT	DISTANCE REQUIRED FOR VELOCITY CHANGE TO SPD (MINIMUM OF 4*SPD)
XPER	REMAINING DISTANCE TO INTERSECTION CONFLICT DIVIDED BY DISTANCE TRAVELED DURING LAST DT
XT	TIME TO DECREASE VELOCITY TO ZERO

SUBROUTINE PREST1 EXTRACTS ENTRY IV OF ENTITY VEHF, RESETS THE PREVIOUS VEHICLE PARAMETERS TO THE NEW NOF IF THE VEHICLE IS LANE CHANGING, AND INITIALIZES SEVERAL PARAMETERS FOR THE VEHICLE
(CALLED FROM OBAP INTERP IBAP)
(CALLS ABORTR EXTRAC FIND)

ININT	T/F FOR VEHICLE IN THE INTERSECTION
JACC	IACC FOR NOF VEHICLE
JPOS	IPOS FOR NOF VEHICLE
JVEHCL	IVEHCL FOR NOF VEHICLE
JVEL	IVEL FOR NOF VEHICLE
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE PREST2 COMPUTES NEW ACC/DEC LOGIC AND EXTRACTS ENTRY IV OF ENTITY VEHD FOR THE VEHICLE
(CALLED FROM OBAP INTERP IBAP)
(CALLS ABORTR STORE FIND LOGIC EXTRAC)

N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE PRESIG SIMULATES THE PRE-TIMED SIGNAL CONTROLLER
(CALLED FROM EXEC)
(CALLS ABORTR)

DTIME	TIME THE SIGNAL CHANGES TO GREEN (FOR MARCH OUT HEADWAYS)
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE PSTATS PRINTS SUMMARY STATISTICS FOR INBOUND APPROACH I AND TURN CODE J AND OPTIONALLY WRITES THE STATISTICS ONTO TAPE 7 USING APPROACH NUMBER IWIA AND TURN CODE IWTC
(CALLED FROM SUMARY)

ADMAST	AVERAGE DELAY BELOW XMPH MPH/AVERAGE TRAVEL TIME
ADMPH	AVERAGE DELAY BELOW XMPH MPH (SECONDS)
ADSPD	AVERAGE DESIRED SPEED (MPH)
AMAXV	AVERAGE MAXIMUM ACCELERATION (FT/SEC/SEC)
AQD	AVERAGE QUEUE DELAY (SECONDS)
AQDAST	AVERAGE QUEUE DELAY/AVERAGE TRAVEL TIME
ASD	AVERAGE STOPPED DELAY (SECONDS)
ASDAST	AVERAGE STOPPED DELAY/AVERAGE TRAVEL TIME
ASTIM	AVERAGE TRAVEL TIME (SECONDS)
ATD	AVERAGE TOTAL DELAY (SECONDS)
ATDAST	AVERAGE TOTAL DELAY/AVERAGE TRAVEL TIME
AVMT	AVERAGE VEHICLE-MILES OF TRAVEL
DMAXV	AVERAGE MAXIMUM DECELERATION (FT/SEC/SEC)
I	INBOUND APPROACH NUMBER
INDEX	SINGLE INTEGER SUBSCRIPT FOR DOUBLE SUBSCRIPTED ARRAYS DIMENSIONED TO (6,3) IN /SUMSTA/ (INDEX) = (I,J)
IPRINT	YES/NO FOR PRINTING OF STATISTICS
IWIA	INBOUND APPROACH NUMBER TO USE FOR WRITING STATISTICS TO TAPE
IWTC	TURN CODE NUMBER TO USE FOR WRITING STATISTICS TO TAPE
J	TURN CODE NUMBER
NUM	NUMBER OF VEHICLES PROCESSED
NYES	CHARACTERS (YES)
OADMPH	OVERALL AVERAGE DELAY BELOW XMPH MPH (SECONDS)
OAQD	OVERALL AVERAGE QUEUE DELAY (SECONDS)
OASD	OVERALL AVERAGE STOPPED DELAY (SECONDS)
OATD	OVERALL AVERAGE TOTAL DELAY (SECONDS)
PDMPH	PERCENT OF VEHICLES EXPERIENCING DELAY BELOW XMPH MPH
PQD	PERCENT OF VEHICLES EXPERIENCING QUEUE DELAY
PSD	PERCENT OF VEHICLES EXPERIENCING STOPPED DELAY
PTD	PERCENT OF VEHICLES EXPERIENCING TOTAL DELAY
SMSPD	SPACE MEAN SPEED (MPH)
TMSPD	TIME MEAN SPEED (MPH)
VOLUME	VOLUME PROCESSED (VEHICLES PER HOUR)
XMPH	XX ASSOCIATED WITH DELAY BELOW XX MPH (MPH)

SUBROUTINE PVAPRT PRINTS POS/VEL/ACC FOR THE VEHICLE
(CALLED FROM OBAP INTERP IBAP LOGIN)
(CALLS ABORTR)

IFORM(2)	FORMAT FOR WRITING DATA
IQACC	COLUMN NUMBER FOR ACC/DEC
IQPOS	COLUMN NUMBER FOR POSITION
IQV	ONES DIGIT OF VEHICLE NUMBER
IQVEL	COLUMN NUMBER FOR VELOCITY
N1	FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2	LAST 2 CHARACTERS OF THE ROUTINE NAME
V	VEHICLE NUMBER DIVIDED BY 10 (FOR FINDING ONES DIGIT)

SUBROUTINE QUEUE DETERMINES WHICH VEHICLES IN THE QUEUE BUFFER ARE TO BE LOGGED INTO THE SYSTEM THIS DT
(CALLED FROM EXEC)
(CALLS ABORTR)

IB	INDEX NUMBER FOR IBUF AND QTIME ARRAYS IN /QUE/ WHICH CONTAINS INFORMATION ABOUT VEHICLE
JA	ENTRY NUMBER FOR APPRO ENTITY OF APPROACH FOR VEHICLE

```
                ENTITY [1#12] (SEE IA IN /INDEX/)
JAN             APPROACH NUMBER FOR VEHICLE ENTRY [1#6] (SEE IAN IN
                /INDEX/)
JLN             LANE NUMBER FOR VEHICLE ENTRY, COUNTED FROM MEDIAN TO
                CURB [1#6] (SEE ILN IN /INDEX/)
N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2              LAST  2 CHARACTERS OF THE ROUTINE NAME


SUBROUTINE RCAMSD READS THE CAM STACK INFORMATION FROM THE INPUT DIRECTLY TO
                THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
                (CALLED FROM INITAL)
                (CALLS ABORTR FIND)

IBLNK1          CHARACTERS (     )
IISIGN(4)       CHARACTER DESIGNATION FOR SIGNAL INDICATION
                    (1)=G=GREEN
                    (2)=A=AMBER
                    (3)=R=RED
                    (4)=P=PROTECTED GREEN
IITURN(3)       CHARACTER DESIGNATION FOR TURN CODES FOR SIGNAL
                    (1)=L=LEFT
                    (2)=S=STRAIGHT
                    (3)=R=RIGHT
ILETTA          CHARACTERS (A    )
ILETTN          CHARACTERS (N    )
ILETTS          CHARACTERS (S    )
ILETTU          CHARACTERS (U    )
IPHTIM          SIGNAL PHASE TIME FOR CAM STACK POSITION (SEC)
ISVAL(3,4,3)    SIGNAL INDICATION NUMBER INDEXED BY (IITURN,IISIGN,
                IISIGN) (-1 MEANS ILLEGAL) (SEE ISISET IN /SIGFAS/)
                [1#25]
JBLN            IBLN FOR LANE J
K               INDEX FOR CHARACTERS FOR TURN CODES
LANESS(75)      CHARACTERS FOR TURN CODES (3 CHARACTERS FOR 25 INBOUND LANES)
MCONTR          LCONTR FOR LANE J
NLC             NUMBER OF CHARACTERS TO BE READ IN FOR EACH SIGNAL
                INTERVAL FOR ALL INBOUND LANES (=3*NIBL)
N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2              LAST  2 CHARACTERS OF THE ROUTINE NAME


SUBROUTINE RDVPRD READS THE DRIVER-VEHICLE PROCESSOR DATA FROM THE
                DRIVER-VEHICLE PROCESSOR TAPE, INITIALIZES THE QUEUE BUFFERS,
                AND CHECKS FOR ERRORS
                (CALLED FROM INITAL)
                (CALLS ABORTR)

IAMAX(15)       MAXIMUM UNIFORM ACCELERATION RATE FOR EACH VEHICLE CLASS
                (FT/SEC/SEC)
IDCHAR(5)       DRIVER CHARACTERISTICS FOR EACH DRIVER CLASS (AVERAGE
                DRIVER=100, AGGRESSIVE DRIVER=>100, SLOW DRIVER=<100)
                (SEE DCHAR IN /CLASS/)
IDMAX(15)       MAXIMUM UNIFORM DECELERATION RATE FOR EACH VEHICLE CLASS
                (FT/SEC/SEC)
IVCHAR(15)      VEHICLE CHARACTERISTIC FOR EACH VEHICLE CLASS (AVERAGE
                VEHICLE=100, RESPONSIVE VEHICLE=>100, SLUGGISH
                VEHICLE=<100) (SEE VCHAR IN /CLASS/)
IVMAX(15)       MAXIMUM VELOCITY FOR EACH VEHICLE CLASS (FT/SEC)
NDRICL          NUMBER OF DRIVER CLASSES
NVEHCL          NUMBER OF VEHICLE CLASSES
N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2              LAST  2 CHARACTERS OF THE ROUTINE NAME
PIJRMI          PIJR MINIMUM VALUE


SUBROUTINE REPACK REPACKS THE VALUES OF THE ATTRIBUTES FROM THE COMMON BLOCK
                FOR ENTITY IY INTO ENTRY IN OF ENTITY IY IN THE STORAGE STACK
                (CALLED FROM RGEOPD OBAP   INTERP IBAP   LOGIN)
                (CALLS LSHIFT IAND   INOT   IOR    SMEP)

IBA             LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
                FOR ENTITY IY
ID              SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES
                IN ALL THE ENTITIES
IE              SMEP ERROR NUMBER
IEA             LOCATION OF THE LAST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
                FOR ENTITY IY
IFW             LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
                ENTRY IN OF ENTITY IY
IIAT            SINGLE INDEX FOR IAT ARRAY OF /ATTB/
IIEN            SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
ILW             LOCATION OF THE LAST COMPUTER WORD IN THE STORAGE STACK FOR
                ENTRY IN OF ENTITY IY
IN              ENTRY NUMBER FOR ENTITY IY
IR              VALUE OF CURRENT ATTRIBUTE BEING REPACKED
ISNAME(2)       SUBROUTINE NAME FOR PRINTING (REPACK)
IT              ATTRIBUTE I LEFT SHIFTED TO ITS PROPER POSITION FOR STORING
                IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
IV              INDEX NUMBER OF CURRENT ATTRIBUTE BEING REPACKED
IWD             LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
                ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
                OF ENTITY IY
IX              TEST IF ATTRIBUTE I IS OUT OF RANGE FOR ENTITY IY
                  <0=OUT OF RANGE
                   0=OK
                  >0=OUT OF RANGE
IY              ENTITY NUMBER
                    1=APPRO
                    2=CONFLT
                    3=LANE
                    4=PATH
                    5=SDR
                    6=VEHD
                    7=VEHF
                    8=VEHIL
NWE             NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY


SUBROUTINE RGEOPD READS THE GEOMETRY PROCESSOR DATA FROM THE GEOMETRY PROCESSOR
                TAPE AND READS THE LANE CONTROL INFORMATION FROM CARD 3 OF
                THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR
                ERRORS
                (CALLED FROM INITAL)
                (CALLS ABORTR REPACK)

IDX             DISTANCE FROM MEDIAN TO CENTER OF LANE (FT)
IENT1           SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY APPRO
IENT2           SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY CONFLT
IENT3           SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY LANE
IENT4           SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY PATH
IENT5           SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY SDR
ITEST           NEXT COLUMN AFTER THE LAST LEGAL LANE CONTROL
IT1             TEMPORARY STORAGE FOR NUMBER OF ARCS AND LINES
                FOR DUMMY READ
IT2             TEMPORARY STORAGE FOR ARC AND LINE INFORMATION FOR
                DUMMY READ
JA              ENTRY NUMBER FOR APPRO ENTITY FOR APPROACH
LCNTRI(50)      LANE CONTROL READ FROM THE INPUT DIRECTLY TO THE
                SIMULATION PROCESSOR
NAP             TOTAL NUMBER OF INBOUND AND OUTBOUND APPROACHES
NUM             NUMBER OF ATTRIBUTES IN THE ENTITY
N1              FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2              LAST  2 CHARACTERS OF THE ROUTINE NAME


SUBROUTINE RLOOPD READS THE DETECTOR INFORMATION FROM THE INPUT DIRECTLY TO THE
```

SUBROUTINE RLOOPD
SIMULATION PROCESSOR AND CHECKS FOR ERRORS
(CALLED FROM INITAL)
(CALLS ABORTR FIND STORE)

IBLNK1 CHARACTERS ( )
ID DETECTOR NUMBER
IENCE CHARACTERS (ENCE)
ILDLN INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF LANE NUMBER FOR DETECTOR (COUNTED FROM MEDIAN TO CURB) [1#6]
IPRES CHARACTERS (PRES)
IPULS CHARACTERS (PULS)
IT1 SECOND 4 CHARACTERS FOR DETECTOR TYPE
IUSED(20) FLAG FOR DATA ENTERED FOR DETECTOR
0=NOT USED
1=USED
JL ENTRY NUMBER FOR LANE ENTITY OF LANE FOR DETECTOR
LDA ENTRY NUMBER FOR APPRO ENTITY FOR APPROACH NUMBER FOR DETECTOR [1#12]
ILDLN INDEX NUMBER FOR LLANES ARRAY OF APPRO ENTITY OF LANE NUMBER FOR DETECTOR (COUNTED FROM MEDIAN TO CURB) [1#6]
LDSTOP DETECTOR STOPPING POSITION
LDSTRT DETECTOR STARTING POSITION
LGEOM3 LGEOM(3) FOR LANE FOR DETECTOR
LGEOM4 LGEOM(4) FOR LANE FOR DETECTOR
LLDLN(6) LIST OF DETECTOR LANE NUMBERS
MLANES NUMBER OF LANES FOR APPROACH FOR DETECTOR
N TEMPORARY STORAGE FOR NLD(JP) (SEE /PHASES/)
NLDL NUMBER OF DETECTORS FOR LANE JL
NLDLN NUMBER OF DETECTOR LANE NUMBERS
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE RPHASD READS THE SIGNAL PHASE INFORMATION FROM THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
(CALLED FROM INITAL)
(CALLS ABORTR)

IBLNK1 CHARACTERS ( )
INO CHARACTERS (NO)
IOFF CHARACTERS (OFF)
ION CHARACTERS (ON)
IOR CHARACTERS (OR)
ITEST 0/1 FOR NO/YES IF THERE ARE POSITIVELY CONNECTED DETECTORS FOR THE SIGNAL PHASE
IT1 TEMPORARY STORAGE FOR STARTING CAM STACK POSITION FOR THE SIGNAL PHASE
IUSED(8) FLAG FOR DATA ENTERED FOR SIGNAL PHASE
0=NOT USED
1=USED
IYES CHARACTERS (YES)
JAND CHARACTERS (AND)
JP SIGNAL CONTROLLER PHASE NUMBER
JPP1 SIGNAL CONTROLLER PHASE NUMBER PLUS 1
JPP2 SIGNAL CONTROLLER PHASE NUMBER PLUS 2
MCAM EXPECTED NUMBER OF CAM STACK POSITIONS FOR SIGNAL PHASE
N NUMBER OF DETECTORS CONNECTED TO SIGNAL PHASE
NCAM ACTUAL NUMBER OF CAM STACK POSITIONS FOR SIGNAL PHASE
NN NUMBER OF SIGNAL PHASES THAT THE SIGNAL PHASE CAN CLEAR TO
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME
TEST TEMPORARY STORAGE FOR TESTING PURPOSES

SUBROUTINE RUSERD READS THE USER DATA FROM CARD 2 OF THE INPUT DIRECTLY TO THE SIMULATION PROCESSOR AND CHECKS FOR ERRORS
(CALLED FROM INITAL)
(CALLS ABORTR)

IBLNK1 CHARACTERS ( )
INO CHARACTERS (NO)
ISTATS INTEGER TIME BETWEEN INTERMEDIATE STATISTICS (SEC)
IXXX EXPERIMENTAL OPTION FOR SKIPPING ALL BOUNDARY CHECKING FOR INPUT CARD 2 TO SIMPRO (X MEANS SKIP CHECKING)
IYES CHARACTERS (YES)
JXXX CHARACTER (X)
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME
XMPH XX ASSOCIATED WITH DELAY BELOW XX MPH (MPH)

SUBROUTINE SETCON SETS THE INTERSECTION CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
(CALLED FROM LOGIBI SIGRES CHKCON INFLZN)
(CALLS ABORTR EXTRAC FIND STORE)

INOW ENTRY NUMBER FOR VEH ENTITIES OF VEHICLE CURRENTLY CHECKING
IPOSCK BIASED POSITION FOR CHECKING WHERE THIS VEHICLE FITS INTO STREAM OF VEHICLES FOR THE INTERSECTION CONFLICT
JCONI ICONI FOR THE OTHER INTERSECTION PATH
JCPSET ICPSET (JCONI) FOR INTERSECTION PATH JP
JGEOCP INDEX NUMBER FOR CONFLT ENTITY FOR INTERSECTION CONFLICT BEING CHECKED
JP ENTRY NUMBER FOR PATH ENTITY OF OTHER INTERSECTION PATH AT INTERSECTION CONFLICT
JPOS IPOS FOR VEHICLE INOW
MCPSET NCPSET FOR INTERSECTION PATH JP
MOGFLG LOGFLG FOR VEHICLE NOR
MORC NORC FOR VEHICLE INOW
MSG914(5) ERROR MESSAGE
NININT MININT FOR VEHICLE INOW
NOFC ENTRY NUMBER FOR VEH ENTITIES OF NEAREST OBJECT FORWARD OF INOW VEHICLE
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME
POSLAT LATERAL POSITION IN LANE (IF LCHGE=2)

SUBROUTINE SETLDF SETS THE DETECTORS CONNECTED POSITIVE TO THE CURRENT SIGNAL PHASE TO FALSE
(CALLED FROM ACTSIG)
(CALLS ABORTR)

JLD INDEX NUMBER FOR /LOOPS/ OF DETECTORS BEING PROCESSED
NUMLD NUMBER OF DETECTORS CONNECTED TO THE CURRENT SIGNAL PHASE
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE SETPTV SETS THIS VEHICLE PARAMETERS FOR PREDICTING TIME AND VELOCITY TO AN INTERSECTION CONFLICT
(CALLED FROM CHKSDR CHKCON)
(CALLS ABORTR)

AO ACCELERATION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
JD DRIVER CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSLIM SPEED LIMIT FOR APPROACH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPD DESIRED SPEED FOR VEHICLE FOR PREDICTING TIME TO INTERSECTION CONFLICT
JSPDP 0/1 FOR NO/YES IF VEHICLE HAS SET DESIRED SPEED FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
JV VEHICLE CLASS FOR PREDICTING TIME TO INTERSECTION CONFLICT
LGEOM4 LGEOM(4) FOR LANE FOR PREDICTING TIME TO INTERSECTION CONFLICT
MIMP SPEED LIMIT FOR INTERSECTION PATH FOR PREDICTING TIME TO INTERSECTION CONFLICT
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME

```
 N2             LAST  2 CHARACTERS OF THE ROUTINE NAME
 PO             POSITION OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT
 SO             ACC/DEC SLOPE OLD FOR PREDICTING TIME TO INTERSECTION
                CONFLICT
 VO             VELOCITY OLD FOR PREDICTING TIME TO INTERSECTION CONFLICT

SUBROUTINE SIGRES DETERMINES THE APPROPRIATE DRIVER RESPONSE FOR THE NEW SIGNAL
                INDICATION
                (CALLED FROM IBAP   INFLZN)
                (CALLS ABORTR FIND   SETCON UNSETC)

 DECMAX         MAXIMUM DECELERATION TO BE USED TO DECELERATE TO A STOP
 DMPDI          DECMAX + OLDACC
 IENT6          SINGLE DIMENSIONED ARRAY EQUIVALENCED TO VARIABLES IN
                ENTITY VEHD
 JLCH           ILCH FOR LANE
 JSISET         SIGNAL SETTING FOR THIS LANE (SEE ISISET IN /SIGFAS/)
 JTURN          TURN CODE TO TEST IF PRIMARY OR SECONDARY INDICATION
                SHOULD BE USED (SEE ITURN IN VEHF ENTITY)
 KSISET         RELATIVE VALUE OF JSISET FOR TURN CODE
 MSG912(8)      ERROR MESSAGE
 NPRO           MPRO FOR NOF VEHICLE
 N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
 N2             LAST  2 CHARACTERS OF THE ROUTINE NAME
 T3             VALUE FOR MINUS TWO=THIRDS
 XCRIT          CRITICAL DISTANCE FOR STOPPING ON AMBER LIGHT

PROGRAM    SIMPRO SIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION MODEL
                (GENERATED BY COLEASE)
                (CALLS LSHIFT EXEC   EXIT)

SUBROUTINE SMEP   SYSTEM MESSAGE ERROR PROCESSOR FOR COLEASE SUBROUTINES
                (CALLED FROM EXTRAC FIND   REPACK STORE  LOGIC)
                (CALLS ABORTR)

 IE             SMEP ERROR NUMBER
 IERROR(8)      ERROR MESSAGE FOR ABORTR
 IN             ENTRY NUMBER FOR ENTITY IY
 IR             VALUE OF ATTRIBUTE BEING PROCESSED
 ISNAME(2)      SUBROUTINE NAME FOR PRINTING
 IV             ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)
 IY             ENTITY NUMBER
                    1=APPRO
                    2=CONFLT
                    3=LANE
                    4=PATH
                    5=SDR
                    6=VEHD
                    7=VEHF
                    8=VEHIL

SUBROUTINE SSIBAP UPDATES THE VEHICLES SIMULATION STATISTICS ON THE INBOUND
                APPROACH
                (CALLED FROM IBAP)
                (CALLS ABORTR FIND)

 INQUE          T/F FLAG FOR VEHICLE IN A QUEUE
 MIMP           LIMP FOR LINKING INTERSECTION PATH FOR VEHICLE
 N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
 N2             LAST  2 CHARACTERS OF THE ROUTINE NAME
 POSCHK         POSITION FOR CHECKING FOR QUEUE BROKEN (ENDLN FOR FIRST
                VEHICLE IN LANE AND PVPOS FOR OTHERS)
 SPFACT         FACTOR FOR DESIRED SPEED TO FIND THE ENTRY DESIRED SPEED
                FOR INBOUND APPROACH

SUBROUTINE SSINTR UPDATES THE VEHICLES SIMULATION STATISTICS IN THE
SUBROUTINE SSINTR
                INTERSECTION
                (CALLED FROM INTERP)
                (CALLS ABORTR FIND)

 JL             ENTRY NUMBER FOR LANE ENTITY FOR LINKING INBOUND LANE
 JLN            LANE NUMBER FOR LINKING INBOUND LANE
 JSNA           ENTRY NUMBER FOR APPRO ENTITY FOR LINKING INBOUND APPROACH
 MLANES         NUMBER OF LANES FOR LINKING INBOUND LANE
 MSG901(10)     ERROR MESSAGE
 N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
 N2             LAST  2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE SSOBAP UPDATES THE VEHICLE SIMULATION STATISTICS ON THE OUTBOUND
                APPROACH
                (CALLED FROM OBAP)
                (CALLS ABORTR)

 N1             FIRST 4 CHARACTERS OF THE ROUTINE NAME
 N2             LAST  2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE STORE  STORES THE VALUE OF LOCAL INTEGER IR INTO ATTRIBUTE IV OF
                ENTRY IN OF ENTITY IY IN THE STORAGE STACK
                (CALLED FROM RLOOPD LOGOUT FLGNOR LOKIOB CLRCON LOGIOB
                             LOKIBI CHKDSP LOGIBI PREST2 ENDLCH GAPACC
                             CHGMLN CHKCON SETCON UNSETC PATHF  LOGIN)
                (CALLS LSHIFT IAND   INOT   IOR    SMEP)

 IBA            LOCATION OF THE FIRST ATTRIBUTE IN THE IAT ARRAY OF /ATTB/
                FOR ENTITY IY
 ID             SINGLE DIMENSIONED ARRAY EQUIVALENCED TO ALL THE ATTRIBUTES
                IN ALL THE ENTITIES
 IE             SMEP ERROR NUMBER
 IFW            LOCATION OF THE FIRST COMPUTER WORD IN THE STORAGE STACK FOR
                ENTRY IN OF ENTITY IY
 IIAT           SINGLE INDEX FOR IAT ARRAY OF /ATTB/
 IIEN           SINGLE INDEX FOR IEN ARRAY OF /ENTITY/
 IN             ENTRY NUMBER FOR ENTITY IY
 IR             LOCAL INTEGER TO BE STORED IN ATTRIBUTE IV OF ENTRY IN OF
                ENTITY IY
 ISNAME(2)      SUBROUTINE NAME FOR PRINTING (STORE)
 IT             ATTRIBUTE I LEFT SHIFTED TO ITS PROPER POSITION FOR STORING
                IN THE STORAGE STACK FOR ENTRY IN OF ENTITY IY
 IV             ATTRIBUTE NUMBER (RELATIVE TO THE FIRST FOR ENTITY IY)
 IWD            LOCATION OF THE COMPUTER WORD IN THE STORAGE STACK FOR
                ATTRIBUTE I (RELATIVE TO THE START OF THE ENTRY) FOR ENTRY IN
                OF ENTITY IY
 IX             TEST IF ATTRIBUTE I IS OUT OF RANGE FOR ENTITY IY
                  <0=OUT OF RANGE
                   0=OK
                  >0=OUT OF RANGE
 IY             ENTITY NUMBER
                   1=APPRO
                   2=CONFLT
                   3=LANE
                   4=PATH
                   5=SDR
                   6=VEHD
                   7=VEHF
                   8=VEHIL
 NWE            NUMBER OF COMPUTER WORDS IN THE STORAGE STACK FOR ENTITY IY

SUBROUTINE SUMARY PRINTS THE SUMMARY STATISTICS
                (CALLED FROM EXEC   ABORTR)
                (CALLS EXTIME PSTATS ADDSTA ACTSTA TIMSTA)

 APLVDV         AVERAGE PERCENT LOGIN VELOCITY TO DESIRED VELOCITY
 IAN            INDEX NUMBER FOR LIBA ARRAYS OF /INTER/ OF APPROACH BEING
                PROCESSED [1,6]
```

IPTURN(3,3) CHARACTER FOR PRINTING TURN CODES FOR WHICH SUMMARY STATISTICS HAVE BEEN GATHERED (SEE ITURN FOR VEHF ENTITY)
(1=3,1)=(U AND LEFT)
(1=3,2)=(STRAIGHT)
(1=3,3)=(RIGHT)
ITC TURN CODE BEING PROCESSED [1,3]
JA ENTRY NUMBER FOR APPRO ENTITY OF INBOUND APPROACH BEING PROCESSED
LANE LANE NUMBER (1,6)
MPLVDV TOTAL NUMBER OF VEHICLES FOR INTERSECTION FOR PERCENT LOGIN VELOCITY TO DESIRED VELOCITY
NUMC TOTAL NUMBER OF COLLISIONS FOR INTERSECTION
NINE INTEGER NINE
NUME TOTAL NUMBER OF VEHICLES ELIMINATED FOR INTERSECTION
NUMTA NUMBER OF VEHICLES PROCESSED THAT ENTERED ON THIS APPROACH
NYES CHARACTERS (YES)
PTURN(3) PERCENTAGES OF VEHICLES MAKING A TURN FOR THIS APPROACH
(1)=U AND LEFT
(2)=STRAIGHT
(3)=RIGHT
QUEUEL AVERAGE QUEUEL LENGTH FOR THE LANE (VEHICLES/SECOND)
TPLVDV TOTAL FOR INTERSECTION FOR AVERAGE PERCENT LOGIN VELOCITY TO DESIRED VELOCITY

SUBROUTINE SVEHU FINDS THE NEAREST VEHICLE TO THE FRONT AND THE NEAREST VEHICLE TO THE REAR IN THE LANE ON THE SIDE OF INTEREST FOR THAT VEHICLE
(CALLED FROM LCHDES DELAY)
(CALLS ABORTR FIND)

IPOSF POSITION OF VEHICLE TO THE FRONT IN ADJACENT LANE (BIASED FEET)
IPOSR POSITION OF VEHICLE TO THE REAR IN ADJACENT LANE (BIASED FEET)
LANSI ENTRY NUMBER FOR LANE ENTITY OF LANE TO BE CHECKED ON THE SIDE OF INTEREST
LGEOM4 LGEOM(4) FOR LANE ON THE SIDE OF INTEREST
MEGAL LEGAL FOR NOSF/NOSR VEHICLE
NOQ NUMBER OF VEHICLES AHEAD OF PRESENT VEHICLE IN ADJACENT LANE
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE TIMSTA PRINTS THE COMPUTER TIME STATISTICS
(CALLED FROM SUMARY)
(CALLS EXTIME)

ANVSY AVERAGE NUMBER OF VEHICLES IN THE SYSTEM DURING SIMULATION TIME
COSTIN COST FOR INITIALIZATION
COSTSI COST FOR SIMULATION
COSTSS COSTS FOR SUMSTA
COSTSU COSTS FOR START UP
COSTTO TOTAL COST FOR RUN ($230.00 PER COMPUTER HOUR AT UT)
IOUT OUTPUT FILE
TMIN TOTAL TM TIME FOR INITIALIZING SIMPRO
TMRAT VEHICLE-SECONDS SIMULATED/TM SIMULATION TIME
TMRDT VEHICLES UPDATES/TM SIMULATION TIME
TMRSI RATIO OF REAL SIMULATION TIME/TM SIMULATION TIME
TMRSU RATIO OF REAL START UP TIME/TM START UP TIME
TMSI TOTAL TM TIME FOR SIMULATION (AFTER START UP)
TMSS TOTAL TM TIME FOR SUMMARY STATISTICS
TMSU TOTAL TM TIME FOR SIMULATION DURING START UP TIME
TMTO TOTAL TM TIME FOR RUN (TMIN+TMSU+TMSI+TMSS)

SUBROUTINE UNBIAS UNBIASES THE VEHICLE ATTRIBUTES AND PREDICTS THE NEW POS/VEL/ACC
(CALLED FROM OBAP INTERP IBAP)
(CALLS ABORTR NEWVEL)

N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

SUBROUTINE UNSETC UNSETS THE INTERSECTION CONFLICTS FOR THE VEHICLE FOR HIS INTERSECTION PATH
(CALLED FROM CHGMLN SIGRES)
(CALLS ABORTR EXTRAC STORE FIND)

JCONI ICONI FOR INTERSECTION PATH JP
JGEOCP INDEX NUMBER FOR CONFLT ENTITY FOR INTERSECTION CONFLICT BEING CHECKED
JP ENTRY NUMBER FOR PATH ENTITY OF OTHER INTERSECTION PATH AT INTERSECTION CONFLICT
MCPSET NCPSET FOR INTERSECTION PATH JP
MORC NORC FOR VEHICLE NOFC
NOFC ENTRY NUMBER FOR VEH ENTITIES OF NEAREST OBJECT FORWARD FOR UNSETTING INTERSECTION CONFLICTS
N1 FIRST 4 CHARACTERS OF THE ROUTINE NAME
N2 LAST 2 CHARACTERS OF THE ROUTINE NAME

7. ALPHABETICAL LISTING OF ALL THE ROUTINES AND THE ROUTINES WHICH CAN CALL THEM

```
ABORTR = ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   CARFOL CHGMLN CHKCON
         CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR CKLALT CLRCON CRIDIS
         DELAY  ENDLCH EXEC   EXTIME FLGNOR GAPACC HOLDSP INFLZN INITAL
         INTLOG INTSTA LCHDES LCHGEO LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI
         LOKIOB LSTOP  NEWVEL PATHF  PREDTV PRESIG PREST1 PREST2 PVAPRT
         QUEUE  RCAMSD RDVPRD RGEOPD RLOOPD RPHASD RUSERD SETCON SETLDF
         SETPTV SIGRES SMEP   SSIBAP SSINTR SSOBAP SVEHU  UNBIAS UNSETC
ACCEL  = ACDCP  CARFOL CRIDIS
ACDCP  = IBAP   INTERP OBAP
ACTSIG = EXEC
ACTSTA = SUMARY
ADDSTA = SUMARY
ADLVAI = ACDCP
BANGS  = IBAP   INTERP OBAP
BIAS   = IBAP   INTERP LOGIN  OBAP
CARFOL = ACDCP  LOGIN
CHGMLN = LCHDES
CHKCON = CHKSDR
CHKDFP = ACTSIG
CHKDSP = IBAP
CHKLDT = IBAP
CHKLSI = DELAY  LCHDES
CHKMLN = LOGIN  LOGIOB PATHF
CHKSDR = INTLOG LSTOP
CKLALT = DELAY
CLRCON = INTERP
CRIDIS = ACDCP
DELAY  = LCHDES
ENDLCH = ADLVAI LCHGEO LOGIBI
EXEC   = SIMPRO
EXTIME = EXEC   INTSTA SUMARY TIMSTA
EXTRAC = CHKCON CHKSDR CLRCON IBAP   INTERP LOGIBI LOGIOB LSTOP  OBAP
         PATHF  PREST1 PREST2 SETCON UNSETC
FIND   = ADLVAI BANGS  CARFOL CHKCON CHKDSP CHKLSI CHKSDR CHGMLN CKLALT
         CLRCON DELAY  ENDLCH FLGNOR GAPACC IBAP   INFLZN INTLOG LCHDES
         LOGIBI LOGIOB LOKIBI LOKIOB LSTOP  PATHF  PREST1 PREST2 RCAMSD
         RLOOPD SETCON SIGRES SSIBAP SSINTR SVEHU  UNSETC
FLGNOR = CHGMLN LOGIBI LOGIOB LOGOUT
GAPACC = LCHDES
HOLDSP = ACDCP  CRIDIS
IBAP   = EXEC
INFLZN = CHGMLN INTLOG
INITAL = EXEC
INTERP = EXEC
INTLOG = IBAP
INTSTA = EXEC
ISLCPF = EXEC
LCHDES = IBAP   OBAP
LCHGEO = IBAP   OBAP
LOGIBI = IBAP
LOGIC  = IBAP   PREST2
LOGIN  = IBAP
LOGIOB = INTERP
LOGOUT = OBAP
LOKIBI = IBAP
LOKIOB = INTERP
LSTOP  = INTLOG
NEWVEL = ACDCP  CRIDIS HOLDSP LOGIN  UNBIAS
OBAP   = EXEC
PATHF  = ADLVAI CHGMLN IBAP   INTLOG LCHDES LOGIBI
PREDTV = CHKCON CHKSDR
PRESIG = EXEC
PREST1 = IBAP   INTERP OBAP
PREST2 = IBAP   INTERP OBAP
PSTATS = SUMARY
PVAPRT = IBAP   INTERP LOGIN  OBAP
QUEUE  = EXEC
RCAMSD = INITAL
RDVPRD = INITAL
REPACK = IBAP   INTERP LOGIN  OBAP   RGEOPD
RGEOPD = INITAL
RLOOPD = INITAL
RPHASD = INITAL
RUSERD = INITAL
SETCON = CHKCON INFLZN LOGIBI SIGRES
SETLDF = ACTSIG
SETPTV = CHKCON CHKSDR
SIGRES = IBAP   INFLZN
SSIBAP = IBAP
SSINTR = INTERP
SSOBAP = OBAP
STORE  = CHGMLN CHKCON CHKDSP CLRCON ENDLCH FLGNOR GAPACC LOGIBI LOGIN
         LOGIOB LOGOUT LOKIBI LOKIOB PATHF  PREST2 RLOOPD SETCON UNSETC
SUMARY = ABORTR EXEC
SVEHU  = DELAY  LCHDES
TIMSTA = SUMARY
UNBIAS = IBAP   INTERP OBAP
UNSETC = CHGMLN SIGRES
```

8. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED

```
A      =        = ACCEL  CARFOL PREDTV
ACC    =        = ACCEL  CARFOL PREDTV
ACCHLD =        = HOLDSP
ACCM   =        = PREDTV
ACCMAX =        = ACCEL  CARFOL
ACCNEW / ABIAS  / ACCEL  ACDCP  BANGS  BIAS   CARFOL CRIDIS HOLDSP IBAP
                  INTERP LOGIN  NEWVEL OBAP   PVAPRT SETPTV
ACCOLD / ABIAS  / ACCEL  ACDCP  BIAS   CARFOL CHKDSP CRIDIS GAPACC HOLDSP
                  LCHDES LOGIN  NEWVEL SIGRES UNBIAS
ACCV   =        = PREDTV
ACCVEH =        = ACCEL  GAPACC
ACH    =        = CHKCON
ACM    =        = CHKCON CHKSDR
ADESPD / SUMSTA / ADDSTA LOGOUT PSTATS
ADMAST =        = PSTATS
ADMPH  =        = PSTATS
ADSPD  =        = PSTATS
ALAGAP =        = GAPACC
ALEGAP =        = GAPACC
AMAX   / CLASS  / ACCEL  GAPACC PREDTV RDVPRD
AMAXV  =        = LOGOUT PSTATS
AN     =        = PREDTV
ANVSY  =        = TIMSTA
AO     =        = CHKCON PREDTV SETPTV
APIJR  / USER   / CHKCON CHKSDR RDVPRD SUMARY
APLVDV =        = SUMARY
AQD    =        = PSTATS
AQDAST =        = PSTATS
ARCOS  =        = LCHGEO
ASD    =        = PSTATS
ASDAST =        = PSTATS
ASPEED / SUMSTA / ADDSTA LOGOUT PSTATS
ASTIM  =        = PSTATS
ATD    =        = PSTATS
ATDAST =        = PSTATS
ATGAPO =        = ACTSTA
ATMAXO =        = ACTSTA
AUTOL  / USER   / ACCEL  BLKDAT CARFOL LOGOUT PREDTV RDVPRD
AVGSPD =        = LOGOUT
AVGVEL =        = LOGOUT
AVMT   =        = PSTATS
AVSF   / LANECH / CHGMLN GAPACC
AVSR   / LANECH / GAPACC
AX     =        = PREDTV
B      =        = ACCEL  CARFOL PREDTV
C      =        = ACCEL  CARFOL PREDTV
CARDEC =        = CARFOL LCHDES
CARDIS =        = CARFOL LCHDES
CAREQA / USER   / CARFOL INITAL LCHDES RUSERD SUMARY
CAREQL / USER   / CARFOL INITAL LCHDES RUSERD SUMARY
CAREQM / USER   / CARFOL INITAL LCHDES RUSERD SUMARY
CLOCK  =        = EXTIME
CLOCK1 =        = EXTIME
COM01  =        = ABORTR
COM02  =        = ABORTR
COM03  =        = ABORTR
COM04  =        = ABORTR
COM05  =        = ABORTR
COM06  =        = ABORTR
COM07  =        = ABORTR
COM08  =        = ABORTR
COM09  =        = ABORTR
COM10  =        = ABORTR
COM11  =        = ABORTR
COM12  =        = ABORTR
COM13  =        = ABORTR
COM14  =        = ABORTR
COM15  =        = ABORTR
COM16  =        = ABORTR
COSTIN =        = TIMSTA
COSTSI =        = TIMSTA
COSTSS =        = TIMSTA
COSTSU =        = TIMSTA
COSTTO =        = TIMSTA
CRISLP =        = ACCEL  CARFOL CRIDIS GAPACC LCHDES LOGIN  PREDTV
DCH    =        = CHKCON CHKSDR
DCHAR  / CLASS  / ACCEL  CARFOL CHGMLN CHKCON CHKDSP CRIDIS DELAY  GAPACC
                  INTLOG LCHDES LCHGEO LOGIN  PREDTV RDVPRD SIGRES
DCHARM / CLASS  / LOGIN  RDVPRD
DCM    =        = CHKCON CHKSDR
DECMAX =        = CHGMLN CRIDIS GAPACC INTLOG LCHDES LOGIN  SIGRES
DECVEH =        = CARFOL
DENOM  =        = CRIDIS GAPACC LCHDES
DESPD  =        = LOGOUT
DESVEL / ABIAS  / ACCEL  ACDCP  BANGS  CARFOL CHGMLN CHKDSP CRIDIS IBAP
                  INTERP LOGIN  OBAP   UNBIAS
DIST   =        = CARFOL LOGIN
DISTAD / PRTPVA / LOGIBI LOGIN  LOGIOB PVAPRT
DMAX   / CLASS  / CARFOL CHGMLN CRIDIS INTLOG LCHDES RDVPRD SIGRES
DMAXV  =        = LOGOUT PSTATS
DMPDI  =        = SIGRES
DMPH   / SUMSTA / ADDSTA LOGOUT PSTATS
DPOS   =        = NEWVEL
DT     / USER   / ACCEL  ACDCP  ACTSIG CARFOL CHGMLN CHKCON CHKDSP CHKSDR
                  CRIDIS EXEC   GAPACC HOLDSP INTSTA LCHDES LCHGEO LOGIBI
                  LOGIN  LOGOUT LSTOP  NEWVEL PREDTV PRESIG RDVPRD RPHASD
                  RUSERD SIGRES SUMARY SVEHU  TIMSTA UNBIAS
DTCU   / USER   / ACDCP  CHKCON CRIDIS HOLDSP PREDTV RUSERD UNBIAS
DTIME  =        = ACTSIG LOGIBI PRESIG
DTSQ   / USER   / ACCEL  ACDCP  CHKCON CRIDIS GAPACC HOLDSP PREDTV RUSERD
                  UNBIAS
DUTOL  / USER   / BLKDAT CHGMLN CRIDIS INTLOG LOGIN  LOGOUT RDVPRD SIGRES
DV     =        = PREDTV
DVFACT =        = LCHGEO
DVH    =        = CHKCON
DVM    =        = CHKCON CHKSDR
ENDLN  / ABIAS  / ACCEL  CARFOL CHGMLN GAPACC IBAP   INTERP LOGIN  OBAP
                  PREST1 SIGRES UNBIAS
EDM    =        = ACTSIG
ERRJUD =        = CHKCON CHKSDR
FACT   =        = CARFOL GAPACC LOGIN
FACTOR / LANECH / BLKDAT GAPACC INTLOG
FLENV  =        = IBAP
F3     =        = CHGMLN CRIDIS INTLOG
GAPLA  =        = GAPACC
GAPLE  =        = GAPACC
GAPMIN =        = GAPACC
I      =        = ABORTR ACTSIG ACTSTA ADDSTA ADLVAI CHGMLN CHKCON CHKSDR
                  CKLALT CLRCON EXTIME EXTRAC FIND   IBAP   INITAL INTERP
                  LOGIBI LOGIC  LOGIN  OBAP   PATHF  PSTATS RCAMSD RDVPRD
                  REPACK RGEOPD RLOOPD RPHASD SETCON SIGRES SIMPRO STORE
                  UNSETC
IA     / INDEX  / BANGS  CHGMLN GAPACC IBAP   LOGIBI LOGIN  OBAP   PATHF
                  RGEOPD
IACC   / VEHD   / BIAS   IBAP   OBAP   UNBIAS
IACDS  / VEHD   / ACDCP  CHKCON
IACLDS / VEHD   / ACDCP
IAFORM =        = BANGS
IALEFT / APPRO  / ADLVAI RGEOPD
IAMAX  =        = RDVPRD
IAN    / INDEX  / CHKCON IBAP   LOGIBI LOGIN  OBAP   RGEOPD SSIBAP SUMARY
IANDOR / PHASES / ACTSTA CHKDFP RPHASD
IAPRT  =        = IBAP   OBAP
IARRPH / SIGCAM / ACTSIG BLKDAT RPHASD
IAT    =        = EXTRAC FIND   REPACK SIMPRO STORE
IAT1   =        = BLKDAT
IAT2   =        = BLKDAT
IAT3   =        = BLKDAT
```

```
IAT4   =          = BLKDAT
IB     =          = LOGIN  QUEUE
IBA    =          = EXTRAC FIND   REPACK STORE
IBAPS  / VEHF     / BANGS  LOGIN  LOGOUT SSINTR
IBF    =          = LOGIC
IBLN   / LANE     / BANGS  CHGMLN IBAP   INFLZN LOGIN  RGEOPD
IBLNK1 =          = RCAMSD RLOOPD RPHASD RUSERD
IBUF   / QUE      / LOGIN  QUEUE  RDVPRD
IBUF1  =          = EXEC
IBUF2  =          = EXEC
IBUF3  =          = EXEC
IBUF4  =          = EXEC
IC     =          = ABORTR
ICAMPC / SIGCAM / ACTSIG BANGS  GAPACC IBAP   INFLZN INITAL LOGIN  PRESIG
                  RCAMSD RPHASD
ICAMPH / SIGCAM / PRESIG RCAMSD RPHASD
ICAMPO / SIGCAM / ACTSIG IBAP   PRESIG RCAMSD RPHASD
ICAMPS / PHASES / ACTSIG RPHASD
ICANSE / SDR    / ABORTR RGEOPD
ICDFS  / VEHD   / ACDCP  CHGMLN GAPACC
ICHKCF / VEHIL  / INTLOG
ICHS   =          = ABORTR
ICOM1  =          = INITAL
ICOM2  =          = INITAL
ICOM3  =          = INITAL
ICONA  / CONFLT / CHKSDR RGEOPD
ICONAN / CONFLT / CHKCON RGEOPD
ICOND  / CONFLT / CHKCON CHKSDR CLRCON RGEOPD
ICONI  / CONFLT / CLRCON RGEOPD SETCON UNSETC
ICONP  / CONFLT / ABORTR CHKCON CHKSDR CLRCON RGEOPD SETCON UNSETC
ICONP1 =          = LSTOP
ICONP2 =          = LSTOP
ICONTN / VEHIL  / INTLOG
ICONTR / INTER  / ACTSTA ADLVAI EXEC   GAPACC IBAP   INFLZN INITAL LSTOP
                  RCAMSD RGEOPD RPHASD RUSERD SUMARY
ICONUP / INDEX  / BLKDAT CHKCON CHKSDR CLRCON SETCON UNSETC
ICONV  / CONFLT / CHKCON CLRCON SETCON UNSETC
ICPHAS / SIGCAM / ABORTR ACTSIG PRESIG RCAMSD RPHASD SETLDF
ICPSET / PATH   / CHKCON CLRCON
ID     =          = EXTRAC REPACK RLOOPD
IDCHAR =          = RDVPRD
IDEDIC / VEHIL  / INTLOG
IDESPD =          = BANGS  IBAP   INTERP LOGIN  OBAP
IDFP   =          = ACTSIG CHKDFP
IDMAX  =          = RDVPRD
IDOG   =          = ACTSIG
IDOR   =          = ACTSIG
IDRICL / VEHF   / ABORTR ACCEL  ACDCP  ADLVAI BANGS  CARFOL CHGMLN CHKCON
                  CHKDSP CHKSDR CRIDIS DELAY  GAPACC IBAP   INTERP INTLOG
                  LCHDES LCHGEO LOGIN  LSTOP  OBAP   SETPTV SIGRES
IDTS   / VEHD   / LOGOUT NEWVEL UNBIAS
IDUALL / PHASES / ACTSIG ACTSTA RPHASD
IDVS   / VEHD   / LOGIN  LOGOUT SSIBAP SSINTR SSOBAP
IDW    =          = LOGIC
IDX    =          = RGEOPD
IE     =          = FIND   REPACK SMEP   STORE
IEA    =          = EXTRAC REPACK
IEF    / QUE    / LOGIC  LOGIN  RDVPRD
IEN    =          = BLKDAT EXTRAC FIND   LOGIC  REPACK STORE
IENCE  =          = RLOOPD
IENT1  =          = IBAP   OBAP   RGEOPD
IENT2  =          = CLRCON RGEOPD
IENT3  =          = IBAP   OBAP   RGEOPD
IENT4  =          = INTERP RGEOPD
IENT5  =          = RGEOPD
IENT6  =          = CHGMLN CHKCON IBAP   INTERP LOGIN  OBAP   SIGRES
IENT7  =          = CHGMLN IBAP   INTERP LOGIN  OBAP
IENT8  =          = IBAP   LOGIN
IERROR / VEHIL  / INTLOG SMEP
IEXTIM / VEHF   / LOGIN  LOGOUT
IFET1  =          = EXEC
IFET2  =          = EXEC
IFET3  =          = EXEC
IFET4  =          = EXEC
IFIX   =          = ACDCP  BIAS   CHKSDR PVAPRT RDVPRD
IFORCE =          = PATHF
IFORM  =          = PVAPRT
IFU    =          = BLKDAT LOGIC  SIMPRO
IFVA   / VEHD   / ACDCP  CHGMLN CHKCON CHKSDR SIGRES
IFVL   / LANE   / CHGMLN IBAP   LOGIBI LOGIN  LOGIOB LOGOUT OBAP
IFVP   / PATH   / INTERP LOGIBI LOGIOB
IFW    =          = EXTRAC FIND   LOGIC  REPACK STORE
IGEOCP / PATH   / CHKCON CHKSDR CLRCON LSTOP  RGEOPD SETCON UNSETC
IGEOP  / USER   / BLKDAT INITAL RGEOPD
IGO    / SIGCAM / ACDCP  CHGMLN CRIDIS IBAP   INTERP OBAP   SIGRES
IHPRT  =          = IBAP   INTERP OBAP
II     =          = ACTSIG INTSTA PATHF  RCAMSD RLOOPD SUMARY
IIAT   =          = EXTRAC FIND   REPACK STORE
IIEN   =          = EXTRAC FIND   LOGIC  REPACK STORE
IIFU   =          = LOGIC
IISIGN =          = RCAMSD
IITURN =          = RCAMSD
IIV    =          = LOGIC
IIW    =          = LOGIC
IK     =          = CLRCON
IL     / INDEX  / IBAP   LOGIN  LOGOUT OBAP   PATHF
ILANE  =          = PATHF
ILCH   / PATH   / RGEOPD
ILD    =          = CHKDFP SETLDF
ILDL   =          = CHKLDT
ILDLN  =          = RLOOPD
ILETTA =          = RCAMSD
ILETTN =          = RCAMSD
ILETTS =          = RCAMSD
ILETTU =          = RCAMSD
ILN    / INDEX  / BANGS  CHGMLN IBAP   LOGIBI LOGIN  LOGIOB LOGOUT OBAP
                  SSIBAP
ILPRT  =          = IBAP   OBAP
ILSTOP / VEHIL  / INTLOG
ILUNC  / VEHIL  / INTLOG
ILVL   / LANE   / CHGMLN LOGIBI LOGIN  LOGIOB LOGOUT
ILVP   / PATH   / CHKCON CHKSDR LOGIBI LOGIOB
ILW    =          = REPACK
ILYELD / VEHIL  / INTLOG
IMINOR / PHASES / ACTSIG ACTSTA RPHASD
IN     =          = EXTRAC FIND   LOGIC  REPACK SMEP   STORE
INDEX  =          = ADDSTA CHKCON CHKSDR LOGOUT LSTOP  PSTATS
INFLZ  / VEHIL  / INTLOG
ININT  =          = PREST1
INO    =          = ACTSIG RPHASD RUSERD
INOT   =          = REPACK STORE
INOW   =          = SETCON
INPUT  / USER   / BLKDAT INITAL RCAMSD RGEOPD RLOOPD RPHASD RUSERD
INQUE  =          = IBAP   SSIBAP
INTER  =          = ACTSIG
IOFF   =          = ACTSIG RPHASD
ION    =          = CHKDFP RPHASD
IONE   =          = IBAP   LOGIN  OBAP
IOPHAS =          = ACTSIG
IOPT   / PATH   / RGEOPD
IOUT   =          = TIMSTA
IP     / INDEX  / BANGS  CHKDFP CLRCON INTERP
IPAGE  =          = EXEC   INTSTA
IPAP   / USER   / RUSERD SUMARY
IPATH  =          = CKLALT
IPCLTO =          = ACTSIG
IPENTC =          = DELAY
IPFLAG / INDEX  / ACCEL  ACDCP  BLKDAT CARFOL CRIDIS IBAP   INTERP LOGIN
                  OBAP   PREST1
IPFURM =          = BANGS
IPHTIM =          = RCAMSD
IPIJR  / CLASS  / ABORTR ACDCP  RDVPRD SIGRES
```

```
IPNDEX =          = CHKSDR
IPOLL  / USER     / RUSERD
IPOS   / VEHD     / BIAS   IBAP   OBAP   SETCON SVEHU  UNBIAS
IPOSCK =          = SETCON
IPOSF  =          = SVEHU
IPOSR  =          = SVEHU
IPOSRB =          = CLRCON
IPPRT  =          = INTERP
IPRES  =          = RLOOPD
IPRINT =          = PSTATS
IPRTLO / VEHF     / ADLVAI CHKCON CHKLDT CHKSDR CLRCON GAPACC IBAP   INTERP
                    LOGIN  LOGOUT OBAP   SETCON UNSETC
IPRTM  / VEHD     / ACDCP  BANGS  CHGMLN CHKCON CRIDIS IBAP   INTERP LOGIN
                    LSTOP  NEWVEL OBAP   PREST2 SIGRES
IPT    / PATH     / RGEOPD
IPTC   / USER     / RUSERD SUMARY
IPTHUP / INDEX    / BLKDAT CHKCON CHKSDR CLRCON INTERP LOGIBI LOGIOB LSTOP
                    SETCON UNSETC
IPTURN =          = INTSTA SUMARY
IPULS  =          = CHKLDT RLOOPD
IPUNCH / USER     / ACTSTA PSTATS RUSERD SUMARY
IQ     / QUE      / BANGS  BLKDAT IBAP   INTERP LOGIBI LOGIN  LOGOUT OBAP
                    PVAPRT
IQACC  =          = PVAPRT
IQDS   / VEHD     / LOGOUT SSIBAP SSINTR
IQF    / QUE      / BLKDAT EXEC   LOGIN  RDVPRD
IQPOS  =          = PVAPRT
IQQ    =          = LOGIN
IQV    =          = PVAPRT
IQVEL  =          = PVAPRT
IR     =          = FIND   REPACK SMEP   STORE
IREC   / PHASES   / ACTSTA CHKDFP RPHASD
IRECAD =          = ABORTR
IREPFX / INDEX    / CHGMLN IBAP   INTERP LOGIBI LOGIOB OBAP   PREST1
IREPIL / INDEX    / ACDCP  IBAP   INFLZN INTLOG SIGRES
IRET   =          = EXEC
IRMIN  / CLASS    / RDVPRD
IRN    =          = ABORTR IBAP   INTERP OBAP
IRNAME / RUTINE   / ABORTR ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   CARFOL
                    CHGMLN CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR
                    CKLALT CLRCON CRIDIS DELAY  ENDLCH EXEC   EXTIME FLGNOR
                    GAPACC HOLDSP IBAP   INFLZN INITAL INTERP INTLOG INTSTA
                    LCHDES LCHGEO LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB
                    LSTOP  NEWVEL OBAP   PATHF  PREDTV PRESIG PREST1 PREST2
                    PVAPRT QUEUE  RCAMSD RDVPRD RGEOPD RLOOPD RPHASD RUSERD
                    SETCON SETLDF SETPTV SIGRES SSIBAP SSINTR SSOBAP SVEHU
                    UNBIAS UNSETC
IRSTOP / VEHD     / ACDCP
IS     =          = EXTRAC FIND   LOGIC  REPACK SIMPRO STORE
ISAME  =          = BANGS
ISDEC  / VEHD     / ACDCP  CHGMLN SIGRES
ISDR   =          = CHKSDR
ISDRA  / APPRO    / CHKSDR RGEOPD
ISDRN  / APPRO    / RGEOPD
ISDS   / VEHD     / LOGOUT SSIBAP SSINTR
ISET   / VEHD     / BANGS  CHKMLN DELAY  ENDLCH GAPACC IBAP   INTERP INTLOG
                    LCHDES LOGIBI LOGIN  LOGIOB NEWVEL OBAP   PATHF  SVEHU
ISIDE  / LANECH   / CHGMLN DELAY  GAPACC LCHDES SVEHU  UNBIAS
ISIDW  =          = LOGIC
ISIG   =          = BANGS
ISIIW  =          = LOGIC
ISISET / SIGCAM   / BANGS  GAPACC IBAP   INFLZN LOGIN  RCAMSD
ISKP   / PHASES   / ACTSIG ACTSTA RPHASD
ISLIM  / APPRO    / CHKDSP LOGIBI LOGIOB RGEOPD SETPTV SSIBAP
ISLP   / VEHD     / ABORTR BIAS   IBAP   INTERP LOGIN  OBAP   UNBIAS
ISNA   / LANE     / LOGIBI LOGIN  LOGIOB LOGOUT RGEOPD
ISNAME =          = EXTRAC FIND   LOGIC  REPACK SMEP   STORE
ISPD   / VEHF     / CARFOL CHKDSP LOGIBI LOGIN  LOGIOB SETPTV SSIBAP SSINTR
                    SSOBAP UNBIAS
ISPDP  / VEHD     / CHKDSP IBAP   LOGIBI SETPTV SSIBAP
ISPDS  / VEHD     / LOGOUT SSIBAP SSINTR SSOBAP
```

```
IST    =          = ACTSTA
ISTATS =          = RUSERD
ISTCON / VEHD     / BANGS  CLRCON INTERP LOGIBI
ISTMO  / VEHD     / ACDCP
ISVAL  =          = RCAMSD
IT     =          = REPACK STORE
ITC    =          = SUMARY
ITEST  =          = RGEOPD RPHASD
ITIM   =          = EXEC
ITIME  =          = ABORTR
ITIMV  / VEHD     / LOGOUT SSIBAP SSINTR SSOBAP
ITITLE / TITLE    / ACTSTA EXEC   INITAL INTSTA RCAMSD RGEOPD RLOOPD RPHASD
                    RUSERD SUMARY TIMSTA
ITM    =          = EXTIME
ITNOW  =          = EXEC
ITURN  / VEHF     / ACDCP  BANGS  DELAY  INFLZN LOGIN  LOGOUT PATHF  SIGRES
ITWO   =          = INTERP
ITYPE  =          = CHKDFP
ITYPLD / LOOPS    / CHKLDT RLOOPD
IT1    =          = RGEOPD RLOOPD RPHASD
IT2    =          = RGEOPD
IUSED  =          = RLOOPD RPHASD
IV     / INDEX    / ABORTR ADLVAI BANGS  CHGMLN CHKCON CHKDSP CHKLDT CHKSDR
                    FIND   IBAP   INTERP LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI
                    LOKIOB LSTOP  OBAP   PATHF  PREST1 PREST2 PVAPRT REPACK
                    SETCON SMEP   STORE  UNSETC
IVATIN =          = ADLVAI LSTOP
IVCHAR =          = RDVPRD
IVCONF =          = CHKCON
IVEHCL / VEHF     / ACCEL  BANGS  BIAS   CARFOL CHGMLN CHKCON CHKLDT CHKSDR
                    CLRCON CRIDIS GAPACC IBAP   INTERP INTLOG LCHDES LCHGEO
                    LOGIN  OBAP   SETPTV SIGRES
IVEHP  / USER     / BLKDAT INITAL LOGIN  RDVPRD
IVEL   / VEHD     / BIAS   IBAP   OBAP   SVEHU  UNBIAS
IVMAX  =          = RDVPRD
IVMAXA / VEHD     / BIAS   LOGOUT
IVMAXD / VEHD     / ACDCP  BIAS   LOGOUT
IVN    / INDEX    / DELAY  IBAP   INTERP OBAP   SSIBAP
IVPV   / INDEX    / BANGS  CHGMLN CHKCON CHKSDR LOKIBI LOKIOB PREST1
IWD    =          = EXTRAC FIND   REPACK STORE
IWHERE =          = BANGS
IWIA   =          = PSTATS
IWTC   =          = PSTATS
IX     =          = REPACK STORE
IXXX   =          = RUSERD
IY     =          = EXTRAC FIND   LOGIC  REPACK SMEP   STORE
IYES   =          = ACTSIG RPHASD RUSERD
IZ     =          = LOGIN
IZERO  =          = INTERP OBAP
I1TO3  =          = INTSTA
J      =          = ACTSTA ADDSTA ADLVAI CHKCON CHKSDR CLRCON LOGIBI LOGIN
                    PSTATS RCAMSD RDVPRD RLOOPD RPHASD SETCON SIMPRO UNSETC
JA     =          = BANGS  CHKSDR QUEUE  RGEOPD SUMARY
JACC   =          = CHKCON FLGNOR GAPACC LOKIBI LOKIOB PREST1
JAN    =          = QUEUE
JAND   =          = CHKDFP RPHASD
JBAPS  =          = BANGS
JBLN   =          = CHGMLN GAPACC RCAMSD
JCANSE =          = CHKSDR
JCONI  =          = CLRCON SETCON UNSETC
JCPSET =          = SETCON
JD     =          = CHKCON PREDTV SETPTV
JDRICL =          = BANGS
JFINL  =          = IBAP   INTERP
JFVA   =          = CHKCON
JGEOCP =          = CLRCON SETCON UNSETC
JGO    =          = CHGMLN IBAP
JJ     =          = ACTSIG RCAMSD RLOOPD
JL     =          = BANGS  CHKCON CHKSDR RLOOPD SSINTR
JLCH   =          = CKLALT DELAY  INFLZN LCHDES SIGRES
JLD    =          = CHKDFP SETLDF
```

```
JLDL   =        = CHKLDT
JLN    =        = BANGS  CHGMLN QUEUE  SSINTR
JNDEX  =        = CHKCON CHKSDR LSTOP
JOPT   =        = PATHF
JP     =        = BANGS  CHKCON CHKSDR CLRCON RPHASD SETCON UNSETC
JPFLAG / INDEX  / ACCEL  ACDCP  BLKDAT CARFOL CRIDIS IBAP   INTERP LOGIN
                  OBAP   PREST1
JPOS   =        = BANGS  CHKCON LOGIOB LOKIBI LOKIOB PREST1 SETCON
JPP1   =        = RPHASD
JPP2   =        = RPHASD
JPRTM  / INDEX  / ACDCP  CHGMLN CHKCON IBAP   INTERP LSTOP  NEWVEL OBAP
                  PREST2 SIGRES
JPT    =        = PATHF
JRECAD =        = ABORTR
JSDRA  =        = CHKSDR
JSET   =        = BANGS  CHGMLN GAPACC LCHDES
JSIG   =        = BANGS
JSISET =        = GAPACC IBAP   INFLZN SIGRES
JSLIM  =        = CHKCON CHKSDR PREDTV SETPTV
JSLP   =        = BANGS  CHKCON
JSNA   =        = ADLVAI CHKCON SSINTR
JSPD   =        = BANGS  CHKCON CHKSDR PREDTV SETPTV
JSPDP  =        = CHKCON PREDTV SETPTV
JSTCON =        = BANGS
JTITLE =        = INITAL RUSERD
JTURN  =        = BANGS  DELAY  SIGRES
JV     =        = ADLVAI CHKCON LSTOP  PREDTV SETPTV
JVEHCL =        = BANGS  GAPACC LOKIBI LOKIOB PREST1
JVEL   =        = CHGMLN CHKCON CHKSDR LOGIBI LOGIOB LOKIBI LOKIOB PREST1
JXXX   =        = RUSERD
K      =        = ACDCP  ADDSTA CRIDIS LOGIN  RCAMSD RGEOPD RLOOPD RPHASD
KCANSE =        = CHKSDR
KK     =        = INTSTA RCAMSD
KOUNT  =        = CHKCON
KPFLAG / INDEX  / ACDCP  BANGS  BLKDAT CARFOL CRIDIS IBAP   INTERP LOGIN
                  LSTOP  NEWVEL OBAP   PREST1 SIGRES
KPRTM  =        = BANGS  CHKCON HOLDSP
KSISET =        = IBAP   SIGRES
KSPD   =        = CHKCON CHKSDR
KTITLE =        = INITAL RUSERD
LAGR   =        = DELAY
LAGSPD / LANECH / DELAY  GAPACC SVEHU  UNBIAS
LALT   / VEHD   / CHGMLN CKLALT DELAY  IBAP   LCHDES LOGIN  LOGIOB OBAP
LANE   =        = SUMARY
LANESS =        = RCAMSD
LANSI  =        = CHKLSI DELAY  GAPACC LCHDES SVEHU
LATNOW =        = CARFOL
LATPOS / VEHD   / BANGS  CARFOL CHGMLN ENDLCH IBAP   LCHGEO LOGIBI LOGIN
                  LOGIOB OBAP   SETCON SIGRES
LAT2GO =        = CARFOL
LB     =        = CHKLSI
LCHGE  / VEHD   / ADLVAI BANGS  CARFOL CHGMLN CHKMLN ENDLCH IBAP   INTERP
                  LOGIN  OBAP   PATHF  PREST1 SETCON
LCNTRI =        = RGEOPD
LCONTR / LANE   / ACDCP  CHGMLN CHKCON CHKSDR IBAP   INFLZN RGEOPD
LDA    =        = RLOOPD
LDSTOP =        = RLOOPD
LDSTRT =        = RLOOPD
LDTRIP / LOOPS  / ACTSIG CHKDFP CHKLDT RLOOPD SETLDF
LE     =        = CHKLSI
LEADR  =        = DELAY
LEADSP / LANECH / DELAY  GAPACC LCHDES SVEHU  UNBIAS
LEGAL  / VEHD   / BANGS  CARFOL CHGMLN CHKMLN ENDLCH GAPACC IBAP   INTERP
                  LCHDES LCHGEO LOGIN  LOGIOB OBAP   PATHF
LEGAP  =        = GAPACC
LENP   / PATH   / ABORTR CLRCON INTERP LOGIOB LOKIOB RGEOPD
LENV   / CLASS  / BIAS   CHKCON CHKLDT CHKSDR CLRCON GAPACC IBAP   INTLOG
                  LCHGEO LOKIBI LOKIOB OBAP   PREST1 RDVPRD
LFALSE =        = ACDCP  ACTSIG BANGS  BIAS   BLKDAT CARFOL CHGMLN CHKCON
                  CHKMLN CHKSDR FLGNOR GAPACC IBAP   INFLZN INTERP INTLOG
                  LOGIBI LOGIN  LOGIOB LOKIBI LOKIOB LSTOP  NEWVEL OBAP
                  PREST1 PREST2 RDVPRD SETCON SIGRES
LFORCE =        = PATHF
LGEOM  / LANE   / CHKMLN IBAP   LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI OBAP
                  PREST1 RGEOPD SETPTV SVEHU
LGEOM1 =        = CHKLSI LOKIBI LOKIOB
LGEOM2 =        = CHGMLN CHKLSI
LGEOM3 =        = CHKLSI RLOOPD
LGEOM4 =        = CHGMLN CHKCON CHKLSI CHKSDR PREDTV RLOOPD SETPTV SVEHU
LIBA   / INTER  / IBAP   INTSTA LOGOUT RGEOPD SSINTR SUMARY
LIBAR  / INTER  / BLKDAT CHKCON QUEUE  RGEOPD RLOOPD
LIBL   / PATH   / RGEOPD SSINTR
LIMP   / PATH   / LOGIBI LOGIOB RGEOPD SETPTV
LINTP  / LANE   / PATHF  RGEOPD
LLANES / APPRO  / IBAP   LOGIOB OBAP   PATHF  RGEOPD
LLD    / PHASES / ACTSIG ACTSTA CHKDFP RLOOPD RPHASD SETLDF
LLDL   / LANE   / CHKLDT
LLDLN  =        = RLOOPD
LLOOPS / LOOPS  / RLOOPD
LNEXT  / VEHF   / BANGS  CHGMLN CHKCON CHKDSP CHKSDR DELAY  ENDLCH IBAP
                  INFLZN INTERP INTLOG LCHDES LOGIBI LOGIN  LOGIOB LOKIBI
                  LOKIOB LSTOP  OBAP   PATHF  SETCON SIGRES SSIBAP UNSETC
LOBA   / INTER  / OBAP   RGEOPD
LOBAP  / PATH   / RGEOPD
LOBAR  / INTER  / BLKDAT RGEOPD
LOBL   / PATH   / LOGIBI RGEOPD
LOGFLG / VEHD   / BANGS  CHGMLN IBAP   INFLZN LOGIN  OBAP   SIGRES
LOGTMP / INDEX  / ACDCP  CHGMLN CHKCON CHKSDR IBAP   INFLZN INTLOG LOGIBI
                  LSTOP  SETCON SIGRES UNSETC
LOK    =        = CHKLSI DELAY  LCHDES
LPATH  =        = PATHF
LPHASE / PHASES / ACTSIG RLOOPD RPHASD
LPHNXT / PHASES / ACTSIG ACTSTA RPHASD
LPRES  / VEHF   / CHGMLN CHKCON LOGIBI LOGIN  LOGIOB PATHF
LPREV  =        = LOGIBI
LPRTM  =        = HOLDSP
LQ     / QUE    / BLKDAT IBAP   LOGIN  QUEUE
LQUEUE / SUMSTA / SSIBAP SSINTR SUMARY
LTF    =        = CHGMLN FLGNOR LOGIC
LTRUE  =        = ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   BLKDAT CHGMLN
                  CHKCON CHKMLN CHKSDR CRIDIS FLGNOR GAPACC IBAP   INFLZN
                  INTERP INTLOG LCHDES LOGIBI LOGIN  LOGIOB LOGOUT LSTOP
                  OBAP   PATHF  PREST1 PREST2 RDVPRD SETCON SIGRES
LTURN  / LANE   / RGEOPD
LVATIN / INTER  / ADLVAI LOGIBI LSTOP
LWID   / LANE   / ABORTR CHGMLN IBAP   OBAP   RGEOPD
MAGSAT =        = ACTSIG
MATPOS =        = BANGS
MATSTL / VEHIL  / ACDCP  CHKCON CHKSDR INTLOG LSTOP
MAXLOG =        = CHKSDR
                  PVAPRT RDVPRD RGEOPD SSIBAP UNSETC
MBLOCK / VEHD   / ACDCP  CHGMLN CHKMLN IBAP   LOGIN  OBAP   PREST1
MCAM   =        = RPHASD
MCHGE  =        = BANGS  ENDLCH LOGIN
MCHKCF / VEHIL  / CHKCON INFLZN SIGRES
MCONTR =        = CHGMLN GAPACC RCAMSD
MCPSET =        = CLRCON LSTOP  SETCON UNSETC
MDEDIC / VEHIL  / ABORTR IBAP   INTLOG LOGIN
MEGAL  =        = BANGS  CHGMLN SVEHU
MEMP   =        = LOKIBI
MENP   =        = LOKIBI
MFINL  / VEHD   / ACCEL  ACDCP  CHGMLN CHKMLN IBAP   INTERP INTLOG LCHDES
                  LOGIBI LOGIN  LOGIOB LOKIBI LOKIOB OBAP   PREST1 SIGRES
MGEOM4 =        = CHKCON
MIBA   =        = INTSTA
MIMP   =        = CHKCON CHKDSP PREDTV SETPTV SSIBAP
MINFLZ / VEHIL  / CHGMLN IBAP   INFLZN
MININT / VEHD   / ACDCP  BANGS  LOGIBI LOGIN  LOGIOB
MIUNC  / VEHIL  / CHKCON CHKSDR INFLZN
MLAG   / VEHD   / CARFOL LOGIN
MLANES =        = BANGS  RLOOPD SSINTR
MLRTOR / VEHIL  / ACDCP  SIGRES
```

```
MLSTOP / VEHIL  / INFLZN
MLUNC  / VEHIL  / CHKCON CHKSDR INFLZN
MLYELD / VEHIL  / INFLZN
MNEXT  =        = BANGS  LSTOP
MNVSY  / SUMSTA / BLKDAT EXEC   TIMSTA
MOASF  / VEHD   / CHGMLN LOGIBI LOGIN  LOGIOB PREST2 SIGRES
MOBAP  =        = CKLALT PATHF
MOBAPD =        = BANGS
MOBL   =        = LOKIBI
MOF    =        = BANGS
MOGFLG =        = BANGS  LOGIBI LSTOP  SETCON
MOR    =        = BANGS  CHKCON
MORC   =        = BANGS  CHKCON SETCON UNSETC
MPINT  =        = CKLALT PATHF
MPLVDV =        = SUMARY
MPOBS  / VEHD   / LOGIN
MPRES  =        = ADLVAI BANGS
MPRO   / VEHD   / ACCEL  ADLVAI CHKCON CHKSDR IBAP   INFLZN INTLOG LOGIN
                  LSTOP  SIGRES
MQUEUE / SUMSTA / SSIBAP SUMARY
MSAOR  / VEHD   / ACDCP  BIAS   LOGIN
MSDR   =        = CHKSDR
MSFLG  / VEHD   / ACDCP  BANGS  CHGMLN CHKCON CRIDIS INFLZN LOGIN  LOGIOB
                  NEWVEL SIGRES
MSG    =        = ABORTR EXEC
MSGPP  =        = ABORTR
MSGR   / RUTINE / ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   BLKDAT CARFOL
                  CHGMLN CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR
                  CKLALT CLRCON CRIDIS DELAY  ENDLCH EXTIME FLGNOR GAPACC
                  HOLDSP INFLZN INITAL INTLOG INTSTA LCHDES LCHGEO LOGIBI
                  LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB LSTOP  NEWVEL PATHF
                  PREDTV PRESIG PREST1 PREST2 PVAPRT QUEUE  RCAMSD RDVPRD
                  RGEOPD RLOOPD RPHASD RUSERD SETCON SETLDF SETPTV SIGRES
                  SSIBAP SSINTR SSOBAP SVEHU  UNBIAS UNSETC
MSG1   =        = EXEC
MSG2   =        = EXEC
MSG3   =        = EXEC
MSG4   =        = EXEC
MSG901 =        = SSINTR
MSG902 =        = LOGIOB
MSG903 =        = LCHDES
MSG904 =        = LCHDES
MSG905 =        = LCHDES
MSG906 =        = ACDCP
MSG907 =        = ACDCP
MSG908 =        = ADLVAI
MSG909 =        = ADLVAI
MSG910 =        = INTLOG
MSG911 =        = INTLOG
MSG912 =        = SIGRES
MSG913 =        = CHKCON
MSG914 =        = SETCON
MSG915 =        = INFLZN
MSG916 =        = PATHF
MSG917 =        = CHKMLN
MSG918 =        = CHKMLN
MSG919 =        = BANGS
MSG920 =        = LOGIN
MSG921 =        = ACTSIG
MSSGRN / VEHIL  / INFLZN SIGRES
MSSRED / VEHIL  / SIGRES
MSTPF  / VEHD   / ACDCP  BANGS  BIAS   LOGIN
MTCARS / VEHD   / ACDCP  CHKCON INFLZN INTLOG LOGIN  SIGRES
MWID   =        = CHGMLN
N      =        = ACTSTA RLOOPD RPHASD
NAP    =        = RGEOPD
NBANG  / SUMSTA / BANGS  SUMARY
NBITS  =        = EXTRAC FIND
NCAM   =        = RPHASD
NCAMSP / SIGCAM / ABORTR PRESIG RCAMSD RPHASD
NCHS   =        = ABORTR
NCOM01 =        = ABORTR
NCOM02 =        = ABORTR
NCOM03 =        = ABORTR
NCOM04 =        = ABORTR
NCOM05 / 123556 / ABORTR
NCOM06 =        = ABORTR
NCOM07 =        = ABORTR
NCOM08 =        = ABORTR
NCOM09 =        = ABORTR
NCOM10 =        = ABORTR
NCOM11 =        = ABORTR
NCOM12 =        = ABORTR
NCOM13 =        = ABORTR
NCOM14 =        = ABORTR
NCOM15 =        = ABORTR
NCOM16 =        = ABORTR
NCPSET / PATH   / CHKCON CLRCON
NDMPH  / SUMSTA / ADDSTA LOGOUT PSTATS
NDRICL =        = RDVPRD
NELIM  / SUMSTA / LOGIN  SUMARY
NEWNOF =        = FLGNOR
NEXTPH =        = ACTSIG
NGAPO  / PHASES / ACTSIG ACTSTA RPHASD
NGEOCP / PATH   / ABORTR CHKCON CHKSDR CLRCON INTERP LSTOP  RGEOPD SETCON
                  UNSETC
NIBA   / INTER  / IBAP   INTSTA RGEOPD SUMARY
NIBL   / INTER  / RCAMSD RGEOPD
NINE   =        = SUMARY
NININT =        = BANGS  CHKCON SETCON
NLANES / APPRO  / ABORTR IBAP   LOGIOB OBAP   PATHF  RGEOPD SIMPRO
NLC    =        = RCAMSD
NLD    / PHASES / ACTSIG ACTSTA CHKDFP RLOOPD RPHASD SETLDF
NLDL   / LANE   / CHKLDT IBAP   RLOOPD
NLDLN  =        = RLOOPD
NLL    / LANE   / CHKMLN CKLALT DELAY  LCHDES LOGIOB RGEOPD SVEHU
NLOOPS / LOOPS  / INITAL RLOOPD RPHASD
NLR    / LANE   / CHKMLN CKLALT DELAY  LCHDES LOGIOB RGEOPD SVEHU
NLUNC  =        = LSTOP
NLVDV  / SUMSTA / LOGIN  SUMARY
NMAXO  / PHASES / ACTSIG ACTSTA RPHASD
NN     =        = ACTSTA RPHASD
NN1    =        = PATHF
NN2    =        = PATHF
NOASF  =        = CHGMLN
NOATTB / NOATTB / ABORTR BLKDAT CLRCON IBAP   INTERP LOGIN  OBAP   RGEOPD
NOBA   / INTER  / OBAP   RGEOPD
NOBAPD / VEHF   / BANGS  CHGMLN CKLALT GAPACC IBAP   INTERP LOGIN  OBAP
                  PATHF
NOCONF / INTER  / RGEOPD
NOF    / VEHF   / BANGS  CARFOL CHGMLN DELAY  ENDLCH IBAP   INTERP INTLOG
                  LOGIBI LOGIN  LOGIOB OBAP   PREST1 SIGRES
NOFC   =        = CHKCON SETCON UNSETC
NOQ    =        = DELAY  LCHDES SVEHU
NOR    / VEHF   / BANGS  CHGMLN ENDLCH FLGNOR IBAP   INTERP LOGIBI LOGIN
                  LOGIOB LOGOUT OBAP   SETCON
NORC   / VEHD   / BANGS  CLRCON IBAP   INTERP LOGIN  OBAP   SETCON UNSETC
NORF   =        = DELAY
NORR   =        = DELAY
NOSF   / LANECH / CHGMLN DELAY  GAPACC LCHDES SVEHU
NOSR   / LANECH / CHGMLN DELAY  GAPACC SVEHU
NPATHS / INTER  / INTERP RGEOPD
NPCLTO =        = ACTSIG
NPHASE / PHASES / ACTSIG ACTSTA INITAL RLOOPD RPHASD SUMARY
NPHNXT / PHASES / ACTSIG ACTSTA RPHASD
NPINT  / LANF   / PATHF  RGEOPD
NPRO   =        = INTLOG LSTOP  SIGRES
NQA    =        = IBAP
NQD    / SUMSTA / ADDSTA LOGOUT PSTATS
NR     / RUTINE / ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   BLKDAT CARFOL
                  CHGMLN CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR
                  CKLALT CLRCON CRIDIS DELAY  ENDLCH EXTIME FLGNOR GAPACC
```

```
                         HOLDSP INFLZN INITAL INTLOG INTSTA LCHDES LCHGEO LOGIBI
                         LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB LSTOP  NEWVEL PATHF
                         PREDTV PRESIG PREST1 PREST2 PVAPRT QUEUE  RCAMSD RDVPRD
                         RGEOPD RLOOPD RPHASD RUSERD SETCON SETLDF SETPTV SIGRES
                         SSIBAP SSINTR SSOBAP SVEHU  UNBIAS UNSETC
NRECAD =               = EXEC
NRLAN  / INTER  /        RCAMSD RGEOPD
NRNAME / RUTINE / ABORTR ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   CARFOL
                         CHGMLN CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR
                         CKLALT CLRCON CRIDIS DELAY  ENDLCH EXEC   EXTIME FLGNOR
                         GAPACC HOLDSP IBAP   INFLZN INITAL INTERP INTLOG INTSTA
                         LCHDES LCHGEO LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB
                         LSTOP  NEWVEL OBAP   PATHF  PREDTV PRESIG PREST1 PREST2
                         PVAPRT QUEUE  RCAMSD RDVPRD RGEOPD RLOOPD RPHASD RUSERD
                         SETCON SETLDF SETPTV SIGRES SSIBAP SSINTR SSOBAP SVEHU
                         UNBIAS UNSETC
NRNAMM / RUTINE / ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   BLKDAT CARFOL
                         CHGMLN CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR
                         CKLALT CLRCON CRIDIS DELAY  ENDLCH EXTIME FLGNOR GAPACC
                         HOLDSP INFLZN INITAL INTLOG INTSTA LCHDES LCHGEO LOGIBI
                         LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB LSTOP  NEWVEL PATHF
                         PREDTV PRESIG PREST1 PREST2 PVAPRT QUEUE  RCAMSD RDVPRD
                         RGEOPD RLOOPD RPHASD RUSERD SETCON SETLDF SETPTV SIGRES
                         SSIBAP SSINTR SSOBAP SVEHU  UNBIAS UNSETC
NSD    / SUMSTA / ADDSTA LOGOUT PSTATS
NSDR   / APPRO  / CHKSDR RGEOPD
NSFLG  =        = FLGNOR
NSKP   =        = LOGIBI
NTD    / SUMSTA / ADDSTA LOGOUT PSTATS
NUM    =        = ABORTR CLRCON IBAP   INTERP INTSTA LOGIN  OBAP   PSTATS
                  RGEOPD
NUMC   =        = SUMARY
NUME   =        = SUMARY
NUMLD  =        = CHKDFP SETLDF
NUMPRO / SUMSTA / ADDSTA INTSTA LOGOUT PSTATS SUMARY TIMSTA
NUMPSU / SUMSTA / ABORTR LOGOUT SUMARY TIMSTA
NUMSDR / INTER  / RGEOPD
NUMTA  =        = SUMARY
NUMV   / QUE    / BLKDAT LOGIN  RDVPRD
NV     =        = IBAP   INTERP OBAP
NVATIN / INTER  / ADLVAI CHKSDR INITAL LOGIBI LSTOP
NVEHCL =        = RDVPRD
NVIA   / INTER  / IBAP   LOGIBI LOGIN  LOGIOB LOGOUT OBAP
NVIBA  / INTER  / EXEC   LOGIBI LOGIN
NVIL   / APPRO  / CHGMLN IBAP   LOGIBI LOGIN  LOGIOB LOGOUT OBAP
NVILL  =        = CHGMLN LOGIBI LOGIN  LOGIOB LOGOUT
NVIN   / INTER  / EXEC   LOGIBI LOGIOB
NVIP   / INTER  / INTERP LOGIBI LOGIOB
NVOBA  / INTER  / EXEC   LOGIOB LOGOUT
NVSY   / INTER  / EXEC   LOGIN  LOGOUT TIMSTA
NVSYA  / SUMSTA / EXEC   TIMSTA
NWDS   =        = ABORTR
NWE    =        = EXTRAC FIND   LOGIC  REPACK STORE
NXVEH  =        = IBAP   INTERP OBAP
NYES   =        = ACTSTA PSTATS SUMARY
N1     =        = ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   CARFOL CHGMLN
                  CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR CKLALT
                  CLRCON CRIDIS DELAY  ENDLCH EXEC   EXTIME FLGNOR GAPACC
                  HOLDSP IBAP   INFLZN INITAL INTERP INTLOG INTSTA LCHDES
                  LCHGEO LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB LSTOP
                  NEWVEL OBAP   PATHF  PREDTV PRESIG PREST1 PREST2 PVAPRT
                  QUEUE  RCAMSD RDVPRD RGEOPD RLOOPD RPHASD RUSERD SETCON
                  SETLDF SETPTV SIGRES SSIBAP SSINTR SSOBAP SVEHU  UNBIAS
                  UNSETC
N2     =        = ACCEL  ACDCP  ACTSIG ADLVAI BANGS  BIAS   CARFOL CHGMLN
                  CHKCON CHKDFP CHKDSP CHKLDT CHKLSI CHKMLN CHKSDR CKLALT
                  CLRCON CRIDIS DELAY  ENDLCH EXEC   EXTIME FLGNOR GAPACC
                  HOLDSP IBAP   INFLZN INITAL INTERP INTLOG INTSTA LCHDES
                  LCHGEO LOGIBI LOGIN  LOGIOB LOGOUT LOKIBI LOKIOB LSTOP
                  NEWVEL OBAP   PATHF  PREDTV PRESIG PREST1 PREST2 PVAPRT
                  QUEUE  RCAMSD RDVPRD RGEOPD RLOOPD RPHASD RUSERD SETCON
```

```
                  SETLDF SETPTV SIGRES SSIBAP SSINTR SSOBAP SVEHU  UNBIAS
                  UNSETC
OADMPH =        = PSTATS
OAQD   =        = PSTATS
OASD   =        = INTSTA PSTATS
OATD   =        = PSTATS
OLDACC =        = CRIDIS GAPACC LCHDES
OLDDTS / ABIAS  / LOGIN  NEWVEL UNBIAS
ONETRD =        = LOGIN
P      =        = CHKCON CHKSDR PREDTV
PDELAY =        = INTSTA
PDMPH  =        = PSTATS
PI     =        = LCHGEO
PIJR   / CLASS  / ACCEL  ADLVAI CARFOL CHGMLN CHKCON CHKSDR CRIDIS INTLOG
                  LSTOP  PREDTV RDVPRD
PIJRMI =        = RDVPRD
PLVDV  / SUMSTA / LOGIN  SUMARY
PN     =        = PREDTV
PO     =        = CHKCON CHKSDR PREDTV SETPTV
POS    =        = BANGS
POSCHK =        = CHKSDR IBAP   SSIBAP
POSL   =        = BANGS
POSLAT =        = BANGS  CHGMLN IBAP   INTERP LCHGEO LOGIN  OBAP   SETCON
POSNEW / ABIAS  / BANGS  BIAS   CHGMLN CHKLDT CHKLSI CHKMLN CHKSDR CLRCON
                  IBAP   INTERP INTLOG LOGIBI LOGIN  LOGIOB NEWVEL OBAP
                  PVAPRT SETCON SETPTV SSIBAP SSINTR
POSNRB =        = CHKLDT
POSOLD / ABIAS  / ACCEL  CARFOL CHGMLN CHKLDT CRIDIS GAPACC LCHDES LOGIBI
                  LOGIN  LOGOUT NEWVEL UNBIAS
POSORB =        = CHKLDT
POSTOT =        = LOGIBI
PQD    =        = PSTATS
PSD    =        = PSTATS
PTD    =        = PSTATS
PTURN  =        = INTSTA SUMARY
PVACC  / ABIAS  / ACCEL  BANGS  BIAS   CHGMLN IBAP   LOGIN  LOKIBI LOKIOB
                  OBAP   PREST1 SIGRES
PVPOS  / ABIAS  / ACCEL  BANGS  BIAS   CARFOL CHGMLN CHKMLN CRIDIS IBAP
                  INTERP LOGIN  LOKIBI LOKIOB NEWVEL OBAP   PREST1 SIGRES
PVRF   =        = DELAY
PVRR   =        = DELAY
PVSF   / LANECH / ABORTR CHGMLN DELAY  GAPACC LCHDES SVEHU
PVSR   / LANECH / DELAY  GAPACC SVEHU
PVVEL  / ABIAS  / ACCEL  ACDCP  BANGS  BIAS   CARFOL CHGMLN CHKCON IBAP
                  LOGIN  LOKIBI LOKIOB NEWVEL OBAP   PREST1 PREST2 SIGRES
QD     / SUMSTA / ADDSTA LOGOUT PSTATS
QTIME  / QUE    / ABORTR BLKDAT LOGIN  QUEUE  RDVPRD
QUEL   =        = DELAY
QUER   =        = DELAY
QUES   =        = DELAY
QUEUEL =        = SUMARY
RADIAN =        = CHKCON
RADICL =        = ACCEL  ACDCP  CARFOL CRIDIS GAPACC LCHDES PREDTV
REACTT =        = CRIDIS
RELDIS =        = GAPACC LCHDES PREDTV
RELEND / ABIAS  / CHGMLN CHKDSP IBAP   INTLOG LOGIN  SIGRES UNBIAS
RELNEW =        = CRIDIS
RELOLD =        = CRIDIS
RELPN  =        = ACCEL
RELPOS / ABIAS  / ACCEL  ACDCP  BANGS  CARFOL CHGMLN GAPACC NEWVEL SIGRES
RELSPD =        = LCHDES
RELVEL / ABIAS  / BANGS  CARFOL CHGMLN NEWVEL
RESPLA =        = GAPACC
RESPLE =        = GAPACC
SD     / SUMSTA / ADDSTA INTSTA LOGOUT PSTATS
SIMTIM / USER   / EXEC   PSTATS RUSERD SUMARY TIMSTA
SLOPE  =        = ACCEL  CARFOL CHKCON CHKDSP GAPACC LOGIN  PREDTV
SLOPEU =        = CARFOL
SLP    =        = BANGS  LOGIN
SLPDEC =        = GAPACC LCHDES
SLPLCH / LANECH / CARFOL CRIDIS GAPACC LCHDES LOGIN  NEWVEL UNBIAS
```

```
SLPNEW / ABIAS  / ACCEL  ACDCP  BANGS  BIAS   CARFOL CRIDIS HOLDSP IBAP
                  INTERP LOGIN  NEWVEL OBAP   SETPTV UNBIAS
SLPOLD / ABIAS  / ABORTR ACCEL  CARFOL GAPACC HOLDSP LOGIN  UNBIAS
SMEP   =        = EXTRAC FIND   LOGIC  REPACK STORE
SMSPD  =        = PSTATS
SN     =        = PREDTV
SO     =        = CHKCON PREDTV SETPTV
SPD    =        = CARFOL CHKDSP PREDTV
SPFACT =        = SSIBAP
STIME  / SUMSTA / ADDSTA LOGOUT PSTATS TIMSTA
STOPLD / LOOPS  / CHKLDT RLOOPD
STRT   =        = CHKLDT
STRTIM / USER   / ABORTR ACTSIG EXEC   INTSTA LOGOUT RUSERD SSIBAP SUMARY
                  TIMSTA
STRTLD / LOOPS  / CHKLDT RLOOPD
SUMDEL =        = INTSTA
SUMVOL =        = INTSTA
T      =        = ACCEL  ACDCP  CARFOL CHKDSP CRIDIS GAPACC LOGIN  NEWVEL
                  PREDTV
TAR    / PHASES / ACTSIG ACTSTA RPHASD
TSIG   =        = ACTSIG
TCAMSP / SIGCAM / INITAL PRESIG RCAMSD
TCH    =        = CHKCON CHKSDR
TCI    / PHASES / ACTSIG ACTSTA RPHASD
TCM    =        = CHKCON CHKSDR
TCRASH =        = CHKCON
TCU    =        = NEWVEL
TD     / SUMSTA / ADDSTA INITAL LOGOUT PSTATS
TEST   =        = RPHASD
TESTLP =        = IBAP
TFZ    =        = CHKCON CHKSDR
TGAPO  / PHASES / ACTSIG ACTSTA RPHASD
THES   =        = LSTOP
TII    / PHASES / ACTSIG ACTSTA RPHASD
TIM    =        = CHKCON CHKSDR
TIME   / USER   / ABORTR ACTSIG ADLVAI BANGS  CHKCON CHKDFP CHKLDT CHKSDR
                  CLRCON EXEC   GAPACC IBAP   INTERP INTSTA LOGIBI LOGIN
                  LOGOUT OBAP   PATHF  PRESIG QUEUE  RDVPRD RUSERD SETCON
                  SETLDF SSIBAP SUMARY UNSETC
TIMEND =        = CHKSDR
TIMNOW =        = INTSTA
TLAG   / USER   / CHKCON RDVPRD RUSERD SUMARY
TLDIST =        = LCHGEO
TLEAD  / USER   / CHKCON CHKSDR RDVPRD RUSERD SUMARY
TMAG1  =        = ACTSIG
TMAG2  =        = ACTSIG
TMAXO  / PHASES / ACTSIG ACTSTA RPHASD
TMIN   =        = TIMSTA
TMINT  =        = INTSTA
TMRAT  =        = TIMSTA
TMRDT  =        = TIMSTA
TMRSI  =        = TIMSTA
TMRSU  =        = TIMSTA
TMSI   =        = TIMSTA
TMSIM  =        = INTSTA
TMSPD  =        = PSTATS
TMSS   =        = TIMSTA
TMSU   =        = TIMSTA
TMTIME / SUMSTA / BLKDAT EXTIME INTSTA TIMSTA
TMTO   =        = TIMSTA
TMX    / PHASES / ACTSIG ACTSTA RPHASD
TOTDEL =        = INTSTA
TOTVOL =        = INTSTA
TP     / SIGCAM / ABORTR ACTSIG PRESIG RCAMSD RPHASD
TPASSH =        = CHKCON
TPASSM =        = CHKCON CHKSDR
TPLVDV =        = SUMARY
TPRINT / USER   / ACTSIG ADLVAI CHKCON CHKDFP CHKLDT CHKSDR CLRCON GAPACC
                  IBAP   INTERP LOGIN  OBAP   PRESIG RDVPRD RUSERD SETCON
                  SETLDF UNSETC
TR     / SIGCAM / ABORTR ACTSIG PRESIG RCAMSD RPHASD
TRZ    =        = CHKCON
TSQ    =        = NEWVEL
TSTATS / USER   / EXEC   RUSERD
TSTOP  =        = LOGIN
TT     =        = PREDTV
TVATIN / INTER  / ADLVAI LOGIBI
TVI    / PHASES / ACTSIG ACTSTA RPHASD
T1     =        = CARFOL GAPACC
T3     =        = SIGRES
V      =        = CRIDIS LOGIN  PVAPRT
VCH    =        = CHKCON
VCHAR  / CLASS  / GAPACC INTLOG LCHGEO RDVPRD
VCM    =        = CHKCON CHKSDR
VELNEW / ABIAS  / ACDCP  BANGS  BIAS   CHGMLN CHKSDR IBAP   INTERP INTLOG
                  LOGIN  NEWVEL OBAP   PVAPRT SETPTV SSIBAP SSINTR SSOBAP
VELOLD / ABIAS  / ACCEL  ACDCP  CARFOL CHKDSP CRIDIS GAPACC IBAP   INTLOG
                  LCHDES LCHGEO LOGIN  LOGOUT NEWVEL SIGRES UNBIAS
VMAX   / CLASS  / ACCEL  GAPACC PREDTV RDVPRD
VMAXA  / SUMSTA / ADDSTA LOGOUT PSTATS
VMAXD  / SUMSTA / ADDSTA LOGOUT PSTATS
VMT    / SUMSTA / ADDSTA LOGOUT PSTATS
VN     =        = PREDTV
VO     =        = CHKCON CHKSDR PREDTV SETPTV
VOLUME =        = INTSTA PSTATS
VSQT4  =        = CRIDIS GAPACC LCHDES
VT     =        = ACCEL
VTT    =        = PREDTV
VT1    =        = CARFOL
VT2    =        = CRIDIS GAPACC LCHDES
VVSF   / LANECH / CHGMLN GAPACC LCHDES
VVSR   / LANECH / GAPACC
VX     =        = PREDTV
X      =        = CRIDIS GAPACC
XCRIT  =        = CHGMLN CHKDSP CRIDIS INTLOG PREDTV SIGRES
XDISTL =        = LOGOUT
XDMPH  =        = LOGOUT
XFPS   / SUMSTA / LOGIN  PSTATS RUSERD SSIBAP SSINTR SSOBAP SUMARY
XMIT   =        = ABORTR EXEC
XMPH   =        = PSTATS RUSERD
XNEW   =        = LCHGEO
XOLD   =        = LCHGEO
XPER   =        = PREDTV
XQD    =        = LOGOUT
XQDIST / SUMSTA / RUSERD SSIBAP SUMARY
XSD    =        = LOGOUT
XSTIME =        = LOGOUT
XSTOP  =        = LOGIN
XT     =        = PREDTV
XTD    =        = LOGOUT
XTIMEL =        = LOGIN
XTOT   =        = LCHGEO
XVMT   =        = LOGOUT
```

9. GENERALIZED CALLING SEQUENCE DIAGRAM

```
SIMPRO
  ↓
EXEC
  ↓
----------------------------------------------------------------
EXTIME

ISLCPF

INITAL
#RUSERD
#RGEOPD
#RCAMSD
#RPHASD
#RLOOPD
#RDVPRD

QUEUE

OBAP
#PREST1
#PREST2
#UNBIAS
 >NEWVEL
#LCHGEO
 >ENDLCH
#LCHDES
 >CHKLSI
 >SVEHU
 >DELAY
   CKLALT
   CHKLSI
   SVEHU
 >GAPACC
 >CHGMLN
   FLGNOR
#ACDCP
 >CARFOL
   ACCEL
 >ACCEL
 >CRIDIS
   NEWVEL
   HOLDSP
     NEWVEL
   ACCEL
 >NEWVEL
 >HOLDSP
   NEWVEL
#PVAPRT
#SSOBAP
#BANGS
#BIAS
#LOGOUT
 >FLGNOR

INTERP
#PREST1
#LOKIOB
#PREST2
#UNBIAS
 >NEWVEL
#ACDCP
 >CARFOL
   ACCEL
 >ACCEL
 >CRIDIS
   NEWVEL
   HOLDSP
     NEWVEL
   ACCEL
 >NEWVEL
 >HOLDSP
   NEWVEL
#PVAPRT
#SSINTR
#CLRCON
#BANGS
#LOGIOB
 >FLGNOR
 >CHKMLN
#BIAS

IBAP
#PREST1
#LOKIBI
#PREST2
#UNBIAS
 >NEWVEL
#CHKDSP
#SIGRES
 >SETCON
 >UNSETC
#LCHGEO
 >ENDLCH
#PATHF
 >CHKMLN
#LCHDES
 >CHKLSI
 >SVEHU
 >DELAY
   CKLALT
   CHKLSI
   SVEHU
 >GAPACC
 >CHGMLN
   FLGNOR
   UNSETC
   PATHF
     CHKMLN
   INFLZN
     SETCON
     SIGRES
       SETCON
       UNSETC
 >PATHF
   CHKMLN
#ACDCP
 >CARFOL
   ACCEL
 >ACCEL
 >CRIDIS
   NEWVEL
   HOLDSP
     NEWVEL
   ACCEL
 >NEWVEL
 >ADLVAI
   ENDLCH
   PATHF
     CHKMLN
 >HOLDSP
   NEWVEL
#PVAPRT
#CHKLDT
#SSIBAP
#INTLOG
 >LSTOP
   CHKSDR
     SETPTV
     PREDTV
     CHKCON
       SETPTV
       PREDTV
       SETCON
 >CHKSDR
   SETPTV
   PREDTV
   CHKCON
     SETPTV
     PREDTV
     SETCON
 >INFLZN
   SETCON
   SIGRES
     SETCON
     UNSETC
 >PATHF
   CHKMLN
#LOGIBI
 >ENDLCH
 >PATHF
   CHKMLN
 >SETCON
 >FLGNOR
#BANGS
#BIAS
#LOGIN
 >CHKMLN
 >NEWVEL
 >CARFOL
   ACCEL
 >BIAS
 >PVAPRT

PRESIG

ACTSIG
#CHKDFP
#SETLDF

INTSTA
#EXTIME

SUMARY
#EXTIME
#PSTATS
#ADDSTA
#ACTSTA
#TIMSTA
 >EXTIME
```

APPENDIX E

COLEASE PRINTED OUTPUT

FOR GEOPRO AND SIMPRO

This page intentionally left blank to facilitate printing on 2 sides.

COLEASE 3.0 = THE UNIVERSITY OF TEXAS CENTER FOR HIGHWAY RESEARCH

```
   CO ORDINATED
   L OGIC
   E NTITY
   A TTRIBUTE
   S IMULATION
   E NVIRONMENT


 IDENTIFY,GEOPRO,60,3,GEOMETRY PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACK
   FILES,INPUT=513,OUTPUT=513,TAPE8=513,TAPE5=INPUT
 ENTITY
   NAME,APPRO,12,***** ENTITY FOR APPROACHES *****
   ORDINARY,IALEFT,12,IARGHT,12,NLANES,6,LLANES(6),50,IAPX,2250
   ORDINARY,IAPY,2250,ISLIM,118,NSDR,5,ISDRN(5),30,ISDRA(5),12
   ORDINARY,IAAZIM,360,NDEGST,45,NDEGUT,45
   NAME,ARC,20,***** ENTITY FOR ARCS *****
   ORDINARY,IARCX,2250,IARCY,2250,IARCAZ,360,IARCSW,720,IARCR,127
   ORDINARY,IDUMAR,0
   NAME,CONFLT,1000,***** ENTITY FOR INTERSECTION CONFLICTS *****
   ORDINARY,ICONP(2),125,ICONA(2),12,ICOND(2),250,ICONAN,360
   ORDINARY,ICONI(2),60,IDUMCO,0
   NAME,LANE,50,***** ENTITY FOR APPROACH LANES *****
   ORDINARY,LWID,15,NLL,50,NLR,50,ISNA,12,NPINT,7,LINTP(7),125
   ORDINARY,LTURN,15,LGEOM(4),1000,LTYPE,2,IDX,90,IBLN,25
   NAME,LINE,100,***** ENTITY FOR LINES *****
   ORDINARY,ILX1,2250,ILY1,2250,ILX2,2250,ILY2,2250
   NAME,PATH,125,***** ENTITY FOR INTERSECTION PATHS *****
   ORDINARY,IGEOCP(60),1000,IXL(2),2250,IYL(2),2250,JXL(2),2250
   ORDINARY,JYL(2),2250,IXA(2),4050,IYA(2),4050,LL1,250,LA1,250,LA2,250
   ORDINARY,LL2,250,IIA,12,IIL,6,IOA,12,IOL,6,IOPT,1,ILCH,1,IBA(2),360
   ORDINARY,IDA(2),720,IRA(2),900,IPTURN,8,LENP,250,LIBL,50,LOBL,50
   ORDINARY,LIMP,118,NGEOCP,60
   NAME,SDR,30,***** ENTITY FOR AVAILABLE APPROACH SIGHT DISTANCE *****
   ORDINARY,ICANSE(40),1000
```

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	IALEFT	1	0	4	4
2	IARGHT	1	4	4	8
3	NLANES	1	8	3	11
4	LLANES(1)	1	11	6	17
5	LLANES(2)	1	17	6	23
6	LLANES(3)	1	23	6	29
7	LLANES(4)	1	29	6	35
8	LLANES(5)	1	35	6	41
9	LLANES(6)	1	41	6	47
10	IAPX	1	47	12	59
11	IAPY	2	0	12	12
12	ISLIM	2	12	7	19
13	NSDR	2	19	3	22
14	ISDRN (1)	2	22	5	27
15	ISDRN (2)	2	27	5	32
16	ISDRN (3)	2	32	5	37
17	ISDRN (4)	2	37	5	42
18	ISDRN (5)	2	42	5	47
19	ISDRA (1)	2	47	4	51
20	ISDRA (2)	2	51	4	55
21	ISDRA (3)	2	55	4	59
22	ISDRA (4)	3	0	4	4
23	ISDRA (5)	3	4	4	8
24	IAAZIM	3	8	9	17
25	NDEGST	3	17	6	23
26	NDEGUT	3	23	6	29

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	IARCX	1	0	12	12
2	IARCY	1	12	12	24
3	IARCAZ	1	24	9	33
4	IARCSW	1	33	10	43
5	IARCR	1	43	7	50
6	IDUMAR	1	50	0	50

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ICONP (1)	1	0	7	7
2	ICONP (2)	1	7	7	14
3	ICONA (1)	1	14	4	18
4	ICONA (2)	1	18	4	22
5	ICOND (1)	1	22	8	30
6	ICOND (2)	1	30	8	38
7	ICONAN	1	38	9	47
8	ICONI (1)	1	47	6	53
9	ICONI (2)	1	53	6	59
10	IDUMCO	1	59	0	59

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	LWID	1	0	4	4
2	NLL	1	4	6	10
3	NLR	1	10	6	16
4	ISNA	1	16	4	20
5	NPINT	1	20	3	23
6	LINTP (1)	1	23	7	30
7	LINTP (2)	1	30	7	37
8	LINTP (3)	1	37	7	44
9	LINTP (4)	1	44	7	51
10	LINTP (5)	1	51	7	58
11	LINTP (6)	2	0	7	7
12	LINTP (7)	2	7	7	14
13	LTURN	2	14	4	18
14	LGEOM (1)	2	18	10	28
15	LGEOM (2)	2	28	10	38
16	LGEOM (3)	2	38	10	48
17	LGEOM (4)	2	48	10	58
18	LTYPE	2	58	2	60
19	IDX	3	0	7	7
20	IBLN	3	7	5	12

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ILX1	1	0	12	12
2	ILY1	1	12	12	24
3	ILX2	1	24	12	36
4	ILY2	1	36	12	48

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	IGEOCP(1)	1	0	10	10
2	IGEOCP(2)	1	10	10	20
3	IGEOCP(3)	1	20	10	30
4	IGEOCP(4)	1	30	10	40
5	IGEOCP(5)	1	40	10	50
6	IGEOCP(6)	1	50	10	60
7	IGEOCP(7)	2	0	10	10
8	IGEOCP(8)	2	10	10	20
9	IGEOCP(9)	2	20	10	30
10	IGEOCP(10)	2	30	10	40
11	IGEOCP(11)	2	40	10	50
12	IGEOCP(12)	2	50	10	60
13	IGEOCP(13)	3	0	10	10
14	IGEOCP(14)	3	10	10	20
15	IGEOCP(15)	3	20	10	30
16	IGEOCP(16)	3	30	10	40
17	IGEOCP(17)	3	40	10	50
18	IGEOCP(18)	3	50	10	60
19	IGEOCP(19)	4	0	10	10
20	IGEOCP(20)	4	10	10	20
21	IGEOCP(21)	4	20	10	30
22	IGEOCP(22)	4	30	10	40
23	IGEOCP(23)	4	40	10	50
24	IGEOCP(24)	4	50	10	60
25	IGEOCP(25)	5	0	10	10
26	IGEOCP(26)	5	10	10	20
27	IGEOCP(27)	5	20	10	30
28	IGEOCP(28)	5	30	10	40
29	IGEOCP(29)	5	40	10	50
30	IGEOCP(30)	5	50	10	60
31	IGEOCP(31)	6	0	10	10
32	IGEOCP(32)	6	10	10	20
33	IGEOCP(33)	6	20	10	30
34	IGEOCP(34)	6	30	10	40
35	IGEOCP(35)	6	40	10	50
36	IGEOCP(36)	6	50	10	60
37	IGEOCP(37)	7	0	10	10
38	IGEOCP(38)	7	10	10	20
39	IGEOCP(39)	7	20	10	30
40	IGEOCP(40)	7	30	10	40
41	IGEOCP(41)	7	40	10	50
42	IGEOCP(42)	7	50	10	60
43	IGEOCP(43)	8	0	10	10
44	IGEOCP(44)	8	10	10	20
45	IGEOCP(45)	8	20	10	30
46	IGEOCP(46)	8	30	10	40
47	IGEOCP(47)	8	40	10	50
48	IGEOCP(48)	8	50	10	60
49	IGEOCP(49)	9	0	10	10
50	IGEOCP(50)	9	10	10	20
51	IGEOCP(51)	9	20	10	30
52	IGEOCP(52)	9	30	10	40
53	IGEOCP(53)	9	40	10	50
54	IGEOCP(54)	9	50	10	60
55	IGEOCP(55)	10	0	10	10
56	IGEOCP(56)	10	10	10	20
57	IGEOCP(57)	10	20	10	30
58	IGEOCP(58)	10	30	10	40
59	IGEOCP(59)	10	40	10	50
60	IGEOCP(60)	10	50	10	60
61	IXL (1)	11	0	12	12
62	IXL (2)	11	12	12	24
63	IYL (1)	11	24	12	36
64	IYL (2)	11	36	12	48
65	JXL (1)	11	48	12	60
66	JXL (2)	12	0	12	12
67	JYL (1)	12	12	12	24
68	JYL (2)	12	24	12	36
69	IXA (1)	12	36	12	48
70	IXA (2)	12	48	12	60
71	IYA (1)	13	0	12	12
72	IYA (2)	13	12	12	24
73	LL1	13	24	8	32
74	LA1	13	32	8	40
75	LA2	13	40	8	48
76	LL2	13	48	8	56
77	IIA	13	56	4	60
78	IIL	14	0	3	3
79	IOA	14	3	4	7
80	IOL	14	7	3	10
81	IOPT	14	10	1	11
82	ILCH	14	11	1	12
83	IBA (1)	14	12	9	21
84	IBA (2)	14	21	9	30
85	IDA (1)	14	30	10	40
86	IDA (2)	14	40	10	50
87	IRA (1)	14	50	10	60
88	IRA (2)	15	0	10	10
89	IPTURN	15	10	4	14
90	LENP	15	14	8	22
91	LIBL	15	22	6	28
92	LOBL	15	28	6	34
93	LIMP	15	34	7	41
94	NGEOCP	15	41	6	47

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ICANSE(1)	1	0	10	10
2	ICANSE(2)	1	10	10	20
3	ICANSE(3)	1	20	10	30
4	ICANSE(4)	1	30	10	40
5	ICANSE(5)	1	40	10	50
6	ICANSE(6)	1	50	10	60
7	ICANSE(7)	2	0	10	10
8	ICANSE(8)	2	10	10	20
9	ICANSE(9)	2	20	10	30
10	ICANSE(10)	2	30	10	40
11	ICANSE(11)	2	40	10	50
12	ICANSE(12)	2	50	10	60
13	ICANSE(13)	3	0	10	10
14	ICANSE(14)	3	10	10	20
15	ICANSE(15)	3	20	10	30
16	ICANSE(16)	3	30	10	40
17	ICANSE(17)	3	40	10	50
18	ICANSE(18)	3	50	10	60
19	ICANSE(19)	4	0	10	10
20	ICANSE(20)	4	10	10	20
21	ICANSE(21)	4	20	10	30
22	ICANSE(22)	4	30	10	40
23	ICANSE(23)	4	40	10	50
24	ICANSE(24)	4	50	10	60
25	ICANSE(25)	5	0	10	10
26	ICANSE(26)	5	10	10	20
27	ICANSE(27)	5	20	10	30
28	ICANSE(28)	5	30	10	40
29	ICANSE(29)	5	40	10	50
30	ICANSE(30)	5	50	10	60
31	ICANSE(31)	6	0	10	10
32	ICANSE(32)	6	10	10	20
33	ICANSE(33)	6	20	10	30
34	ICANSE(34)	6	30	10	40
35	ICANSE(35)	6	40	10	50
36	ICANSE(36)	6	50	10	60
37	ICANSE(37)	7	0	10	10
38	ICANSE(38)	7	10	10	20
39	ICANSE(39)	7	20	10	30
40	ICANSE(40)	7	30	10	40

```
EXECUTIVE
  ROUTINE,READAP,APPRO           ,LANE     ,NOATTB
  ROUTINE,READAI      ,ARC                 ,NOATTB
  ROUTINE,READLI                      ,LINE,NOATTB
  ROUTINE,WRITAL      ,ARC            ,LINE
  ROUTINE,FNDXYP,APPRO
  ROUTINE,FNDSDR,APPRO           ,LANE                 ,SDR
  ROUTINE,WRITAP,APPRO
  ROUTINE,DRWAPR,APPRO,ARC       ,LANE,LINE
  ROUTINE,DRWBOX,APPRO
  ROUTINE,DRWINT,APPRO,ARC       ,LANE,LINE
  ROUTINE,DRWUTA,APPRO
  ROUTINE,FNDPTH                           ,NOATTB,PATH
  ROUTINE,ADDPTH                                  ,PATH
  ROUTINE,DRWPTH                                  ,PATH
  ROUTINE,CHKPTH,APPRO           ,LANE
  ROUTINE,WRITLA                 ,LANE                 ,SDR
  ROUTINE,FNDCON                                  ,PATH
  ROUTINE,CLTOLC                                  ,PATH
  ROUTINE,ADDCON          ,CONFLT
  ROUTINE,CLTOAC                                  ,PATH
  ROUTINE,ADDLA                                   ,PATH
  ROUTINE,CATOLC                                  ,PATH
  ROUTINE,ADDAL                                   ,PATH
  ROUTINE,CATOAC                                  ,PATH
  ROUTINE,ADDAA                                   ,PATH
  ROUTINE,SRTCON          ,CONFLT                 ,PATH
  ROUTINE,WRITPA                                  ,PATH
  ROUTINE,NDXCON          ,CONFLT                 ,PATH
  ROUTINE,WRITCO          ,CONFLT
  ROUTINE,ABORTR,APPRO,ARC,CONFLT,LANE,LINE,NOATTB,PATH,SDR
  ROUTINE,ECHO  ,APPRO,ARC,CONFLT,LANE,LINE,NOATTB,PATH,SDR
  EXECUTE,EXEC
TASKS

  TASK,READAP
  COLEASE,REPACK,LANE,IL
  COLEASE,REPACK,APPRO,IA
  COLEASE,FIND,JAAZIM,APPRO,IA,IAAZIM
  COLEASE,FIND,KAAZIM,APPRO,JA,IAAZIM
  COLEASE,STORE,IALEFT,APPRO,IA,IALEFT
  COLEASE,STORE,IARGHT,APPRO,IA,IARGHT

  TASK,READAI
  COLEASE,REPACK,ARC,J

  TASK,READLI
  COLEASE,REPACK,LINE,J

  TASK,WRITAL
  COLEASE,EXTRAC,ARC,IARC
  COLEASE,EXTRAC,LINE,ILINE

  TASK,FNDXYP
  COLEASE,EXTRAC,APPRO,IA
  COLEASE,FIND,LWID,LANE,IL,LWID
  COLEASE,FIND,LGEOM3,LANE,IL,LGEOM(3)
  COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
  COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
  COLEASE,STORE,IDX,LANE,IL,IDX
  COLEASE,EXTRAC,APPRO,IA
  COLEASE,FIND,LWID,LANE,IL,LWID
  COLEASE,FIND,LGEOM1,LANE,IL,LGEOM(1)
  COLEASE,FIND,LGEOM2,LANE,IL,LGEOM(2)
  COLEASE,FIND,LGEOM4,LANE,IL,LGEOM(4)
  COLEASE,STORE,IDX,LANE,IL,IDX

  TASK,FNDSDR
  COLEASE,EXTRAC,APPRO,IA
  COLEASE,EXTRAC,LANE,IL
  COLEASE,EXTRAC,APPRO,JA
```

```
COLEASE,EXTRAC,LANE,JL
COLEASE,REPACK,SDR,NSDRS
COLEASE,STORE,NSDRAP,APPRO,IA,NSDR
COLEASE,STORE,NSDRS,APPRO,IA,ISDRN(NSDRAP)
COLEASE,STORE,JA,APPRO,IA,ISDRA(NSDRAP)

TASK,WRITAP
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,APPRO,IA

TASK,DRWAPR
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,LANE,IL
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,LANE,IL
COLEASE,EXTRAC,ARC,IARC
COLEASE,EXTRAC,LINE,ILINE

TASK,DRWBOX,IX1,IX2,IL1,IL2

TASK,DRWINT
COLEASE,EXTRAC,APPRO,KA
COLEASE,EXTRAC,LANE,KL
COLEASE,EXTRAC,APPRO,KA
COLEASE,EXTRAC,LANE,KL
COLEASE,EXTRAC,ARC,IARC
COLEASE,EXTRAC,LINE,ILINE

TASK,DRWUTA,ILANE

TASK,FNDPTH
COLEASE,FIND,JAZIM,APPRO,IA,IAAZIM
COLEASE,FIND,NLANEI,APPRO,IA,NLANES
COLEASE,FIND,IL,APPRO,IA,LLANES(ILN)
COLEASE,FIND,KAZIM,APPRO,JA,IAAZIM
COLEASE,FIND,NLANEJ,APPRO,JA,NLANES
COLEASE,FIND,JL,APPRO,JA,LLANES(JLN)
COLEASE,FIND,NDEGST,APPRO,IA,NDEGST
COLEASE,FIND,NDEGUT,APPRO,IA,NDEGUT
COLEASE,FIND,ITURN,LANE,IL,LTURN
COLEASE,FIND,JTURN,LANE,JL,LTURN
COLEASE,FIND,LN,APPRO,IA,LLANES(LNI)
COLEASE,FIND,MTURN,LANE,LN,LTURN
COLEASE,FIND,LN,APPRO,JA,LLANES(LNJ)
COLEASE,FIND,MTURN,LANE,LN,LTURN
COLEASE,FIND,LN,APPRO,IA,LLANES(LNI)
COLEASE,FIND,MTURN,LANE,LN,LTURN
COLEASE,FIND,LN,APPRO,JA,LLANES(LNJ)
COLEASE,FIND,MTURN,LANE,LN,LTURN

TASK,ADDPTH
COLEASE,FIND,JSLIM,APPRO,IA,ISLIM
COLEASE,FIND,KSLIM,APPRO,JA,ISLIM
COLEASE,REPACK,PATH,NPATHS
COLEASE,FIND,NPINT,LANE,IL,NPINT
COLEASE,STORE,NPINT,LANE,IL,NPINT
COLEASE,STORE,NPATHS,LANE,IL,LINTP(NPINT)

TASK,DRWPTH

TASK,CHKPTH
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,LANE,IL
COLEASE,FIND,IPTURN,PATH,JPINT,IPTURN

TASK,WRITLA
COLEASE,EXTRAC,LANE,ILANE
COLEASE,EXTRAC,SDR,ISDRS

TASK,FNDCON
COLEASE,EXTRAC,PATH,MPTH
COLEASE,EXTRAC,PATH,NPTH

TASK,CLTOLC,IFS,IBAND,JFS,NC

TASK,ADDCON,INP,INA,INL,AI,JNP,JNA,JNL,AJ,NC
COLEASE,EXTRAC,CONFLT,ICON
COLEASE,REPACK,CONFLT,ICON
COLEASE,FIND,MGEOCP,PATH,INP,NGEOCP
COLEASE,STORE,NCONFS,PATH,INP,IGEOCP(MGEOCP)
COLEASE,STORE,MGEOCP,PATH,INP,NGEOCP
COLEASE,FIND,MGEOCP,PATH,JNP,NGEOCP
COLEASE,STORE,NCONFS,PATH,JNP,IGEOCP(MGEOCP)
COLEASE,STORE,MGEOCP,PATH,JNP,NGEOCP
COLEASE,REPACK,CONFLT,NCONFS

TASK,CLTOAC,IFS,IBAND,JFS,NC

TASK,ADDLA,IFS,IBAND,JFS,NC,NUM

TASK,CATOLC,IFS,IBAND,JFS,NC

TASK,ADDAL,IFS,IBAND,JFS,NC,NUM

TASK,CATOAC,IFS,IBAND,JFS,NC

TASK,ADDAA,IFS,IBAND,JFS,NC,NUM

TASK,SRTCON
COLEASE,EXTRAC,PATH,IPTH
COLEASE,EXTRAC,CONFLT,JCON
COLEASE,REPACK,PATH,IPTH

TASK,WRITPA
COLEASE,EXTRAC,PATH,I

TASK,NDXCON
COLEASE,EXTRAC,CONFLT,ICON
COLEASE,EXTRAC,PATH,IPTH
COLEASE,REPACK,CONFLT,ICON

TASK,WRITCO
COLEASE,EXTRAC,CONFLT,ICON

TASK,ABORTR,MSG,NCHS

TASK,ECHO
COLEASE,EXTRAC,ARC,J
COLEASE,EXTRAC,LINE,J
COLEASE,EXTRAC,APPRO,J
COLEASE,EXTRAC,APPRO,J
COLEASE,EXTRAC,LANE,I
COLEASE,EXTRAC,SDR,I
COLEASE,EXTRAC,PATH,I
COLEASE,EXTRAC,CONFLT,I
TERMINATE
```

COLEASE 3.0 - THE UNIVERSITY OF TEXAS CENTER FOR HIGHWAY RESEARCH

CO ORDINATED
L OGIC
E NTITY
A TTRIBUTE
S IMULATION
E NVIRONMENT

```
IDENTIFY,SIMPRO,60,3,SIMULATION PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PA
  FILES,TAPE5=513,TAPE7=65,TAPE8=513,TAPE9=513,OUTPUT=513
ENTITY
  NAME,APPRO,12,***** ENTITY FOR APPROACHES *****
  ORDINARY,NLANES,6,LLANES(6),50,NVIL(6),63,ISLIM,118,IALEFT,12
  ORDINARY,NSDR,5,ISDRN(5),30,ISDRA(5),12
  NAME,CONFLT,1000,***** ENTITY FOR INTERSECTION CONFLICTS *****
  ORDINARY,ICONP(2),125,ICONA(2),12,ICOND(2),250,ICONAN,360
  ORDINARY,ICONI(2),60,ICONV(2),200,IDUMCO,0
  NAME,LANE,50,***** ENTITY FOR APPROACH LANES *****
  ORDINARY,LWID,15,NLL,50,NLR,50,ISNA,12,NPINT,7,LINTP(7),125
  ORDINARY,IFVL,200,ILVL,200,LCONTR,7,LTURN,15,LGEOM(4),1000
  ORDINARY,NLDL,5,LLDL(5),20,IBLN,25,IDUMLA,0
  NAME,PATH,125,***** ENTITY FOR INTERSECTION PATHS *****
  ORDINARY,LENP,250,IOPT,1,LIBL,50,LOBL,50,IFVP,200,ILVP,200,LIMP,118
  ORDINARY,IPT,8,NGEOCP,60,NCPSET,60,ICPSET(60),1,LOBAP,12,ILCH,1
  ORDINARY,IGEOCP(60),1000
  NAME,SDR,30,***** ENTITY FOR SIGHT DISTANCE RESTRICTION *****
  ORDINARY,ICANSE(40),1000
  NAME,VEHD,200,***** ENTITY FOR DYNAMIC VEHICLE ATTRIBUTES *****
  ORDINARY,ISLP,8000,IACC,16000,IVEL,4034,IPOS,25000,ISET,6
  ORDINARY,LCHGE,3,ISPDP,1,LEGAL,30,IPRTM,15,ITIMV,2000,IQDS,2000
  ORDINARY,ISPDS,258134,ISDS,2000,IDVS,2000,ISTCON,61,IVMAXA,320
  ORDINARY,IVMAXD,320,LATPOS,240,IDTS,56240,LALT,5,NORC,201,LOGFLG,15
  LOGICI,MSTPF,MLAG,MTCAPS,MFINL,MSFLG,MPOBS,MOASF,MSAOR,MPRO,MBLOCK
  LOGICI,MININT
  LOGICD,IFVA,IACDS,ICDFS,ISDEC,ISTMO,IACLDS,IRSTOP
  FUNCTION,MSTPF   ,MPOBS   ,MLAG    ,MLAG    ,IFVA    ,MFINL =1
  FUNCTION,MFINL =1,MTCARS  ,MOASF =1,MOASF =1,MSFLG   ,IFVA
  FUNCTION,MTCARS  ,MSFLG   ,MBLOCK  ,MBLOCK  ,MSFLG   ,MPRO  =1
  FUNCTION,MPRO  =1,IACDS   ,MSFLG   ,MSFLG   ,ICDFS   ,ISDEC
  FUNCTION,MPOBS   ,ISTMO   ,MFINL =2,MFINL =2,MSAOR =1,MOASF =2
  FUNCTION,MOASF =2,MSAOR =2,IACLDS  ,MSAOR =1,MPRO  =2,IACDS
  FUNCTION,MSAOR =2,IRSTOP  ,IACDS   ,MPRO  =2,IACDS   ,IRSTOP
  NAME,VEHF,200,***** ENTITY FOR FIXED VEHICLE ATTRIBUTES *****
  ORDINARY,IDRICL,5,IVEHCL,15,ISPD,161,NOF,200,NOR,200,LNEXT,125
  ORDINARY,LPRES,125,ITURN,3,IBAPS,6,IPRTLO,1,IEXTTM,25,NOBAPD,12
  NAME,VEHIL,200,***** ENTITY FOR VEHICLE INTERSECTION LOGIC *****
  LOGICI,MDEDIC,MINFLZ,MLUNC,MIUNC,MLYELD,MLSTOP,MATSTL,MSSRED,MLRTOR
  LOGICI,MSSGRN,MCHKCF,MDUMIL
  LOGICD,IDEDIC,INFLZ,ILUNC,ILYELD,ILSTOP,ICONTN,ICHKCF,IERROR
  FUNCTION,MDEDIC  ,MINFLZ  ,IDEDIC  ,MINFLZ  ,MLUNC   ,INFLZ
  FUNCTION,MLUNC   ,MIUNC   ,MLYELD  ,MLYELD  ,ILYELD  ,MLSTOP
  FUNCTION,MLSTOP  ,MATSTL  ,MSSRED  ,MATSTL  ,ILSTOP  ,ICONTN
  FUNCTION,MSSRED  ,MLRTOR  ,MSSGRN  ,MLRTOR  ,ICHKCF  ,ICONTN
  FUNCTION,MSSGRN  ,MCHKCF  ,IERROR  ,MCHKCF  ,ICHKCF  ,ICONTN
  FUNCTION,MIUNC   ,ILUNC   ,MCHKCF
```

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	NLANES	1	0	3	3
2	LLANES(1)	1	3	6	9
3	LLANES(2)	1	9	6	15
4	LLANES(3)	1	15	6	21
5	LLANES(4)	1	21	6	27
6	LLANES(5)	1	27	6	33
7	LLANES(6)	1	33	6	39
8	NVIL (1)	1	39	6	45
9	NVIL (2)	1	45	6	51
10	NVIL (3)	1	51	6	57
11	NVIL (4)	2	0	6	6
12	NVIL (5)	2	6	6	12
13	NVIL (6)	2	12	6	18
14	ISLIM	2	18	7	25
15	IALEFT	2	25	4	29
16	NSDR	2	29	3	32
17	ISDRN (1)	2	32	5	37
18	ISDRN (2)	2	37	5	42
19	ISDRN (3)	2	42	5	47
20	ISDRN (4)	2	47	5	52
21	ISDRN (5)	2	52	5	57
22	ISDRA (1)	3	0	4	4
23	ISDRA (2)	3	4	4	8
24	ISDRA (3)	3	8	4	12
25	ISDRA (4)	3	12	4	16
26	ISDRA (5)	3	16	4	20

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ICONP (1)	1	0	7	7
2	ICONP (2)	1	7	7	14
3	ICONA (1)	1	14	4	18
4	ICONA (2)	1	18	4	22
5	ICOND (1)	1	22	8	30
6	ICOND (2)	1	30	8	38
7	ICONAN	1	38	9	47
8	ICONI (1)	1	47	6	53
9	ICONI (2)	1	53	6	59
10	ICONV (1)	2	0	8	8
11	ICONV (2)	2	8	8	16
12	IDUMCO	2	16	0	16

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	LWID	1	0	4	4
2	NLL	1	4	6	10
3	NLR	1	10	6	16
4	ISNA	1	16	4	20
5	NPINT	1	20	3	23
6	LINTP (1)	1	23	7	30
7	LINTP (2)	1	30	7	37
8	LINTP (3)	1	37	7	44
9	LINTP (4)	1	44	7	51
10	LINTP (5)	1	51	7	58
11	LINTP (6)	2	0	7	7
12	LINTP (7)	2	7	7	14
13	IFVL	2	14	8	22
14	ILVL	2	22	8	30
15	LCONTR	2	30	3	33
16	LTURN	2	33	4	37
17	LGEOM (1)	2	37	10	47
18	LGEOM (2)	2	47	10	57
19	LGEOM (3)	3	0	10	10
20	LGEOM (4)	3	10	10	20
21	NLDL	3	20	3	23
22	LLDL (1)	3	23	5	28
23	LLDL (2)	3	28	5	33
24	LLDL (3)	3	33	5	38
25	LLDL (4)	3	38	5	43
26	LLDL (5)	3	43	5	48
27	IBLN	3	48	5	53
28	IDUMLA	3	53	0	53

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	LENP	1	0	8	8
2	IOPT	1	8	1	9
3	LIBL	1	9	6	15
4	LOBL	1	15	6	21
5	IFVP	1	21	8	29
6	ILVP	1	29	8	37
7	LIMP	1	37	7	44
8	IPT	1	44	4	48
9	NGEOCP	1	48	6	54
10	NCPSET	1	54	6	60
11	ICPSET(1)	2	0	1	1
12	ICPSET(2)	2	1	1	2
13	ICPSET(3)	2	2	1	3
14	ICPSET(4)	2	3	1	4
15	ICPSET(5)	2	4	1	5
16	ICPSET(6)	2	5	1	6
17	ICPSET(7)	2	6	1	7
18	ICPSET(8)	2	7	1	8
19	ICPSET(9)	2	8	1	9
20	ICPSET(10)	2	9	1	10
21	ICPSET(11)	2	10	1	11
22	ICPSET(12)	2	11	1	12
23	ICPSET(13)	2	12	1	13
24	ICPSET(14)	2	13	1	14
25	ICPSET(15)	2	14	1	15
26	ICPSET(16)	2	15	1	16
27	ICPSET(17)	2	16	1	17
28	ICPSET(18)	2	17	1	18
29	ICPSET(19)	2	18	1	19
30	ICPSET(20)	2	19	1	20
31	ICPSET(21)	2	20	1	21
32	ICPSET(22)	2	21	1	22
33	ICPSET(23)	2	22	1	23
34	ICPSET(24)	2	23	1	24
35	ICPSET(25)	2	24	1	25
36	ICPSET(26)	2	25	1	26
37	ICPSET(27)	2	26	1	27
38	ICPSET(28)	2	27	1	28
39	ICPSET(29)	2	28	1	29
40	ICPSET(30)	2	29	1	30
41	ICPSET(31)	2	30	1	31
42	ICPSET(32)	2	31	1	32
43	ICPSET(33)	2	32	1	33
44	ICPSET(34)	2	33	1	34
45	ICPSET(35)	2	34	1	35
46	ICPSET(36)	2	35	1	36
47	ICPSET(37)	2	36	1	37
48	ICPSET(38)	2	37	1	38
49	ICPSET(39)	2	38	1	39
50	ICPSET(40)	2	39	1	40
51	ICPSET(41)	2	40	1	41
52	ICPSET(42)	2	41	1	42
53	ICPSET(43)	2	42	1	43
54	ICPSET(44)	2	43	1	44
55	ICPSET(45)	2	44	1	45
56	ICPSET(46)	2	45	1	46
57	ICPSET(47)	2	46	1	47
58	ICPSET(48)	2	47	1	48
59	ICPSET(49)	2	48	1	49
60	ICPSET(50)	2	49	1	50
61	ICPSET(51)	2	50	1	51
62	ICPSET(52)	2	51	1	52
63	ICPSET(53)	2	52	1	53
64	ICPSET(54)	2	53	1	54
65	ICPSET(55)	2	54	1	55
66	ICPSET(56)	2	55	1	56
67	ICPSET(57)	2	56	1	57
68	ICPSET(58)	2	57	1	58
69	ICPSET(59)	2	58	1	59
70	ICPSET(60)	2	59	1	60
71	LOBAP	3	0	4	4
72	ILCH	3	4	1	5
73	IGEOCP(1)	3	5	10	15
74	IGEOCP(2)	3	15	10	25
75	IGEOCP(3)	3	25	10	35
76	IGEOCP(4)	3	35	10	45
77	IGEOCP(5)	3	45	10	55
78	IGEOCP(6)	4	0	10	10
79	IGEOCP(7)	4	10	10	20
80	IGEOCP(8)	4	20	10	30
81	IGEOCP(9)	4	30	10	40
82	IGEOCP(10)	4	40	10	50
83	IGEOCP(11)	4	50	10	60
84	IGEOCP(12)	5	0	10	10
85	IGEOCP(13)	5	10	10	20
86	IGEOCP(14)	5	20	10	30
87	IGEOCP(15)	5	30	10	40
88	IGEOCP(16)	5	40	10	50
89	IGEOCP(17)	5	50	10	60
90	IGEOCP(18)	6	0	10	10
91	IGEOCP(19)	6	10	10	20
92	IGEOCP(20)	6	20	10	30
93	IGEOCP(21)	6	30	10	40
94	IGEOCP(22)	6	40	10	50
95	IGEOCP(23)	6	50	10	60
96	IGEOCP(24)	7	0	10	10
97	IGEOCP(25)	7	10	10	20
98	IGEOCP(26)	7	20	10	30
99	IGEOCP(27)	7	30	10	40
100	IGEOCP(28)	7	40	10	50
101	IGEOCP(29)	7	50	10	60
102	IGEOCP(30)	8	0	10	10
103	IGEOCP(31)	8	10	10	20
104	IGEOCP(32)	8	20	10	30
105	IGEOCP(33)	8	30	10	40
106	IGEOCP(34)	8	40	10	50
107	IGEOCP(35)	8	50	10	60
108	IGEOCP(36)	9	0	10	10
109	IGEOCP(37)	9	10	10	20
110	IGEOCP(38)	9	20	10	30
111	IGEOCP(39)	9	30	10	40
112	IGEOCP(40)	9	40	10	50
113	IGEOCP(41)	9	50	10	60
114	IGEOCP(42)	10	0	10	10
115	IGEOCP(43)	10	10	10	20
116	IGEOCP(44)	10	20	10	30
117	IGEOCP(45)	10	30	10	40
118	IGEOCP(46)	10	40	10	50
119	IGEOCP(47)	10	50	10	60
120	IGEOCP(48)	11	0	10	10
121	IGEOCP(49)	11	10	10	20
122	IGEOCP(50)	11	20	10	30
123	IGEOCP(51)	11	30	10	40
124	IGEOCP(52)	11	40	10	50
125	IGEOCP(53)	11	50	10	60
126	IGEOCP(54)	12	0	10	10
127	IGEOCP(55)	12	10	10	20
128	IGEOCP(56)	12	20	10	30
129	IGEOCP(57)	12	30	10	40
130	IGEOCP(58)	12	40	10	50
131	IGEOCP(59)	12	50	10	60
132	IGEOCP(60)	13	0	10	10

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ICANSE(1)	1	0	10	10
2	ICANSE(2)	1	10	10	20
3	ICANSE(3)	1	20	10	30
4	ICANSE(4)	1	30	10	40
5	ICANSE(5)	1	40	10	50
6	ICANSE(6)	1	50	10	60
7	ICANSE(7)	2	0	10	10
8	ICANSE(8)	2	10	10	20
9	ICANSE(9)	2	20	10	30
10	ICANSE(10)	2	30	10	40
11	ICANSE(11)	2	40	10	50
12	ICANSE(12)	2	50	10	60
13	ICANSE(13)	3	0	10	10
14	ICANSE(14)	3	10	10	20
15	ICANSE(15)	3	20	10	30
16	ICANSE(16)	3	30	10	40
17	ICANSE(17)	3	40	10	50
18	ICANSE(18)	3	50	10	60
19	ICANSE(19)	4	0	10	10
20	ICANSE(20)	4	10	10	20
21	ICANSE(21)	4	20	10	30
22	ICANSE(22)	4	30	10	40
23	ICANSE(23)	4	40	10	50
24	ICANSE(24)	4	50	10	60
25	ICANSE(25)	5	0	10	10
26	ICANSE(26)	5	10	10	20
27	ICANSE(27)	5	20	10	30
28	ICANSE(28)	5	30	10	40
29	ICANSE(29)	5	40	10	50
30	ICANSE(30)	5	50	10	60
31	ICANSE(31)	6	0	10	10
32	ICANSE(32)	6	10	10	20
33	ICANSE(33)	6	20	10	30
34	ICANSE(34)	6	30	10	40
35	ICANSE(35)	6	40	10	50
36	ICANSE(36)	6	50	10	60
37	ICANSE(37)	7	0	10	10
38	ICANSE(38)	7	10	10	20
39	ICANSE(39)	7	20	10	30
40	ICANSE(40)	7	30	10	40

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	ISLP	1	0	13	13
2	IACC	1	13	14	27
3	IVEL	1	27	12	39
4	IPOS	1	39	15	54
5	ISET	1	54	3	57
6	LCHGE	1	57	2	59
7	ISPDP	1	59	1	60
8	LEGAL	2	0	5	5
9	IPRTM	2	5	4	9
10	ITIMV	2	9	11	20
11	IQD8	2	20	11	31
12	ISPDS	2	31	18	49
13	ISDS	2	49	11	60
14	IDVS	3	0	11	11
15	ISTCON	3	11	6	17
16	IVMAXA	3	17	9	26
17	IVMAXD	3	26	9	35
18	LATPOS	3	35	8	43
19	IDTS	3	43	16	59
20	LALT	4	0	3	3
21	NURC	4	3	8	11
22	LOGFLG	4	11	4	15
23	MSTPF	5	0	2	2
24	MLAG	5	2	2	4
25	MTCARS	5	4	2	6
26	MFINL	5	6	2	8
27	MSFLG	5	8	2	10
28	MPOBS	5	10	2	12
29	MOASF	5	12	2	14
30	MSADR	5	14	2	16
31	MPRO	5	16	2	18
32	MBLOCK	5	18	2	20
33	MININT	5	20	2	22
34	IFVA	6	0	2	2
35	IACDS	6	2	2	4
36	ICDFS	6	4	2	6
37	ISDEC	6	6	2	8
38	ISTMD	6	8	2	10
39	IACLDS	6	10	2	12
40	IRSTOP	6	12	2	14

```
DEPENDENT ATTRIBUTE IFVA   IS TRUE FOR:

  + MLAG       - MSTPF

                  - OR -

  - MOASF      - MFINL      - MLAG       - MSTPF

DEPENDENT ATTRIBUTE IACDS  IS TRUE FOR:

  + MPRO       - MBLOCK     - MTCARS     + MFINL      - MLAG
  - MSTPF

                  - OR -

  - MSAOR      + MFINL      - MPOBS      + MSTPF

                  - OR -

  - MSAOR      + MOASF      - MFINL      - MPOBS      + MSTPF

                  - OR -

  + MPRO       + MSAOR      + MFINL      - MPOBS      + MSTPF

DEPENDENT ATTRIBUTE ICOFS  IS TRUE FOR:

  + MSFLG      + MOASF      - MFINL      - MLAG       - MSTPF

                  - OR -

  + MSFLG      + MTCARS     + MFINL      - MLAG       - MSTPF

                  - OR -

  + MSFLG      + MBLOCK     - MTCARS     + MFINL      - MLAG
  - MSTPF

                  - OR -

  + MSFLG      - MPRO       - MBLOCK     - MTCARS     + MFINL
  - MLAG       - MSTPF

DEPENDENT ATTRIBUTE ISDEC  IS TRUE FOR:

  - MSFLG      + MOASF      - MFINL      - MLAG       - MSTPF

                  - OR -

  - MSFLG      + MTCARS     + MFINL      - MLAG       - MSTPF

                  - OR -

  - MSFLG      + MBLOCK     - MTCARS     + MFINL      - MLAG
  - MSTPF

                  - OR -

  - MSFLG      - MPRO       - MBLOCK     - MTCARS     + MFINL
  - MLAG       - MSTPF

DEPENDENT ATTRIBUTE ISTMO  IS TRUE FOR:

  + MPOBS      + MSTPF

DEPENDENT ATTRIBUTE IACLDS IS TRUE FOR:

  - MOASF      - MFINL      - MPOBS      + MSTPF

  + MSAOR      + MOASF      - MFINL      - MPOBS      + MSTPF

                  - OR -

  - MPRO       + MSAOR      + MFINL      - MPOBS      + MSTPF
```

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	IDRICL	1	0	3	3
2	IVEHCL	1	3	4	7
3	ISPD	1	7	8	15
4	NOF	1	15	8	23
5	NOR	1	23	8	31
6	LNEXT	1	31	7	38
7	LPRES	1	38	7	45
8	ITURN	1	45	2	47
9	IBAPS	1	47	3	50
10	IPRTLD	1	50	1	51
11	IEXTIM	1	51	5	56
12	NOBAPD	1	56	4	60

NUMBER	NAME	WORD IN ENTITY	STARTING BIT IN WORD	NUMBER OF BITS	TOTAL BITS FOR WORD
1	MDEDIC	1	0	2	2
2	MINFLZ	1	2	2	4
3	MLUNC	1	4	2	6
4	MIUNC	1	6	2	8
5	MLYELD	1	8	2	10
6	MLSTOP	1	10	2	12
7	MATSTL	1	12	2	14
8	MSSRED	1	14	2	16
9	MLRTOR	1	16	2	18
10	MSSGRN	1	18	2	20
11	MCHKCF	1	20	2	22
12	MDUMIL	1	22	2	24
13	IDEDIC	2	0	2	2
14	INFLZ	2	2	2	4
15	ILUNC	2	4	2	6
16	ILYELD	2	6	2	8
17	ILSTOP	2	8	2	10
18	ICONTN	2	10	2	12
19	ICHKCF	2	12	2	14
20	IERROR	2	14	2	16

```
DEPENDENT ATTRIBUTE IDEDIC IS TRUE FOR:

  - MDEDIC

DEPENDENT ATTRIBUTE INFLZ  IS TRUE FOR:

  - MINFLZ      + MDEDIC

DEPENDENT ATTRIBUTE ILUNC  IS TRUE FOR:

  + MIUNC       + MLUNC       + MINFLZ      + MDEDIC

DEPENDENT ATTRIBUTE ILYELD IS TRUE FOR:

  + MLYELD      - MLUNC       + MINFLZ      + MDEDIC

DEPENDENT ATTRIBUTE ILSTOP IS TRUE FOR:

  + MATSTL      + MLSTOP      - MLYELD      - MLUNC       + MINFLZ
  + MDEDIC

DEPENDENT ATTRIBUTE ICONTN IS TRUE FOR:

  - MATSTL      + MLSTOP      - MLYELD      - MLUNC       + MINFLZ
  + MDEDIC

                     - OR -

  - MLRTOR      + MSSRED      - MLSTOP      - MLYELD      - MLUNC
  + MINFLZ      + MDEDIC

                     - OR -

  - MCHKCF      + MSSGRN      - MSSRED      - MLSTOP      - MLYELD
  - MLUNC       + MINFLZ      + MDEDIC

                     - OR -

  - MCHKCF      - MIUNC       + MLUNC       + MINFLZ      + MDEDIC

DEPENDENT ATTRIBUTE ICHKCF IS TRUE FOR:

  + MLRTOR      + MSSRED      - MLSTOP      - MLYELD      - MLUNC
  + MINFLZ      + MDEDIC

                     - OR -

  + MCHKCF      + MSSGRN      - MSSRED      - MLSTOP      - MLYELD
  - MLUNC       + MINFLZ      + MDEDIC

                     - OR -

  + MCHKCF      - MIUNC       + MLUNC       + MINFLZ      + MDEDIC

DEPENDENT ATTRIBUTE IERROR IS TRUE FOR:

  - MSSGRN      - MSSRED      - MLSTOP      - MLYELD      - MLUNC
  + MINFLZ      + MDEDIC
```

```
EXECUTIVE
  ROUTINE,RGEOPD,APPRO,CONFLT,LANE       ,NOATTB,PATH,SDR
  ROUTINE,RDVPRD                  ,LOGICV
  ROUTINE,OBAP  ,APPRO       ,LANE,LOGICV,NOATTB         ,VEHD,VEHF
  ROUTINE,SSOBAP                                         ,VEHD,VEHF
  ROUTINE,LOGOUT,APPRO       ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,FLGNOR                  ,LOGICV                     ,VEHF
  ROUTINE,INTERP      ,CONFLT     ,LOGICV,NOATTB,PATH    ,VEHD,VEHF
  ROUTINE,LOKIOB                  ,LOGICV       ,PATH    ,VEHD,VEHF
  ROUTINE,SSINTR                                ,PATH    ,VEHD,VEHF
  ROUTINE,CLRCON      ,CONFLT            ,NOATTB,PATH    ,VEHD,VEHF
  ROUTINE,LOGIOB,APPRO       ,LANE,LOGICV       ,PATH    ,VEHD,VEHF
  ROUTINE,IBAP  ,APPRO       ,LANE,LOGICV,NOATTB         ,VEHD,VEHF,VEHIL
  ROUTINE,LOKIBI             ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,CHKDSP,APPRO                                   ,VEHD,VEHF
  ROUTINE,CHKLDT             ,LANE                            ,VEHF
  ROUTINE,SSIBAP,APPRO            ,LOGICV                ,VEHD,VEHF
  ROUTINE,LOGIBI,APPRO       ,LANE,LOGICV       ,PATH    ,VEHD,VEHF
  ROUTINE,PREST1             ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,PREST2                  ,LOGICV                ,VEHD
  ROUTINE,UNBIAS                                         ,VEHD,VEHF
  ROUTINE,NEWVEL                  ,LOGICV                ,VEHD
  ROUTINE,LCHGEO                                         ,VEHD,VEHF
  ROUTINE,ENDLCH                                         ,VEHD,VEHF
  ROUTINE,LCHDES             ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,SVEHU              ,LANE                       ,VEHD
  ROUTINE,DELAY              ,LANE                       ,VEHD,VEHF
  ROUTINE,CKLALT             ,LANE                       ,VEHD,VEHF
  ROUTINE,GAPACC                  ,LOGICV                ,VEHD,VEHF
  ROUTINE,CHGMLN,APPRO       ,LANE,LOGICV                ,VEHD,VEHF,VEHIL
  ROUTINE,ACDCP              ,LANE,LOGICV                ,VEHD,VEHF,VEHIL
  ROUTINE,CARFOL                  ,LOGICV                ,VEHD,VEHF
  ROUTINE,ACCEL                   ,LOGICV                ,VEHD,VEHF
  ROUTINE,CRIDIS                  ,LOGICV                ,VEHD,VEHF
  ROUTINE,ADLVAI,APPRO            ,LOGICV                ,VEHD,VEHF
  ROUTINE,INTLOG                  ,LOGICV                ,VEHD,VEHF,VEHIL
  ROUTINE,SIGRES                  ,LOGICV                ,VEHD,VEHF,VEHIL
  ROUTINE,LSTOP                   ,LOGICV       ,PATH    ,VEHD,VEHF,VEHIL
  ROUTINE,CHKSDR,APPRO,CONFLT,LANE,LOGICV       ,PATH    ,VEHD,VEHF,VEHIL
  ROUTINE,CHKCON      ,CONFLT,LANE,LOGICV       ,PATH    ,VEHD,VEHF,VEHIL
  ROUTINE,SETPTV,APPRO       ,LANE              ,PATH    ,VEHD,VEHF
  ROUTINE,SETCON      ,CONFLT     ,LOGICV       ,PATH    ,VEHD,VEHF
  ROUTINE,UNSETC      ,CONFLT                   ,PATH    ,VEHD,VEHF
  ROUTINE,INFLZN             ,LANE,LOGICV                ,VEHD,VEHF,VEHIL
  ROUTINE,PATHF ,APPRO       ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,CHKMLN             ,LANE,LOGICV                ,VEHD
  ROUTINE,BANGS              ,LANE,LOGICV                ,VEHD,VEHF
  ROUTINE,BIAS                    ,LOGICV                ,VEHD,VEHF
  ROUTINE,LOGIN ,APPRO       ,LANE,LOGICV,NOATTB         ,VEHD,VEHF,VEHIL
  ROUTINE,ACTSIG                  ,LOGICV
  ROUTINE,ABORTR,APPRO,CONFLT,LANE       ,NOATTB,PATH,SDR,VEHD,VEHF,VEHIL
  EXECUTE,EXEC
TASKS

  TASK,RGEOPD
  COLEASE,REPACK,APPRO,JA
  COLEASE,REPACK,LANE,I
  COLEASE,REPACK,SDR,I
  COLEASE,REPACK,PATH,I
  COLEASE,REPACK,CONFLT,I
  COLEASE,FIND,JBLN,LANE,J,IBLN
  COLEASE,FIND,MCONTR,LANE,J,LCONTR
  COLEASE,FIND,MLANES,APPRO,LDA,NLANES
  COLEASE,FIND,JL,APPRO,LDA,LLANES(ILDLN)
  COLEASE,FIND,NLDL,LANE,JL,NLDL
  COLEASE,STORE,NLDL,LANE,JL,NLDL
  COLEASE,STORE,ID,LANE,JL,LLDL(NLDL)
  COLEASE,FIND,LGEOM3,LANE,JL,LGEOM(3)
  COLEASE,FIND,LGEOM4,LANE,JL,LGEOM(4)

  TASK,RDVPRD
```

```
TASK,OBAP
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,LANE,IL
COLEASE,REPACK,VEHD,IV
COLEASE,REPACK,VEHF,IV

TASK,SSOBAP

TASK,LOGOUT
COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
COLEASE,STORE,NOR,LANE,IL,IFVL
COLEASE,STORE,0,LANE,IL,ILVL

TASK,FLGNOR,LTF,NEWNOF
COLEASE,STORE,LTF,VEHD,NOR,MFINL
COLEASE,STORE,LTF,VEHD,NOR,MOASF
COLEASE,FIND,NSFLG,VEHD,NOR,MSFLG
COLEASE,FIND,JACC,VEHD,NOR,IACC
COLEASE,STORE,JACC,VEHD,NOR,IACC
COLEASE,STORE,LFALSE,VEHD,NOR,MSFLG
COLEASE,STORE,NEWNOF,VEHF,NOR,NOF

TASK,INTERP
COLEASE,EXTRAC,PATH,IP
COLEASE,REPACK,VEHD,IV
COLEASE,REPACK,VEHF,IV

TASK,LOKIOB
COLEASE,FIND,IVPV,LANE,LNEXT,ILVL
COLEASE,STORE,LFALSE,VEHD,IV,MFINL
COLEASE,FIND,LGEOM1,LANE,LNEXT,LGEOM(1)
COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
COLEASE,FIND,JVEL,VEHD,IVPV,IVEL
COLEASE,FIND,JACC,VEHD,IVPV,IACC

TASK,SSINTR
COLEASE,FIND,MLANES,APPRO,JSNA,NLANES
COLEASE,FIND,JL,APPRO,JSNA,LLANES(JLN)

TASK,CLRCON
COLEASE,EXTRAC,CONFLT,JGEOCP
COLEASE,STORE,NORC,CONFLT,JGEOCP,ICONV(J)
COLEASE,FIND,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,0,PATH,JP,ICPSET(JCONI)

TASK,LOGIOB
COLEASE,EXTRAC,LANE,LNEXT
COLEASE,EXTRAC,APPRO,ISNA
COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
COLEASE,STORE,NOR,PATH,LPRES,IFVP
COLEASE,STORE,0,PATH,LPRES,ILVP
COLEASE,FIND,JPOS,VEHD,NOF,IPOS
COLEASE,FIND,NOF,VEHF,NOF,NOF
COLEASE,STORE,IV,LANE,LNEXT,IFVL
COLEASE,FIND,JVEL,VEHD,NOF,IVEL
COLEASE,FIND,NOR,VEHF,NOF,NOR
COLEASE,STORE,IV,VEHF,NOF,NOR
COLEASE,STORE,IV,LANE,LNEXT,ILVL

TASK,IBAP
COLEASE,EXTRAC,APPRO,IA
COLEASE,EXTRAC,LANE,IL
COLEASE,FIND,MPRO,VEHD,IV,MPRO
COLEASE,LOGIC,VEHIL,IV
COLEASE,EXTRAC,VEHIL,IV
COLEASE,REPACK,VEHD,IV
COLEASE,REPACK,VEHF,IV
COLEASE,REPACK,VEHIL,IV

TASK,LOKIBI
COLEASE,FIND,IVPV,PATH,LNEXT,ILVP
COLEASE,STORE,LFALSE,VEHD,IV,MFINL
COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
COLEASE,FIND,JVEL,VEHD,IVPV,IVEL
COLEASE,FIND,JACC,VEHD,IVPV,IACC
COLEASE,FIND,MOBL,PATH,LNEXT,LOBL
COLEASE,FIND,IVPV,LANE,MOBL,ILVL
COLEASE,FIND,MENP,PATH,LNEXT,LENP
COLEASE,FIND,LGEOM1,LANE,MOBL,LGEOM(1)

TASK,CHKDSP
COLEASE,FIND,MIMP,PATH,LNEXT,LIMP
COLEASE,STORE,ISPD,VEHF,IV,ISPD

TASK,CHKLDT

TASK,SSIBAP,POSCHK,INQUE
COLEASE,FIND,MIMP,PATH,LNEXT,LIMP

TASK,LOGIBI
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,STORE,IV,PATH,LNEXT,ILVP
COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
COLEASE,STORE,IV,PATH,LNEXT,IFVP
COLEASE,STORE,NOR,LANE,LPREV,IFVL
COLEASE,STORE,0,LANE,LPREV,ILVL
COLEASE,FIND,MOGFLG,VEHD,NOR,LOGFLG
COLEASE,STORE,2,VEHD,NOR,LOGFLG
COLEASE,STORE,IV,VEHF,NOF,NOR
COLEASE,FIND,JVEL,VEHD,NOF,IVEL

TASK,PREST1,ININT
COLEASE,EXTRAC,VEHF,IV
COLEASE,FIND,MBLOCK,VEHD,IV,MBLOCK
COLEASE,FIND,JPOS,VEHD,NOF,IPOS
COLEASE,FIND,JVEHCL,VEHF,NOF,IVEHCL
COLEASE,FIND,JVEL,VEHD,NOF,IVEL
COLEASE,FIND,JACC,VEHD,NOF,IACC
COLEASE,FIND,MFINL,VEHD,IV,MFINL

TASK,PREST2
COLEASE,STORE,MOASF,VEHD,IV,MOASF
COLEASE,FIND,JPRTM,VEHD,IV,IPRTM
COLEASE,LOGIC,VEHD,IV
COLEASE,EXTRAC,VEHD,IV

TASK,UNBIAS

TASK,NEWVEL,T,TSQ,TCU

TASK,LCHGEO

TASK,ENDLCH
COLEASE,FIND,MCHGE,VEHD,NOF,LCHGE
COLEASE,FIND,MCHGE,VEHD,NOR,LCHGE
COLEASE,STORE,MCHGE,VEHD,NOR,LCHGE

TASK,LCHDES
COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
COLEASE,FIND,LGEOM3,LANE,LANSI,LGEOM(3)
COLEASE,FIND,LGEOM4,LANE,LANSI,LGEOM(4)
COLEASE,FIND,LGEOM1,LANE,LANSI,LGEOM(1)
COLEASE,FIND,LGEOM2,LANE,LANSI,LGEOM(2)

TASK,SVEHD,NOW
COLEASE,FIND,LGEOM4,LANE,LANSI,LGEOM(4)
COLEASE,FIND,NOSF,LANE,LANSI,IFVL
COLEASE,FIND,IPOSF,VEHD,NOSF,IPOS
```

```
COLEASE,FIND,LEADSP,VEHD,NOSF,IVEL
COLEASE,FIND,MEGAL,VEHD,NOSF,LEGAL
COLEASE,FIND,NOSR,VEHF,NOSF,NOR
COLEASE,FIND,IPOSR,VEHD,NOSR,IPOS
COLEASE,FIND,LAGSPD,VEHD,NOSR,IVEL
COLEASE,FIND,MEGAL,VEHD,NOSR,LEGAL

TASK,DELAY
COLEASE,FIND,JTURN,VEHF,NOF,ITURN
COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
COLEASE,FIND,JTURN,VEHF,NOSF,ITURN
COLEASE,FIND,JTURN,VEHF,NOSF,ITURN

TASK,CKLALT
COLEASE,FIND,MPINT,LANE,NLR,NPINT
COLEASE,FIND,IPATH,LANE,NLR,LINTP(I)
COLEASE,FIND,JLCH,PATH,IPATH,ILCH
COLEASE,FIND,MOBAP,PATH,IPATH,LOBAP
COLEASE,FIND,MPINT,LANE,NLL,NPINT
COLEASE,FIND,IPATH,LANE,NLL,LINTP(I)
COLEASE,FIND,JLCH,PATH,IPATH,ILCH
COLEASE,FIND,MOBAP,PATH,IPATH,LOBAP

TASK,GAPACC,LANSI
COLEASE,FIND,MCONTR,LANE,LANSI,LCONTR
COLEASE,FIND,JBLN,LANE,LANSI,IBLN
COLEASE,FIND,JACC,VEHD,NOSF,IACC
COLEASE,FIND,JVEHCL,VEHF,NOSF,IVEHCL
COLEASE,FIND,JACC,VEHD,NOSR,IACC
COLEASE,FIND,JSET,VEHD,NOSR,ISET
COLEASE,STORE,LTRUE,VEHD,NOSR,MLAG

TASK,CHGMLN
COLEASE,STORE,NVILL,APPRO,IA,NVIL(ILN)
COLEASE,STORE,NOR,LANE,LPRES,IFVL
COLEASE,STORE,NOR,VEHF,NOF,NOR
COLEASE,STORE,NOF,LANE,LPRES,ILVL
COLEASE,FIND,JVEL,VEHD,NOF,IVEL
COLEASE,STORE,NOASF,VEHD,NOR,MOASF
COLEASE,STORE,NVILL,APPRO,IA,NVIL(JLN)
COLEASE,STORE,IV,LANE,LPRES,IFVL
COLEASE,FIND,JSET,VEHD,NOR,ISET
COLEASE,FIND,MEGAL,VEHD,NOR,LEGAL
COLEASE,STORE,5,VEHD,NOR,ISET
COLEASE,STORE,IV,VEHF,NOF,NOR
COLEASE,FIND,JVEL,VEHD,NOF,IVEL
COLEASE,STORE,IV,LANE,LPRES,ILVL
COLEASE,STORE,3,VEHD,NOR,LCHGE
COLEASE,FIND,MWID,LANE,LPRES,LWID
COLEASE,FIND,LGEOM2,LANE,LPRES,LGEOM(2)
COLEASE,FIND,LGEOM4,LANE,LPRES,LGEOM(4)
COLEASE,FIND,LCONTR,LANE,LPRES,LCONTR
COLEASE,FIND,IBLN,LANE,LPRES,IBLN

TASK,ACDCP

TASK,CARFOL
COLEASE,FIND,LATNOW,VEHD,NOF,LATPOS
COLEASE,FIND,LAT2GO,VEHD,NOF,LEGAL

TASK,ACCEL

TASK,CRIDIS,K

TASK,ADLVAI
COLEASE,FIND,MPRES,VEHF,JV,LPRES
COLEASE,FIND,JSNA,LANE,MPRES,ISNA

TASK,INTLOG
COLEASE,FIND,NPRO,VEHD,NOF,MPRO
```

```
TASK,SIGRES,JSISET
COLEASE,FIND,JLCH,PATH,LNEXT,ILCH
COLEASE,FIND,NPRO,VEHD,NOF,MPRO

TASK,LSTOP
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,FIND,NPRO,VEHD,JV,MPRO
COLEASE,FIND,NLUNC,VEHIL,JV,MLUNC
COLEASE,FIND,MNEXT,VEHF,JV,LNEXT
COLEASE,FIND,MCPSET,PATH,MNEXT,NCPSET
COLEASE,FIND,MOGFLG,VEHD,JV,LOGFLG
COLEASE,FIND,ICONP1,CONFLT,JNDEX,ICONP(1)
COLEASE,FIND,ICONP2,CONFLT,JNDEX,ICONP(2)

TASK,CHKSDR
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,FIND,JVEL,VEHD,ILVP,IVEL
COLEASE,EXTRAC,CONFLT,JNDEX
COLEASE,FIND,KSPD,VEHF,IVPV,ISPD
COLEASE,FIND,JSLIM,APPRO,JA,ISLIM
COLEASE,FIND,JCANSE,SDR,ISDR,ICANSE(IPNDEX)
COLEASE,FIND,JL,PATH,JP,LIBL
COLEASE,FIND,LGEOM4,LANE,JL,LGEOM(4)
COLEASE,FIND,KCANSE,SDR,ISDR,ICANSE(IPNDEX)

TASK,CHKCON
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,EXTRAC,CONFLT,JNDEX
COLEASE,FIND,JL,PATH,JP,LIBL
COLEASE,FIND,MGEOM4,LANE,JL,LGEOM(4)
COLEASE,FIND,NININT,VEHD,NOFC,MININT
COLEASE,FIND,NOFC,LANE,JL,IFVL
COLEASE,FIND,KSPD,VEHF,IVPV,ISPD
COLEASE,FIND,JVEL,VEHD,ILVP,IVEL
COLEASE,FIND,MORC,VEHD,NOFC,NORC
COLEASE,FIND,MOR,VEHF,NOFC,NOR
COLEASE,FIND,NOFC,LANE,JL,IFVL
COLEASE,FIND,JSLP,VEHD,IVCONF,ISLP
COLEASE,FIND,JACC,VEHD,IVCONF,IACC
COLEASE,FIND,JVEL,VEHD,IVCONF,IVEL
COLEASE,FIND,JPOS,VEHD,IVCONF,IPOS
COLEASE,FIND,NININT,VEHD,IVCONF,MININT
COLEASE,FIND,JSPD,VEHF,IVCONF,ISPD
COLEASE,FIND,JSPDP,VEHD,IVCONF,ISPDP
COLEASE,FIND,JSNA,LANE,JL,ISNA
COLEASE,FIND,KPRTM,VEHD,IVCONF,IPRTM
COLEASE,FIND,MIMP,PATH,JP,LIMP
COLEASE,FIND,JSLIM,APPRO,JSNA,ISLIM
COLEASE,FIND,JD,VEHF,IVCONF,IDRICL
COLEASE,FIND,JV,VEHF,IVCONF,IVEHCL
COLEASE,FIND,JFVA,VEHD,IVCONF,IFVA
COLEASE,FIND,IVCONF,VEHD,NOFC,NORC
COLEASE,STORE,LFALSE,VEHIL,IV,MCHKCF

TASK,SETPTV

TASK,SETCON
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,FIND,MOGFLG,VEHD,NOR,LOGFLG
COLEASE,STORE,MOGFLG,VEHD,NOR,LOGFLG
COLEASE,EXTRAC,CONFLT,JGEOCP
COLEASE,FIND,JCPSET,PATH,JP,ICPSET(JCONI)
COLEASE,STORE,IV,CONFLT,JGEOCP,ICONV(J)
COLEASE,FIND,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,1,PATH,JP,ICPSET(JCONI)
COLEASE,FIND,MORC,VEHD,INOW,NORC
COLEASE,FIND,JPOS,VEHD,INOW,IPOS
COLEASE,FIND,NININT,VEHD,INOW,MININT
COLEASE,STORE,IV,CONFLT,JGEOCP,ICONV(J)
COLEASE,STORE,IV,VEHD,NOFC,NORC
```

```
COLEASE,STORE,IV,VEHD,INOR,NORC

TASK,UNSETC
COLEASE,EXTRAC,PATH,LNEXT
COLEASE,EXTRAC,CONFLT,JGEOCP
COLEASE,STORE,NORC,CONFLT,JGEOCP,ICONV(J)
COLEASE,STORE,Ø,CONFLT,JGEOCP,ICONV(J)
COLEASE,FIND,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,MCPSET,PATH,JP,NCPSET
COLEASE,STORE,Ø,PATH,JP,ICPSET(JCONI)
COLEASE,FIND,MORC,VEHD,NOFC,NORC
COLEASE,STORE,NORC,VEHD,NOFC,NORC

TASK,INFLZN
COLEASE,FIND,JLCH,PATH,LNEXT,ILCH

TASK,PATHF,IFORCE,NN1,NN2
COLEASE,EXTRAC,LANE,LPRES
COLEASE,FIND,JOPT,PATH,LPATH,IOPT
COLEASE,FIND,MOBAP,PATH,LPATH,LOBAP
COLEASE,FIND,JPT,PATH,LPATH,IPT
COLEASE,FIND,MPINT,LANE,ILANE,NPINT
COLEASE,FIND,LPATH,LANE,ILANE,LINTP(I)
COLEASE,FIND,JOPT,PATH,LPATH,IOPT
COLEASE,FIND,MOBAP,PATH,LPATH,LOBAP
COLEASE,FIND,JPT,PATH,LPATH,IPT
COLEASE,STORE,ITURN,VEHF,IV,ITURN
COLEASE,STORE,LPATH,VEHF,IV,LNEXT
COLEASE,STORE,MOBAP,VEHF,IV,NOBAPD
COLEASE,EXTRAC,LANE,IL

TASK,CHKMLN

TASK,BANGS,IWHERE
COLEASE,FIND,MPRES,VEHF,IVPV,LPRES
COLEASE,FIND,NININT,VEHD,IVPV,MININT
COLEASE,FIND,JA,LANE,MPRES,ISNA
COLEASE,FIND,MLANES,APPRO,JA,NLANES
COLEASE,FIND,JL,APPRO,JA,LLANES(JLN)
COLEASE,FIND,MOF,VEHF,IVPV,NOF
COLEASE,FIND,MOR,VEHF,IVPV,NOR
COLEASE,FIND,MORC,VEHD,IVPV,NORC
COLEASE,FIND,JPOS,VEHD,IVPV,IPOS
COLEASE,FIND,JSLP,VEHD,IVPV,ISLP
COLEASE,FIND,JSPD,VEHF,IVPV,ISPD
COLEASE,FIND,JVEHCL,VEHF,IVPV,IVEHCL
COLEASE,FIND,JDRICL,VEHF,IVPV,IDRICL
COLEASE,FIND,MNEXT,VEHF,IVPV,LNEXT
COLEASE,FIND,MOBAPD,VEHF,IVPV,NOBAPD
COLEASE,FIND,JSET,VEHD,IVPV,ISET
COLEASE,FIND,MEGAL,VEHD,IVPV,LEGAL
COLEASE,FIND,MOGFLG,VEHD,IVPV,LOGFLG
COLEASE,FIND,MCHGE,VEHD,IVPV,LCHGE
COLEASE,FIND,KPRTM,VEHD,IVPV,IPRTM
COLEASE,FIND,JTURN,VEHF,IVPV,ITURN
COLEASE,FIND,JBAPS,VEHF,IVPV,IBAPS
COLEASE,FIND,MATPOS,VEHD,IVPV,LATPOS
COLEASE,FIND,JSTCON,VEHD,IVPV,ISTCON

TASK,BIAS

TASK,LOGIN
COLEASE,STORE,IV,LANE,IL,IFVL
COLEASE,STORE,IV,VEHF,NOF,NOR
COLEASE,STORE,IV,LANE,IL,ILVL
COLEASE,STORE,NVILL,APPRO,ISNA,NVIL(ILN)
COLEASE,REPACK,VEHD,IV
COLEASE,REPACK,VEHF,IV
COLEASE,REPACK,VEHIL,IV
COLEASE,STORE,NOF,LANE,IL,ILVL
COLEASE,STORE,Ø,LANE,IL,IFVL

COLEASE,STORE,Ø,VEHF,NOF,NOR

TASK,ACTSIG

TASK,ABORTR,MSG,NCHS
TERMINATE
```

APPENDIX F

DATA COLLECTION AND REDUCTION PROGRAMS

This page intentionally left blank to facilitate printing on 2 sides.

```
ASMB,R,B              PR184     PR18416
      HED PR184 - PROJECT 184 HP2115 A/D CONVERSION PROGRAM
      NAM PR184
      ENT EOP
      ENT RESTR
      EXT PMTTY
      EXT INCOM
      EXT LOCAL
      EXT WRING
      EXT POS30
      EXT PRUN
      EXT FDMAI
      EXT DMAIN
      EXT UNPAK
      EXT FILTR
      EXT HUNT
      EXT DMAOT
      EXT WAITA
      EXT .IOC.
      SPC 2
*
***** PR184 - PROJECT 184 HP2115 A/D CONVERSION PROGRAM
*
***** WRITES A 60 CHARACTER IDENT ONTO THE 9 TRACK TAPE
*     READS FROM 1 TO 6 CASSETTES THROUGH THE +16 BIT DUPLEX REGS
*     UNPACKS THE INPUT BUFFER
*     OPTIONALLY FILTERS THE INPUT SAMPLES FOR NOISE
*     HUNTS FOR 0/1 BIT OR EOR IN THE SAMPLES
*     WRITES 10 SETS OF 3 16 BIT WORDS ONTO THE 9 TRACK TAPE:
*           WORD 1 = 16 BIT TAPE NUMBER (0 FOR END OF DATA)
*           WORD 2 = UPPER 16 BITS OF 32 BIT DATA
*           WORD 3 = LOWER 16 BITS OF 32 BIT DATA
*     WRITES 2 END-OF-FILE MARKS AFTER DATA AS AN END-OF-DATA MARK
*
***** HLT CODES:
*     00  PR184      END OF PR184
*     01  PR184      WAITING FOR PRESET AND RUN AFTER INITIALIZATION
*
***** SWITCH REGISTER CODES:
*     SS00 = 0 = CONTINUE PROGRAM
*          = 1 = HALT PROGRAM AT END OF PROCESSING CURRENT INPUT BUFFER
*     SS01 = 0 = FILTER INPUT DATA
*          = 1 = DO NOT FILTER INPUT DATA
*     SS02 = 0 = SKIP 32 BITS ON ERROR
*          = 1 = HALT PROGRAM ON ERROR
*     SS03 = 0 = CONTINUE PROGRAM NORMALLY
*          = 1 = WRITE END-OF-DATA MARK ON 9 TRACK TAPE
*     SS04 = 0 = CONTINUE PROGRAM NORMALLY
*          = 1 = BACKSPACE TO LAST EOF AND WRITE EOD ON 3030
*
***** TABS            7,11,21,31
*
      SPC 2
      ORB
IBUFI BSS 16          DMA CH 1 BUFFERS - DMA STORE
IBUFO BSS 16                           - PROGRAM READ
OBUFI BSS 30          DMA CH 2 BUFFERS - PROGRAM STORE
OBUFO BSS 30                           - DMA WRITE
      ORR
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NBP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
      ORB
XBUFI DEF IBUFI
XBUFO DEF IBUFO
YBUFI DEF OBUFI
YBUFO DEF OBUFO
```

```
      ORR
      SPC 2
START HLT 00B         END OF PROGRAM - WAIT FOR RE-RUN
PR184 NOP             START OF PROGRAM
      JSB PMTTY       WRITE BLANK LINE ON TTY
      DEF BLANK                 BLANK LINE ADDRESS
      DEC 1                     LINE LENGTH = 2 CHARACTERS
      JSB PMTTY       =UT CASSETTE TO HP 9 TRACK A/D CONVERSION PROGRAM=
      DEF MSG1                  MESSAGE 1 ADDRESS
      DEF 24                    MESSAGE 1 LENGTH = 48 CHARACTERS
      LIA 01B         GET SWITCH REGISTER
      RAR,RAR         POSITION SS03 IN B00
      RAR
      SLA             IF SS03=1 THEN GO TO WEOD AND WRITE EOD ON 3030
      JMP WEOD
      RAR             POSITION SS04 IN B00
      SLA             IF SS04=1 THEN GO TO RESTR
      JMP RESTR
      JSB INCOM       INITIALIZE COMMON
      LDA XBUFI       STORE DMA1 BUFFER ADDRESSES IN COMMON
      STA ABUFI
      LDA XBUFO
      STA ABUFO
      LDA YBUFI       STORE DMA2 BUFFER ADDRESSES IN COMMON
      STA BBUFI
      LDA YBUFO
      STA BBUFO
      JSB LOCAL       CHECK FOR 3030 IN LOCAL
      DEF *+2                   RETURN ADDRESS
      DEF ICC30                 3030 COMMAND CHANNEL
      JSB WRING       CHECK FOR 3030 WRITE WRING
      DEF *+2                   RETURN ADDRESS
      DEF ICC30                 3030 COMMAND CHANNEL
      JSB POS30       POSITION 3030 TAPE AND WRITE 60 CHARACTER IDENT
      JSB PRUN        PRINT =PRESS =PRESET= AND =RUN= WHEN READY= ON TTY
      CLA             CLEAR A AND B REG
      CLB
      HLT 01B         HALT AND WAIT FOR PRESET AND RUN
      CLC 00B         CLEAR ALL CONTROL BITS
      CLF 00B         DISABLE INTERRUPT SYSTEM
      JSB FDMAI       INITIALIZE INPUT FROM DUPLEX REGISTERS
LOOP1 JSB DMAIN       INPUT FROM DUPLEX REGISTERS
      JSB UNPAK       UNPACK THE INPUT BUFFER
      LIA 01B         GET SWITCH REGISTER
      RAR             POSITION SS01 IN B00
      SLA,RSS         IF SS01=0 THEN FILTER DATA
      JSB FILTR       FILTER 32 BIT SAMPLE FOR NOISE
      JSB HUNT        HUNT FOR 0/1 BIT OR EOR IN SAMPLE
      LIA 01B         GET SWITCH REGISTER
      SLA,RSS         IF SS00=0 THEN GO TO LOOP1 AND CONTINUE PROCESSING DATA
      JMP LOOP1
      LDA BBUFI       SET QBUFI FOR OBUFI(MBUFI)
      ADA MBUFI
      STA QBUFI
      CLB             SET B REG=0
LOOP2 LDA MBUFI       IF MBUFI=30 THEN GO TO CALL1
      CPA D30
      JMP CALL1
      STB QBUFI,I     SET OBUFI(MBUFI)=0
      ISZ MBUFI       MBUFI=MBUFI+1
      ISZ QBUFI       SET QBUFI FOR OBUFI(MBUFI+1)
      JMP LOOP2       GO TO LOOP2
CALL1 JSB DMAOT       WRITE OUT LAST OUTPUT BUFFER ONTO 3030
      SFS 07B         WAIT UNTIL DMA FINISHED
      JMP *-1
      SFS 16B         WAIT UNTIL 3030 FINISHED
      JMP *-1
WEOD  CLC 00B         CLEAR ALL CONTROL BITS
CALL2 JSB .IOC.       WRITE END-OF-FILE MARK ON 3030
      OCT 030112                IOC WRITE FILE MARK REQUEST
      JMP CALL2                 RE-SUBMIT IF BUSY
```

```
CALL3 JSB .IOC.      WRITE END-OF-FILE MARK ON 3030
      OCT 030112               IOC WRITE FILE MARK REQUEST
      JMP CALL3                RE-SUBMIT IF BUSY
      JSB WAITA      WAIT UNTIL 3030 AVAILABLE
      DEF *+2                  RETURN ADDRESS
      DEF IOC30                3030 IOC SELECT CODE
EOP   JSB PMTTY      "END OF PROGRAM - RUN AGAIN"
      DEF MSG2                 MESSAGE 2 ADDRESS
      DEC 13                   MESSAGE 2 LENGTH - 26 CHARACTERS
      JSB PRUN       PRINT "PRESS "PRESET" AND "RUN" WHEN READY" ON TTY
      CLA            CLEAR A AND B REGISTERS
      CLB
      JMP START      GO TO START AND END PROGRAM
RESTR JSB .IOC.      BACKSPACE 3030 1 RECORD
      OCT 030212               IOC BACKSPACE ONE RECORD REQUST
      JMP RESTR                RE-SUBMIT IF BUSY
      JSB WAITA      WAIT UNTIL 3030 AVAILABLE
      DEF *+4                  RETURN ADDRESS
      DEF IOC30                3030 IOC SELECT CODE
      DEF IRSTA                3030 STATUS ADDRESS
      DEF IRTRL                3030 TRANSMISSION LOG ADDRESS
      AND MSOT       IF SOT ON 3030 THEN GO TO START
      SZA
      JMP EOP        GO TO EOP AND END PROGRAM
      LDA IRSTA      IF EOF ON 3030 THEN GO TO WEOD
      AND MEOF       MASK OUT EOF BIT
      SZA
      JMP WEOD       GO TO WEOD AND WRITE EOD MARK
      JMP RESTR      GO TO RESTR AND BACKSPACE AGAIN
      SPC 2
      ORB
BLANK OCT 020040
D30   DEC 30
ICC30 OCT 000016
IOC30 OCT 000012
IRSTA BSS 1
IRTRL BSS 1
MEOF  OCT 000200
MSOT  OCT 000100
MSG1  ASC 24,UT CASSETTE TO HP 9 TRACK A/D CONVERSION PROGRAM
MSG2  ASC 13,END OF PROGRAM - RUN AGAIN
QBUFI BSS 1
      END PR184
```

```
ASMB,R,B               INCOM      PR18416
      HED INCOM - INITIALIZE COMMON
      NAM INCOM
      ENT INCOM
      EXT PMTTY
      EXT .DIO.
      EXT .IOI.
      SPC 2
*
***** INCOM - INITIALIZED COMMON
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
      ORB
FIRST DEF NBIT
ISAVE DEF NSAVE
LAST  DEF NULL
      ORR
      SPC 2
INCOM BSS 1          ENTRY/EXIT LINE
      LDA FIRST      SET NCOM FOR FIRST WORD OF COMMON
      STA NCOM
      CLA            SET A REG=0
LOOP1 STA NCOM,I     SET COMMON(I)=0
      LDB NCOM       IF NCOM EQ LAST THEN GO TO DONE1
      CPB LAST
      JMP DONE1
      ISZ NCOM       SET NCOM FOR COMMON(I+1)
      JMP LOOP1      GO TO LOOP1
DONE1 JSB PMTTY      "ENTER NUMBER OF TAPES - MAX = 6"
      DEF MSG1                 MESSAGE 1 ADDRESS
      DEC 16                   MESSAGE 1 LENGTH - 31 CHARACTERS
TYPE1 LDA INPUT      READ INTEGER FROM KEYBOARD INPUT
      CLB,INB
      JSB .DIO.
      ABS 0                    FREE FIELD
      DEF *+3                  RETURN ADDRESS
      JSB .IOI.                STORE INTEGER AS NTAPE
      STA NTAPE
      CCA            IF NTAPE GE 1 THEN GO TO TEST1
      ADA NTAPE
      SSA,RSS
      JMP TEST1
      JSB PMTTY      "NTAPE LT 1 - RE-ENTER"
      DEF MSG2                 MESSAGE 2 ADDRESS
      DEC 11                   MESSAGE 2 LENGTH - 21 CHARACTERS
      JMP TYPE1      GO TO TYPE1 AND RE-ENTER NTAPE
TEST1 LDA NTAPE      IF NTAPE LE 6 THEN GO TO LABL2
      CMA,INA
      ADA D6
      SSA,RSS
      JMP LABL2
      JSB PMTTY      "NTAPE GT 6 - RE-ENTER"
      DEF MSG3                 MESSAGE 3 ADDRESS
      DEC 11                   MESSAGE 3 LENGTH - 21 CHARACTERS
      JMP TYPE1      GO TO TYPE1 AND RE-ENTER NTAPE
LABL2 LDA ISAVE      SET JSAVE FOR NSAVE(1)
      STA JSAVE
      ADA D9         SET KSAVE FOR NSAVE(10)
      STA KSAVE
LOOP2 ISZ JSAVE,I    SET NSAVE(I)=1
      LDA JSAVE      IF JSAVE EQ KSAVE THEN GO TO DONE2
      CPA KSAVE
      JMP DONE2
      ISZ JSAVE      SET JSAVE FOR NSAVE(I+1)
      JMP LOOP2      GO TO LOOP2
```

```
DONE2 LDA DEC32     SET D32=32
      STA D32
      LDA DEC16     SET D16=16
      STA D16
      LDA D13       SET NSAMP=13
      STA NSAMP
      LDA NSAMP     SET NBUFO=NSAMP
      STA NBUFO
      ADA D3        SET NBUFI=NSAMP+3
      STA NBUFI
      LDA D2        SET NULL=2
      STA NULL
      JMP INCOM,I   RETURN
      SPC 2
      ORB
D2    DEC 2
D3    DEC 3
D6    DEC 6
D9    DEC 9
D13   DEC 13
DEC16 DEC 16
DEC32 DEC 32
INPUT OCT 000001    KEYBOARD INPUT
JSAVE BSS 1
KSAVE BSS 1
MSG1  ASC 16,ENTER NUMBER OF TAPES - MAX = 6
MSG2  ASC 11,NTAPE LT 1 - RE-ENTER
MSG3  ASC 11,NTAPE GT 6 - RE-ENTER
NCOM  BSS 1
      END
```

```
ASMB,R,B                POS30     PR18416
      HED POS30 - POSITION 3030 TAPE AND WRITE 60 CHARACTER IDENT
      NAM POS30
      ENT POS30
      EXT PMTTY
      EXT .DIO.
      EXT .IOI.
      EXT .IOC.
      EXT LOCAL
      EXT WAITA
      EXT PRUN
      EXT EOP
      SPC 2
*
***** POS30 - POSITION 3030 TAPE AND WRITE 60 CHARACTER IDENT
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
POS30 BSS 1             ENTRY/EXIT LINE
      JSB PMTTY         =ENTER 60 CHARACTERS FOR IDENT=
      DEF MSG1                   MESSAGE 1 ADDRESS
      DEC 15                     MESSAGE 1 LENGTH - 29 CHARACTERS
      LDA BBUFI         SET BUFFER ADDRESS FOR IOC CALLS
      STA DEFI          STORE IN IOC READ CALL
      STA DEFJ          STORE IN IOC WRITE CALL
      STA ADDR          SET ADDR FOR BBUFI(1)
      LDA D30           SET COUNT=-30
      CMA,INA
      STA COUNT
      LDA BLANK         GET ASCII CODE FOR 2 8-BIT BLANK CHARACTERS
LOOP1 STA ADDR,I        BLANK IDENT BUFFER
      ISZ ADDR          SET ADDR FOR BBUFI(I+1)
      ISZ COUNT         COUNT=COUNT+1 THEN IF COUNT=0 SKIP NEXT INSTRUCTION
      JMP LOOP1         GO TO LOOP1
CALL1 JSB .IOC.         READ 60 ASCII CHARACTERS FROM TTY
      OCT 010401                 IOC READ ASCII FROM TTY AND ECHO-PRINT REQUEST
      JMP CALL1                  IF BUSY THEN RE-SUBMIT
DEFI  DEF BBUFI,I                BUFFER ADDRESS
      DEC -60                    BUFFER LENGTH - 60 CHARACTERS
      JSB PMTTY         =ENTER [-1,0,+1] FOR [CONTINUE,START-OF-TAPE,END-OF-DATA]=
      DEF MSG2                   MESSAGE 2 ADDRESS
      DEC 28                     MESSAGE 2 LENGTH - 56 CHARACTERS
      LDA INPUT         READ INTEGER FROM KEYBOARD INPUT
      CLB,INB
      JSB .DIO.
      ABS 0                      FREE FIELD
      DEF *+3                    RETURN ADDRESS
      JSB .IOI.                  STORE INTEGER AS N
      STA N
      SSA               IF N LT 0 THEN GO TO CALL4
      JMP CALL4
CALL2 JSB .IOC.         REWIND 3030
      OCT 030412                 IOC REWIND TO LOAD POINT AND READY REQUEST
      JMP CALL2                  IF BUSY THEN RE-SUBMIT
      LDA N             IF N=0 THEN GO TO WRTID
      SZA,RSS
      JMP WRTID
LABL1 LDA DM2           SET EOF=-2
      STA EOF
LABL2 JSB LOCAL         CHECK 3030 IN LOCAL
      DEF *+2                    RETURN ADDRESS
      DEF ICC30                  3030 COMMAND CHANNEL
CALL3 JSB .IOC.         FORWARD SPACE 3030 ONE RECORD
      OCT 030312                 IOC FORWARD SPACE ONE RECORD REQUEST
      JMP CALL3                  IF BUSY THEN RE-SUBMIT
      JSB WAITA         WAIT UNTIL 3030 AVAILABLE
```

```
      DEF *+4              RETURN ADDRESS
      DEF IOC30            IOC 3030 SELECT CODE
      DEF STA30            3030 STATUS WHEN AVAILABLE
      DEF TRLOG            3030 TRANSMISSION LOG WHEN AVAILABLE
      AND MEOT      MASK OUT EOT BIT
      SZA           IF EOT THEN GO TO ERROR
      JMP ERROR
      LDA STA30     GET STATUS OF 3030
      AND MEOF      MASK OUT EOF BIT
      SZA,RSS       IF NOT EOF THEN GO TO LABL1
      JMP LABL1
      ISZ EOF       EOF=EOF+1 THEN IF EOF=0 SKIP NEXT INSTRUCTION
      JMP LABL2     GO TO LABL2
CALL4 JSB .IOC.     BACKSPACE 3030 ONE RECORD
      OCT 030212           IOC BACKSPACE ONE RECORD REQUEST
      JMP CALL4            IF BUSY THEN RE-SUBMIT
WRTID JSB .IOC.     WRITE 60 ASCII CHARACTER IDENT ONTO 3030
      OCT 020112           IOC WRITE BINARY RECORD REQUEST
      JMP WRTID            IF BUSY THEN RE-SUBMIT
DEFJ  DEF BBUFI,I          BUFFER ADDRESS
      DEC -60              BUFFER LENGTH = 60 CHARACTERS
      JSB WAITA     WAIT UNTIL 3030 AVAILABLE
      DEF *+2              RETURN ADDRESS
      DEF IOC30            IOC 3030 SELECT CODE
      JMP POS30,I   RETURN
ERROR JSB PMTTY     =END-OF-TAPE ON 3030 - PR184 RESTARTED=
      DEF MSG3             MESSAGE 3 ADDRESS
      DEC 19               MESSAGE 3 LENGTH = 37 CHARACTERS
      JSB PRUN      PRINT =PRESS =PRESET= AND =RUN= WHEN READY= ON TTY
      CLA           CLEAR A AND B REG
      CLB
      JMP EOP       GO TO EOP AND RESTART PR184
      SPC 2
      ORB
ADDR  BSS 1
BLANK OCT 020040
COUNT BSS 1
D30   DEC 30
DM2   DEC -2
EOF   BSS 1
ICC30 OCT 000016
IOC30 OCT 000012
INPUT OCT 000001    KEYBOARD INPUT
MEOF  OCT 000200    0/000/000/010/000/000
MEOT  OCT 000040    0/000/000/000/100/000
MSG1  ASC 15,ENTER 60 CHARACTERS FOR IDENT
MSG2  ASC 28,ENTER [-1,0,+1] FOR [CONTINUE,START-OF-TAPE,END-OF-DATA]
MSG3  ASC 19,END-OF-TAPE ON 3030 - PR184 RESTARTED
N     BSS 1
STA30 BSS 1
TRLOG BSS 1
      END
```

```
ASMB,R,B                 DMAIN      PR18416
      HED DMAIN  - INPUT FROM +16 BIT DUPLEX REGISTERS
      NAM DMAIN
      ENT FDMAI
      ENT DMAIN
      EXT PMTTY
      EXT PRUN
      EXT RESTR
      SPC 2
*
***** DMAIN - INPUT FROM +16 BIT DUPLEX REGISTERS
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
FDMAI BSS 1         ENTRY/EXIT LINE
      LDA FDMAI     GET RETURN ADDRESS
      STA DMAIN     STORE RETURN ADDRESS AT MAIN START POINT
      JMP LABL1     GO TO LABL1
      SPC 2
DMAIN BSS 1         ENTRY/EXIT LINE
      LDA NBUFI     SWITCH BUFFER LENGTHS
      LDB NBUFO
      STA NBUFO
      STB NBUFI
      LDA ABUFI     SWITCH BUFFER ADDRESSES
      LDB ABUFO
      STA ABUFO
      STB ABUFI
      SFC 06B       IF LAST DMA FINISHED THEN GO TO ERROR
      JMP ERROR
      SFS 06B       WAIT UNTIL LAST DMA FINISHED
      JMP *-1
LABL1 LDA PCW       GET DMA PROGRAM CONTROL WORD
      OTA 06B       SEND TO DMA CH 1
      CLC 02B       PREPARE DMA CH 1 MEMORY ADDRESS REGISTER
      LDA ABUFI     GET DMA ADDRESS WORD
      IOR ONE       SET ADDRESS FOR INPUT
      OTA 02B       SEND TO DMA CH 1
      STC 02B       PREPARE DMA CH 1 WORD COUNT RECORD
      LDA NBUFI     GET BUFFER LENGTH
      CMA,INA       SET NEGATIVE
      OTA 02B       SEND TO DMA CH 1
      STC 17B,C     INITIATE +16 BIT DUPLEX REGISTERS
      STC 06B,C     ACTIVATE DMA CH 1
      JMP DMAIN,I   RETURN
ERROR CLC 00B       CLEAR ALL CONTROL BITS
      JSB PMTTY     =INPUT DATA RATE TOO FAST=
      DEF MSG1             MESSAGE 1 ADDRESS
      DEC 12               MESSAGE 1 LENGTH = 24 CHARACTERS
      JSB PMTTY     =SET SS15 TO [0,1] FOR [RESTART,CONTINUE]=
      DEF MSG2             MESSAGE 2 ADDRESS
      DEC 20               MESSAGE 2 LENGTH = 40 CHARACTERS
      JSB PRUN      PRINT =PRESS =PRESET= AND =RUN= WHEN READY= ON TTY
      CLA           CLEAR A AND B REG
      CLB
      HLT 00B       HALT AND WAIT FOR PRESET AND RUN
      LIA 01B       GET SWITCH REGISTER
      SSA           IF SS15=1 THEN GO TO LABL1 AND CONTINUE
      JMP LABL1
      JMP RESTR     GO TO RESTR AND RESTART PR184
      SPC 2
      ORB
MSG1  ASC 12,INPUT DATA RATE TOO FAST
MSG2  ASC 20,SET SS15 TO [0,1] FOR [RESTART,CONTINUE]
ONE   OCT 100000    ADDRESS WORD CODE FOR INPUT
PCW   OCT 120017    STC,CLC,+16 BIT DUPLEX REGISTERS
```

```
END
```

```
ASMB,R,B               UNPAK      PR18416
      HED UNPAK - UNPACK THE INPUT BUFFER
      NAM UNPAK
      ENT UNPAK
      EXT PMTTY
      EXT PRUN
      EXT RESTR
      SPC 2
*
***** UNPAK - UNPACK THE INPUT BUFFER
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
      ORB
ICB   DEF NCB
ISP   DEF NSP
      ORR
      SPC 2
A     EQU 0
B     EQU 1
      SPC 2
UNPAK BSS 1            ENTRY/EXIT LINE
      LDA ICB          SET JCB FOR NCB(1)
      STA JCB
      LDA ISP          SET JSP FOR NSP(1)
      STA JSP
      CLA              SET I=0
      STA I
LOOP1 LDA I            IF I EQ NTAPE THEN GO TO DONE1
      CPA NTAPE
      JMP DONE1
      LDA ABUFO        SET KBUFO FOR IBUFO(1)
      STA KBUFO
      LDA JSP,I        SET J=NSP(I)
      STA J
      ADA NBUFO        SET JO=NSP(I)+NBUFO
      STA JO
      CMA,INA          IF JO GT 16 THEN GO TO ERROR
      ADA D16
      SSA
      JMP ERROR
LOOP2 LDB J            IF J=16 THEN GO TO DONE2
      CPB D16
      JMP DONE2
      LDA JO           IF J GE JO THEN GO TO LABL1
      CMA,INA
      ADB A
      CLE              SET E REG=0 FOR ZERO FILL FOR JUMP
      SSB,RSS
      JMP LABL1
      ISZ JSP,I        NSP(I)=NSP(I)+1
      LDA KBUFO,I      GET IBUFO(J)
      ERA              ROTATE RIGHT 1 BIT AND SET E REG=B00
      STA KBUFO,I      STORE IBUFO(J) FOR NEXT TIME
LABL1 LDA JCB,I        GET NCB(I)
      ELA              ROTATE A REG LEFT 1 BIT AND OR IN E REG TO B00
      STA JCB,I        STORE NCB(I)
      ISZ J            J=J+1
      ISZ KBUFO        SET KBUFO FOR IBUFO(J+1)
      JMP LOOP2        GO TO LOOP2
DONE2 ISZ I            I=I+1
      ISZ JCB          SET JCB FOR NCB(I+1)
      ISZ JSP          SET JSP FOR NSP(I+1)
      JMP LOOP1        GO TO LOOP1
DONE1 JMP UNPAK,I      RETURN
ERROR CLC 00B          CLEAR ALL CONTROL BITS
```

```
      JSB PMTTY      *NSP(I) GT 16 - PR184 RESTARTED*
      DEF MSG1                 MESSAGE 1 ADDRESS
      DEC 15                   MESSAGE 1 LENGTH = 30 CHARACTERS
      JMP RESTR     GO TO RESTR AND RESTART PR184
      SPC 2
      ORB
I     BSS 1
J     BSS 1
J0    BSS 1
KBUFO BSS 1
MASK1 OCT 000001    0/000/000/000/000/001
MSG1  ASC 15,NSP(I) GT 16 - PR184 RESTARTED
      END
```

```
ASMB,R,B               FILTR     PR18416
      HED FILTR - FILTER 32 BIT SAMPLE FOR NOISE
      NAM FILTR
      ENT FILTR
      SPC 2
*
***** FILTR - FILTER 32 BIT SAMPLE FOR NOISE
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
      ORB
ICB   DEF NCB
      ORR
      SPC 2
FILTR BSS 1          ENTRY/EXIT LINE
      LDA ICB        SET JCB FOR NCB(1)
      STA JCB
      CLA            SET I=0
      STA I
LOOP1 LDA I          IF I EQ NTAPE THEN GO TO DONE1
      CPA NTAPE
      JMP DONE1
      CLA            SET J=0
      STA J
LOOP2 LDA J          IF J=12 THEN GO TO DONE2
      CPA D12
      JMP DONE2
      LDA JCB,I      MASK UPPER 4 BITS
      AND MASK1
      CPA O0100      IF 4 BITS=0100 THEN GO TO C0000
      JMP C0000
      CPA O1011      IF 4 BITS=1011 THEN GO TO C1111
      JMP C1111
      CPA O1101      IF 4 BITS=1101 THEN GO TO C1111
      JMP C1111
      LDA JCB,I      LOAD NCB(I) FOR ROTATING AND STORING
      JMP LABL1      GO TO LABL1
C0000 LDA JCB,I      CHANGE UPPER 4 BITS OF NCB(I) TO 0000
      AND MASK2
      JMP LABL1      GO TO LABL1
C1111 LDA JCB,I      CHANGE UPPER 4 BITS OF NCB(I) TO 1111
      AND MASK2
      IOR MASK1
LABL1 RAL            ROTATE NCB(I) LEFT 1 BIT
      STA JCB,I      STORE NCB(I)
      ISZ J          J=J+1
      JMP LOOP2      GO TO LOOP2
DONE2 LDA JCB,I      ROTATE NCB(I) LEFT 4 BITS
      ALF
      STA JCB,I
      ISZ I          I=I+1
      ISZ JCB        SET JCB FOR NCB(I+1)
      JMP LOOP1      GO TO LOOP1
DONE1 JMP FILTR,I    RETURN
      SPC 2
      ORB
D12   DEC 12
I     BSS 1
J     BSS 1
MASK1 OCT 170000     1/111/000/000/000/000
MASK2 OCT 007777     0/000/111/111/111/111
O0100 OCT 040000     0/100/000/000/000/000
O1011 OCT 130000     1/011/000/000/000/000
O1101 OCT 150000     1/101/000/000/000/000
      END
```

```
ASMB,R,B            HUNT      PR18416
      HED HUNT - HUNT FOR 0/1 BIT OR EOR IN SAMPLE
      NAM HUNT
      ENT HUNT
      EXT DMAOT
      EXT PMTTY
      EXT PRUN
      EXT RESTR
      SPC 2
*
***** HUNT - HUNT FOR 0/1 BIT OR EOR IN SAMPLE
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
      ORB
IBIT  DEF NBIT
IBS   DEF NBS
ICB   DEF NCB
ISP   DEF NSP
IONES DEF NONES
ICNT  DEF NCNT
ISAVE DEF NSAVE
      ORR
      SPC 2
HUNT  BSS 1            ENTRY/EXIT LINE
      LDA IBIT         SET JBIT FOR NBIT(1,1)
      STA JBIT
      INA              SET KBIT FOR NBIT(2,1)
      STA KBIT
      LDA IBS          SET JBS FOR NBS(1)
      STA JBS
      LDA ICB          SET JCB FOR NCB(1)
      STA JCB
      LDA ISP          SET JSP FOR NSP(1)
      STA JSP
      LDA IONES        SET JONES FOR NONES(1)
      STA JONES
      LDA ICNT         SET JCNT FOR NCNT(1)
      STA JCNT
      LDA ISAVE        SET JSAVE FOR NSAVE(1)
      STA JSAVE
      LDA BBUFI        IF MBUFI=0 THEN SET QBUFI FOR OBUFI(1)
      LDB MBUFI
      SZB,RSS
      STA QBUFI
      CLA              SET I=0
      STA I
LOOP1 LDA I            IF I EQ NTAPE THEN GO TO DONE1
      CPA NTAPE
      JMP DONE1
      LDA D3           SET J=3 TO LEAVE 3 BITS UNPROCESSED
      STA J
LOOP2 LDA J            IF J EQ NSP(I) THEN GO TO DONE2
      CPA JSP,I
      JMP DONE2
      LDA JCB,I        GET NCB(I)
      AND MASK1        SET ITEST=UPPER 3 BTS
      STA ITEST
      AND MASK2        SET LBIT=UPPER 1 BIT
      STA LBIT
      LDA ITEST        IF ITEST=111 THEN GO TO LABL2
      CPA MASK1
      JMP LABL2
LABL1 LDA LBIT         SET NSAVE(I)=LBIT
      STA JSAVE,I
      RAL              PUT LBIT IN B00 POSITION FOR INTEGER
      ADA JONES,I      NONES(I)=NONES(I)+LBIT
      STA JONES,I
      ISZ JCNT,I       NCNT(I)=NCNT(I)+1
      ISZ J            J=J+1
      LDA JCB,I        ROTATE NCB(I) LEFT 1 BIT
      RAL
      STA JCB,I
      JMP LOOP2        GO TO LOOP2
LABL2 LDA JSAVE,I      IF NSAVE(I) NE 0 THEN GO TO LABL1
      SSA
      JMP LABL1
      LDA LBIT         SET NSAVE(I)=1
      STA JSAVE,I
      LDA JONES,I      IF NONES(I) LE NULL THEN GO TO ERROR
      CMA,INA
      ADA NULL
      SSA,RSS
      JMP ERROR
      LDA JCNT,I       IF NCNT(I) GT 32 THEN GO TO EOR
      CMA,INA
      ADA D32
      SSA
      JMP EOR
      LDB JCNT,I       SET B = 2*NONES(I)-NCNT(I)
      CMB,INB                = 2* 4-20 =  8-20 = -12 FOR 0 BIT
      ADB JONES,I            = 2*16-20 = 32-20 = +12 FOR 1 BIT
      ADB JONES,I
      CLE              SET E REG=0 FOR 0 BIT OF 32 BITS OF INFO
      SSB,RSS
      CCE              SET E REG=1 FOR 1 BIT OF 32 BITS OF INFO
      LDA JBIT,I       LOAD 32 BITS
      LDB KBIT,I
      ERA              ROTATE 32 BITS RIGHT 1 BIT AND OR IN E REG TO B15
      ERB
      STA JBIT,I       STORE 32 BITS
      STB KBIT,I
      ISZ JBS,I        NBS(I)=NBS(I)+1
LABL3 CLA              SET NONES(I)=0
      STA JONES,I
      STA JCNT,I       SET NCNT(I)=0
      JMP LOOP2        GO TO LOOP2
DONE2 LDA D3           SET NSP(I)=3
      STA JSP,I
      LDA JCB,I        ROTATE NCB(I) LEFT 3 BITS FOR UNPROCESSED 3 BITS
      ALF,RAR
      STA JCB,I
      ISZ I            I=I+1
      LDA JBIT         SET JBIT FOR NBIT(1,I+1)
      INA
      INA
      STA JBIT
      INA              SET KBIT FOR NBIT(2,I+1)
      STA KBIT
      ISZ JBS          SET JBS FOR NBS(I+1)
      ISZ JCB          SET JCB FOR NCB(I+1)
      ISZ JSP          SET JSP FOR NSP(I+1)
      ISZ JONES        SET JONES FOR NONES(I+1)
      ISZ JCNT         SET JCNT FOR NCNT(I+1)
      ISZ JSAVE        SET JSAVE FOR NSAVE(I+1)
      JMP LOOP1        GO TO LOOP1
DONE1 LDA NSAMP        SET NBUFO=NSAMP
      STA NBUFO
      JMP HUNT,I       RETURN
ERROR LDA JSP,I        SET ERROR FLAG - NBS(I) GT 32
      ADA D32
      INA
      STA JBS,I
      LDA LBIT         SET NSAVE(I)=1
      STA JSAVE,I
      JMP LABL3
EOR   LDA JBS,I        IF NBS(I)=32 THEN GO TO OUTB
```

```
      CPA D32
      JMP OUTB
      LIA 01B        GET SWITCH REGISTER
      RAR,RAR        POSITION SS02 IN B00
      SLA            IF SS02=1 THEN GO TO ERR32
      JMP ERR32
      JMP EOR0       GO TO EOR0
OUTB  LDA I          STORE TAPE NUMBER IN OUTPUT BUFFER
      INA
      STA QBUFI,I
      ISZ QBUFI      SET QBUFI FOR BBUFI(I+1)
      LDA JBIT,I     STORE UPPER 16 BITS IN OUTPUT BUFFER
      STA QBUFI,I
      ISZ QBUFI      SET QBUFI FOR BBUFI(I+1)
      LDA KBIT,I     STORE LOWER 16 BITS IN OUTPUT BUFFER
      STA QBUFI,I
      ISZ QBUFI      SET QBUFI FOR BBUFI(I+1)
      LDA MBUFI      MBUFI=MBUFI+3
      ADA D3
      STA MBUFI
      CPA D30        IF MBUFI NE 30 THEN GO TO EOR0
      JMP *+2
      JMP EOR0
      JSB DMAOT      WRITE OUTPUT BUFFER ONTO 3030
      LDA BBUFI      SET QBUFI FOR OBUFI(1)
      STA QBUFI
EOR0  CLA            SET NBIT(1,I)=0
      STA JBIT,I
      STA KBIT,I     SET NBIT(2,I)=0
      STA JBS,I      SET NBS(I)=0
      JMP LABL3
ERR32 CLC 00B        CLEAR ALL CONTROL BITS
      JSB PMTTY      =BAD 32 BITS DETECTED=
      DEF MSG1                 MESSAGE 1 ADDRESS
      DEC 10                   MESSAGE 1 LENGTH = 20 CHARACTERS
      JSB PMTTY      =SET SS15 TO [0,1] FOR [RESTART,IGNORE]=
      DEF MSG2                 MESSAGE 2 ADDRESS
      DEC 19                   MESSAGE 2 LENGTH = 38 CHARACTERS
      JSB PRUN       PRINT =PRESS =PRESET= AND =RUN= WHEN READY= ON TTY
      CLA            CLEAR A AND B REG
      CLB
      HLT 00B        HALT AND WAIT FOR PRESET AND RUN
      LIA 01B        GET SWITCH REGISTER
      SSA            IF SS15=1 THEN GO TO EOR0
      JMP EOR0
      JMP RESTR      GO TO RESTR AND RESTART PR184
      SPC 2
      ORB
D3    DEC 3
D30   DEC 30
I     BSS 1
ITEST BSS 1
J     BSS 1
JCNT  BSS 1
JONES BSS 1
JSAVE BSS 1
LBIT  BSS 1
MASK1 OCT 160000      1/110/000/000/000/000
MASK2 OCT 100000      1/000/000/000/000/000
MSG1  ASC 10,BAD 32 BITS DETECTED
MSG2  ASC 19,SET SS15 TO [0,1] FOR [RESTART,IGNORE]
QBUFI BSS 1
      END
```

```
ASMB,R,B              DMAOT     PR18416
      HED DMAOT - OUTPUT TO 9 TRACK MAGNETIC TAPE
      NAM DMAOT
      ENT DMAOT
      SPC 2
*
***** DMAOT - OUTPUT TO 9 TRACK MAGNETIC TAPE
*
      SPC 2
      COM NBIT(20),NBS(10),NCB(10),NSP(10),NTAPE
      COM NONES(10),NCNT(10),NSAVE(10)
      COM ABUFI,ABUFO,BBUFI,BBUFO
      COM NBUFI,NBUFO,MBUFI,MBUFO
      COM JBIT,KBIT,JBS,JCB,KCB,JSP,D32,D16,NSAMP,NULL
      SPC 2
DMAOT BSS 1          ENTRY/EXIT LINE
      LDA MBUFI      SWITCH BUFFER LENGTHS
      CLB                      SET B REG=0 FOR MBUFI
      STA MBUFO
      STB MBUFI
      LDA BBUFI      SWITCH BUFFER ADDRESSES
      LDB BBUFO
      STA BBUFO
      STB BBUFI
      SFS 07B        WAIT UNTIL LAST DMA FINISHED
      JMP *-1
      LDA PCW        GET DMA PROGRAM CONTROL WORD
      OTA 07B        SEND TO DMA CH 2
      CLC 03B        PREPARE DMA CH 2 MEMORY ADDRESS REGISTER
      LDA BBUFO      GET DMA ADDRESS WORD
      OTA 03B        SEND TO DMA CH 2
      STC 03B        PREPARE DMA CH 2 WORD COUNT RECORD
      LDA MBUFO      GET BUFFER LENGTH
      CMA,INA        SET NEGATIVE
      OTA 03B        SEND TO DMA CH 2
      SFS 16B        WAIT UNTIL 3030 NOT BUSY
      JMP *-1
      LDA CW116      SEND CONTROL WORD TO 3030 9 TRACK TAPE
      OTA 16B,C      INITIATE 9 TRACK MAGNETIC TAPE
      STC 07B,C      ACTIVATE DMA CH 2
      JMP DMAOT,I    RETURN
      SPC 2
      ORB
CW116 OCT 31         WRITE CHARACTERS
PCW   OCT 160015     STC,CLC,9 TRACK MAGNETIC TAPE
      END
```

```
ASMB,R,B                HPCDC      HPCDC
      HED HPCDC - HP 9 TRACK TO CDC 7 TRACK CONVERSION PROGRAM
      NAM HPCDC
      ENT START
      EXT PMTTY
      EXT PRUN
      EXT LOCAL
      EXT WRING
      EXT IOC
      EXT .DIO.
      EXT .IOI.
      EXT MTCCR
      EXT WRTID
      EXT WRTUT
*
***** HPCDC - HP 9 TRACK TO CDC 7 TRACK CONVERSION PROGRAM
*
***** READS 60 CHARACTER IDENT TO BE PROCESSED FROM TTY
*     POSITIONS THE 2020 TAPE FOR WRITING DATA
*     FINDS TTY IDENT ON 3030
*     WRITES 60 CHARACTER IDENT ONTO 2020 IN UT INTERNAL BCD
*     WRITES DATA ONTO 2020 IN 60 BIT UT BINARY WORDS
*
***** HLT CODES:
*     00  HPCDC     END OF HPCDC
*     01  HPCDC     WAITING FOR =PRESET= AND =RUN=
*     02  WRTID     TIMING ERROR
*     03  WRTUT     TIMING ERROR
*     04  WRTUT     BAD ICNT
*
***** SWITCH REGISTER CODES:
*     SS00 = 0 = NO LIST OF IDENTS READ FROM 3030
*          = 1 = LIST IDENTS READ FROM 3030
*     SS01 = 0 = NO REWIND 3030 BEFORE SEARCH FOR IDENT
*          = 1 = REWIND 3030 BEFORE SEARCH FOR IDENT
*     SS02 = 0 = DO NOT SKIP EXTRA MESSAGES
*          = 1 = SKIP EXTRA MESSAGES
*
***** TABS          7,11,21,31
*
      SPC 2
IDPRT ASC 02,ID [
IDENT BSS 30
      ASC 01,]
IBUF  BSS 1500
JDPRT DEF IDPRT
JDENT DEF IDENT
JBUF  DEF IBUF
      SPC 2
START HLT 00B       HALT - END OF PROGRAM
HPCDC NOP           ENTRY/EXIT LINE
      CLC 00B       CLEAR ALL CONTROL BITS
      CLF 00B       DISABLE INTERRUPT SYSTEM
      JSB PMTTY     WRITE BLANK LINE ON TTY
      DEF BLANK               BLANK LINE ADDRESS
      DEC 1                   LINE LENGTH - 2 CHARACTERS
      JSB PMTTY     =HP 9 TRACK TO CDC 7 TRACK CONVERSION PROGRAM=
      DEF MSG01               MESSAGE 1 ADDRESS
      DEC 22                  MESSAGE 1 LENGTH - 44 CHARACTERS
      LIA 01B       GET SWITCH REGISTERS
      RAR,RAR       POSITION SS02 IN B00
      SLA           IF SS02=1 THEN GO TO SKIP1
      JMP SKIP1
      JSB PMTTY     =9 TRACK TAPE IS 3030 AND 7 TRACK TAPE IS 2020=
      DEF MSG02               MESSAGE 2 ADDRESS
      DEC 23                  MESSAGE 2 LENGTH - 45 CHARACTERS
      JSB PMTTY     =SS00 = [0,1] FOR [NO LIST,LIST] IDENTS READ FROM 3030=
      DEF MSG03               MESSAGE 3 ADDRESS
      DEC 27                  MESSAGE 3 LENGTH - 53 CHARACTERS
      JSB PMTTY     =SS01 = [0,1] FOR [NO REWIND,REWIND] 3030 BEFORE SEARCH FOR
      DEF MSG04               MESSAGE 4 ADDRESS
      DEC 32                  MESSAGE 4 LENGTH - 64 CHARACTERS
SKIP1 JSB PRUN      PRINT =PRESS =PRESET= AND =RUN= WHEN READY= ON TTY
      HLT 01B       HALT AND WAIT FOR =PRESET= AND =RUN=
      JSB LOCAL     CHECK 2020 IN LOCAL
      DEF *+2                 RETURN ADDRESS
      DEF ICC20               2020 COMMAND CHANNEL
      JSB LOCAL     CHECK 3030 IN LOCAL
      DEF *+2                 RETURN ADDRESS
      DEF ICC30               3030 COMMAND CHANNEL
      JSB WRING     CHECK 2020 FOR WRITE RING
      DEF *+2                 RETURN ADDRESS
      DEF ICC20               2020 COMMAND CHANNEL
      JSB PMTTY     =ENTER 60 CHARACTER IDENT TO BE PROCESSED=
      DEF MSG05               MESSAGE 5 ADDRESS
      DEC 20                  MESSAGE 5 LENGTH - 40 CHARACTERS
      LDA JBUF      SET ADDR1 FOR IBUF(1)
      STA ADDR1
      LDA D30       SET ICNT=-30
      CMA,INA
      STA ICNT
      LDA BLANK     GET ASCII CODE FOR 2 8-BIT BLANK CHARACTERS
LOOP1 STA ADDR1,I   BLANK IDENT BUFFER
      ISZ ADDR1     SET ADDR1 FOR IBUF(I+1)
      ISZ ICNT      ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
      JMP LOOP1     GO TO LOOP1
      JSB IOC       READ 60 ASCII CHARACTERS AND ECHO PRINT FROM TTY
      DEF *+7                 RETURN ADDRESS
      DEF IOCKI               IOC KEYBOARD INPUT SELECT CODE
      DEF IREP                IOC READ AND ECHO PRINT REQUEST
      DEF IRSTA               STATUS AFTER READ
      DEF IRTRL               TRANSMISSION LOG AFTER READ
      DEF IBUF                BUFFER ADDRESS
      DEF D30                 BUFFER LENGTH
      JSB PMTTY     =ENTER [-1,0,+1] FOR [CONTINUE,START-OF-TAPE,END-OF-DATA]=
      DEF MSG06               MESSAGE 6 ADDRESS
      DEC 28                  MESSAGE 6 LENGTH - 56 CHARACTERS
      LIA 01B       GET SWITCH REGISTERS
      RAR,RAR       POSITION SS02 IN B00
      SLA           IF SS02=1 THEN GO TO SKIP2
      JMP SKIP2
      JSB PMTTY     =FOR POSITIONING 2020 FOR WRITING DATA=
      DEF MSG07               MESSAGE 7 ADDRESS
      DEC 21                  MESSAGE 7 LENGTH - 37 CHARACTERS
SKIP2 LDA INPUT     READ INTEGER FROM KEYBOARD INPUT
      CLB,INB
      JSB .DIO.
      ABS 0                   FREE FIELD
      DEF *+3                 RETURN ADDRESS
      JSB .IOI.               STORE INTEGER AS IPOS
      STA IPOS
      SSA           IF IPOS LT 0 THEN GO TO LAB03
      JMP LAB03
      JSB MTCCR     REWIND 2020
      DEF *+3                 RETURN ADDRESS
      DEF ICC20               2020 COMMAND CHANNEL
      DEF IREW2               2020 REWIND AND READY REQUEST
      LDA IPOS      IF IPOS=0 THEN GO TO LAB04
      SZA,RSS
      JMP LAB04
LAB01 LDA DM2       SET IEOF=-2
      STA IEOF
LAB02 JSB MTCCR     FORWARD SPACE 2020 ONE RECORD
      DEF *+4                 RETURN ADDRESS
      DEF ICC20               2020 COMMAND CHANNEL
      DEF IFSR2               2020 FORWARD SPACE ONE RECORD REQUEST
      DEF IRSTA               2020 STATUS AFTER READ
      AND MEOT      MASK OUT EOT BIT FROM 2020 STATUS WORD
      SZA           IF EOT THEN GO TO ER901
      JMP ER901
      LDA IRSTA     GET STATUS OF 2020
      AND MEOF      MASK OUT EOF BIT
```

```
      SZA,RSS        IF NOT EOF THEN GO TO LAB01
      JMP LAB01
      ISZ IEOF       IEOF=IEOF+1 THEN IF IEOF=0 SKIP NEXT INSTRUCTION
      JMP LAB02      GO TO LAB02
LAB03 JSB MTCCR      BACK SPACE 2020 ONE RECORD
      DEF *+4                  RETURN ADDRESS
      DEF ICC20                2020 COMMAND CHANNEL
      DEF IBSR2                2020 BACK SPACE ONE RECORD REQUEST
      DEF IRSTA                2020 STATUS AFTER BACK SPACE
LAB04 CLA            SET ISOT=0
      STA ISOT
      LIA 01B        GET SWITCH REGISTER
      RAR            POSITION SS01 IN B00
      SLA            IF SS01=1 THEN GO TO LAB05
      JMP LAB05
      JSB IOC        GET STATUS OF 3030
      DEF *+5                  RETURN ADDRESS
      DEF IOC30                IOC 3030 SELECT CODE
      DEF ISTA                 IOC STATUS REQUEST
      DEF IRSTA                3030 STATUS ADDRESS
      DEF IRTRL                3030 TRANSMISSION LOG ADDRESS
      AND MSOT       MASK OUT SOT BIT FROM 3030 STATUS WORD
      SZA            IF SOT THEN GO TO LAB06 ELSE GO TO LAB07
      JMP LAB06
      JMP LAB07
LAB05 JSB IOC        REWIND 3030
      DEF *+3                  RETURN ADDRESS
      DEF IOC30                IOC 3030 SELECT CODE
      DEF IREW                 IOC REWIND TO LOAD POINT AND READY REQUEST
LAB06 CCA            SET ISOT=-1
      STA ISOT
LAB07 JSB LOCAL      CHECK 3030 IN LOCAL
      DEF *+2                  RETURN ADDRESS
      DEF ICC30                3030 COMMAND CHANNEL
      JSB IOC        READ 60 CHARACTER IDENT FROM 3030
      DEF *+7                  RETURN ADDRESS
      DEF IOC30                IOC 3030 SELECT CODE
      DEF IRED                 IOC READ ONE RECORD REQUEST
      DEF IRSTA                3030 STATUS AFTER READ
      DEF IRTRL                3030 TRANSMISSION LOG AFTER READ
      DEF IDENT                BUFFER ADDRESS
      DEF D30                  BUFFER LENGTH
      AND MEOT       MASK OUT EOT BIT FROM 3030 STATUS WORD
      SZA            IF EOT THEN GO TO ER902
      JMP ER902
      LDA IRSTA      GET STATUS OF 3030
      AND MERR       MASK OUT ERROR BITS
      SZA,RSS        IF NO READ ERROR THEN GO TO LAB08
      JMP LAB08
      JSB PMTTY      "READ ERROR ON 3030 - SS15 = (0,1) FOR (RE-READ,SKIP) RECORD
      DEF MSG08                MESSAGE 8 ADDRESS
      DEC 30                   MESSAGE 8 LENGTH - 59 CHARACTERS
      JSB PRUN       "PRINT "PRESS "PRESET" AND "RUN" WHEN READY" ON TTY
      HLT 00B        HALT AND WAIT FOR "PRESET" AND "RUN"
      LIA 01B        GET SWITCH REGISTER
      SSA            IF SS15=1 THEN GO TO LAB07 AND SKIP RECORD
      JMP LAB07
      JSB MTCCR      BACK SPACE 3030 ONE RECORD
      DEF *+4                  RETURN ADDRESS
      DEF ICC30                3030 COMMAND CHANNEL
      DEF IBSR3                3030 BACK SPACE ONE RECORD REQUEST
      DEF IRSTA                3030 STATUS AFTER BACK SPACE
      JMP LAB07      GO TO LAB07 AND RE-READ RECORD
LAB08 LDA IRSTA      GET STATUS OF 3030
      AND MEOF       MASK OUT EOF BIT
      SZA            IF EOF THEN GO TO ER904
      JMP ER904
      CLA            SET ISOT=0
      STA ISOT
      LIA 01B        GET SWITCH REGISTER
      SLA,RSS        IF SS00=0 THEN GO TO LAB09
      JMP LAB09
      JSB PMTTY      WRITE 3030 60 CHARACTER IDENT ON TTY
      DEF IDPRT                BUFFER ADDRESS
      DEC 33                   BUFFER LENGTH - 65 CHARACTERS
LAB09 LDA D30        SET ICNT=-30
      CMA,INA
      STA ICNT
      LDA JDENT      SET ADDR1 FOR IDENT(1)
      STA ADDR1
      LDA JBUF       SET ADDR2 FOR IBUF(1)
      STA ADDR2
LOOP2 LDA ADDR1,I    GET IDENT(I)
      CMA,INA        SET NEGATIVE
      ADA ADDR2,I    ADD IBUF(I)
      SZA            IF IDENT(I) NE IBUF(I) THEN GO TO LAB10
      JMP LAB10
      ISZ ADDR1      SET ADDR1 FOR IDENT(I+1)
      ISZ ADDR2      SET ADDR2 FOR IBUF(I+1)
      ISZ ICNT       ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
      JMP LOOP2      GO TO LOOP2 AND CHECK NEXT CHARACTERS
      JMP LAB11      GO TO LAB11 - IDENTS MATCH
LAB10 JSB MTCCR      FORWARD SPACE 3030 ONE RECORD
      DEF *+4                  RETURN ADDRESS
      DEF ICC30                3030 COMMAND CHANNEL
      DEF IFSR3                3030 FORWARD SPACE ONE RECORD REQUEST
      DEF IRSTA                3030 STATUS AFTER FORWARD SPACE
      AND MEOT       MASK OUT EOT BIT FROM 3030 STATUS WORD
      SZA            IF EOT THEN GO TO ER902
      JMP ER902
      LDA IRSTA      GET STATUS OF 3030
      AND MEOF       MASK OUT EOF BIT
      SZA            IF EOF THEN GO TO LAB07 AND READ NEXT IDENT ELSE GO TO LAB10
      JMP LAB07
      JMP LAB10
LAB11 CLA,INA        SET IREC=1
      STA IREC
LAB12 JSB WRTID      WRITE 60 CHARACTER IDENT ON 2020 IN UT INTERNAL BCD
      DEF *+4                  RETURN ADDRESS
      DEF IDENT                BUFFER ADDRESS
      DEF D30                  BUFFER LENGTH
      DEF IWSTA                2020 STATUS AFTER WRITE
      AND MEOT       MASK EOT BIT FROM 2020 STATUS WORD
      SZA            IF EOT THEN GO TO ER906
      JMP ER906
      LDA IWSTA      GET STATUS OF 2020
      AND MERR       MASK OUT ERROR BITS
      SZA,RSS        IF NO WRITE ERRORS THEN GO TO LAB13
      JMP LAB13
      JSB BRGAP      BACK SPACE 2020 AND WRITE 3 INCH GAP
      AND MEOT       MASK EOT BIT FROM 2020 STATUS WORD
      SZA            IF EOT THEN GO TO ER906 ELSE LAB12
      JMP ER906
      JMP LAB12
LAB13 LDA JBUF       SET ADDR1 FOR IBUF(1)
      STA ADDR1
      ISZ IREC       IREC=IREC+1
LAB14 JSB LOCAL      CHECK 3030 IN LOCAL
      DEF *+2                  RETURN ADDRESS
      DEF ICC30                3030 COMMAND CHANNEL
      JSB IOC        READ 30 WORDS FROM 3030 INTO IBUF(I)
      DEF *+7                  RETURN ADDRESS
      DEF IOC30                3030 SELECT CODE
      DEF IRED                 IOC READ ONE RECORD REQUEST
      DEF IRSTA                3030 STATUS AFTER READ
      DEF IRTRL                3030 TRANSMISSION LOG AFTER READ
      DEF ADDR1,I              BUFFER ADDRESS
      DEF D30                  BUFFER LENGTH
      AND MEOT       MASK EOT BIT FROM 3030 STATUS WORD
      SZA            IF EOT THEN GO TO ER911
      JMP ER911
      LDA IRSTA      GET STATUS OF 3030
```

```
      AND MERR     MASK OUT ERROR BITS
      SZA,RSS      IF NO READ ERRORS THEN GO TO LAB15
      JMP LAB15
      JSB PMTTY    ≡READ ERROR ON 3030 - SS15 = [0,1] FOR [RE-READ,SKIP] RECORD
      DEF MSG08              MESSAGE 8 ADDRESS
      DEC 30                 MESSAGE 8 LENGTH - 59 CHARACTERS
      JSB PRUN     PRINT ≡PRESS ≡PRESET≡ AND ≡RUN≡ WHEN READY≡ ON TTY
      HLT 00B      HALT AND WAIT FOR ≡PRESET≡ AND ≡RUN≡
      LIA 01B      GET SWITCH REGISTER
      SSA          IF SS15=1 THEN GO TO LAB14 AND SKIP RECORD
      JMP LAB14
      JSB MTCCR    BACK SPACE 3030 ONE RECORD
      DEF *+4                RETURN ADDRESS
      DEF ICC30              3030 COMMAND CHANNEL
      DEF IBSR3              3030 BACK SPACE ONE RECORD REQUEST
      DEF IRSTA              3030 STATUS AFTER BACK SPACE
      JMP LAB14    GO TO LAB14 AND RE-READ RECORD
LAB15 LDA IRSTA    GET STATUS OF 3030
      AND MEOF     MASK OUT EOF BIT
      STA IEOF     STORE EOF BIT FOR LATER USE
      SZA          IF EOF THEN GO TO LAB16
      JMP LAB16
      LDA ADDR1    SET ADDR1 FOR IBUF(I+30)
      ADA D30
      STA ADDR1
      LDA JBUF     IF ADDR1 LT IBUF(1501) THEN GO TO LAB14
      ADA D1500
      CMA,INA
      ADA ADDR1
      SSA
      JMP LAB14
LAB16 LDA ADDR1    IF ADDR1 EQ IBUF(1) THEN GO TO LAB19
      CPA JBUF
      JMP LAB19
LAB17 LDA JBUF     SET ICNT=ADDR1-JBUF
      CMA,INA
      ADA ADDR1
      STA ICNT
      JSB WRTUT    WRITE DATA ONTO 2020 IN 60 BIT UT BINARY
      DEF *+4                RETURN ADDRESS
      DEF IBUF               BUFFER ADDRESS
      DEF ICNT               BUFFER LENGTH
      DEF IWSTA              2020 STATUS AFTER WRITE
      AND MEOT     MASK OUT EOT BIT FROM 2020 STATUS WORD
      SZA          IF EOT THEN GO TO ER906
      JMP ER906
      LDA IWSTA    GET STATUS OF 2020
      AND MERR     MASK OUT ERROR BITS
      SZA,RSS      IF NO WRITE ERROR THEN GO TO LAB18
      JMP LAB18
      JSB BRGAP    BACK SPACE 2020 ONE RECORD AND WRITE 3 INCH GAP
      AND MEOT     MASK OUT EOT BIT
      SZA          IF EOT THEN GO TO ER906 ELSE LAB17
      JMP ER906
      JMP LAB17
LAB18 LDA IEOF     GET EOF BIT FROM LAST 3030 READ
      SZA,RSS      IF NOT EOF THEN GO TO LAB13
      JMP LAB13
LAB19 JSB PMTTY    ≡UT CDC COMPATABLE TAPE COMPLETED≡
      DEF MSG09              MESSAGE 9 ADDRESS
      DEC 16                 MESSAGE 9 LENGTH - 32 CHARACTERS
LAB20 JSB IOC      CLEAR 2020
      DEF *+3                RETURN ADDRESS
      DEF IOC20              IOC 2020 SELECT CODE
      DEF ICLR               IOC CLEAR REQUEST
      JSB IOC      WRITE EOF ON 2020
      DEF *+3                RETURN ADDRESS
      DEF IOC20              IOC 2020 SELECT CODE
      DEF IWFM               IOC WRITE FILE MARK REQUEST
      JSB IOC      WRITE EOF ON 2020
      DEF *+3                RETURN ADDRESS
```

```
      DEF IOC20              IOC 2020 SELECT CODE
      DEF IWFM               IOC WRITE FILE MARK REQUEST
LAB21 JSB PMTTY    ≡END OF PROGRAM - RUN AGAIN≡
      DEF MSG10              MESSAGE 10 ADDRESS
      DEC 13                 MESSAGE 10 LENGTH - 26 CHARACTERS
      JSB PRUN     PRINT ≡PRESS ≡PRESET≡ AND ≡RUN≡ WHEN READY≡ ON TTY
      JMP START    GO TO START AND START AGAIN
ER901 JSB PMTTY    ≡END-OF-TAPE ON 2020≡
      DEF EM01               ERROR MESSAGE 1 ADDRESS
      DEC 10                 ERROR MESSAGE 1 LENGTH - 19 CHARACTERS
      JSB PMTTY    ≡WHILE POSITIONING FOR WRITE≡
      DEF EM02               ERROR MESSAGE 2 ADDRESS
      DEC 14                 ERROR MESSAGE 2 LENGTH - 27 CHARACTERS
      JSB PMTTY    ≡MOUNT ANOTHER TAPE≡
      DEF EM03               ERROR MESSAGE 3 ADDRESS
      DEC 9                  ERROR MESSAGE 3 LENGTH - 18 CHARACTERS
      JMP LAB21    GO TO LAB21 AND END PROGRAM
ER902 JSB PMTTY    ≡END-OF-TAPE ON 3030≡
      DEF EM04               ERROR MESSAGE 4 ADDRESS
      DEC 10                 ERROR MESSAGE 4 LENGTH - 19 CHARACTERS
ER903 JSB PMTTY    ≡WHILE SEARCHING FOR IDENT≡
      DEF EM05               ERROR MESSAGE 5 ADDRESS
      DEC 13                 ERROR MESSAGE 5 LENGTH - 25 CHARACTERS
      JSB PMTTY    ≡POSSIBLY NO DATA ON TAPE≡
      DEF EM06               ERROR MESSAGE 6 ADDRESS
      DEC 12                 ERROR MESSAGE 6 LENGTH - 24 CHARACTERS
      JMP LAB21    GO TO LAB21 AND END PROGRAM
ER904 LDA ISOT     IF ISOT GE 0 THEN GO TO ER905
      SSA,RSS
      JMP ER905
      JSB PMTTY    ≡END-OF-FILE ON 3030 AT START-OF-TAPE≡
      DEF EM07               ERROR MESSAGE 7 ADDRESS
      DEC 18                 ERROR MESSAGE 7 LENGTH - 36 CHARACTERS
      JMP ER903    GO TO ER903 AND CONTINUE
ER905 JSB PMTTY    ≡END-OF-DATA ON 3030≡
      DEF EM08               ERROR MESSAGE 8 ADDRESS
      DEC 10                 ERROR MESSAGE 8 LENGTH - 19 CHARACTERS
      JSB PMTTY    ≡POSSIBLY NO DATA ON TAPE≡
      DEF EM06               ERROR MESSAGE 6 ADDRESS
      DEC 12                 ERROR MESSAGE 6 LENGTH - 24 CHARACTERS
      JSB PMTTY    ≡OR INCORRECT IDENT SPECIFIED≡
      DEF EM09               EZROR MESSAGE 9 ADDRESS
      DEC 14                 ERROR MESSAGE 9 LENGTH - 28 CHARACTERS
      JMP LAB21    GO TO LAB21 AND END PROGRAM
ER906 JSB PMTTY    ≡END-OF-TAPE ON 2020≡
      DEF EM01               ERROR MESSAGE 1 ADDRESS
      DEC 10                 ERROR MESSAGE 1 LENGTH - 19 CHARACTERS
      JSB PMTTY    ≡WHILE WRITING DATA≡
      DEF EM10               ERROR MESSAGE 10 ADDRESS
      DEC 9                  ERROR MESSAGE 10 LENGTH - 18 CHARACTERS
ER907 LDA IREC     SET IREC=-IREC
      CMA,INA
      STA IREC
ER908 JSB MTCCR    BACK SPACE 2020 ONE RECORD
      DEF *+4                RETURN ADDRESS
      DEF ICC20              2020 COMMAND CHANNEL
      DEF IBSR2              2020 BACK SPACE ONE RECORD REQUEST
      DEF IRSTA              2020 STATUS AFTER BACK SPACE
      AND MSOT     MASK OUT SOT BIT FROM 2020 STATUS WORD
      SZA          IF START-OF-TAPE THEN GO TO ER910
      JMP ER910
      LDA IRSTA    GET STATUS OF 2020
      AND MEOF     MASK OUT END-OF-FILE BIT
      SZA          IF END-OF-FILE THEN GO TO ER909
      JMP ER909
      ISZ ICNT     ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
      JMP ER908    GO TO ER908 AND BACK SPACE 2020 ONE RECORD
ER909 JSB PMTTY    ≡MOUNT ANOTHER TAPE≡
      DEF EM03               ERROR MESSAGE 3 ADDRESS
      DEC 9                  ERROR MESSAGE 3 LENGTH - 18 CHARACTERS
      JMP LAB20    GO TO LAB20, WRITE EOD, AND END PROGRAM
```

```
ER910 JSB PMTTY      =START=OF=TAPE ON 2020=
      DEF EM11                 ERROR MESSAGE 11 ADDRESS
      DEC 11                   ERROR MESSAGE 11 LENGTH = 21 CHARACTERS
      JMP LAB21      GO TO LAB21 AND END PROGRAM
ER911 JSB PMTTY      =END=OF=TAPE ON 3030=
      DEF EM04                 ERROR MESSAGE 4 ADDRESS
      DEC 10                   ERROR MESSAGE 4 LENGTH = 19 CHARACTERS
      JSB PMTTY      =WHILE READING DATA=
      DEF EM12                 ERROR MESSAGE 12 ADDRESS
      DEC 9                    ERROR MESSAGE 12 LENGTH = 18 CHARACTERS
      JMP ER907      GO TO ER907, BACK SPACE IREC RECORDS, WRITE EOD, AND END PRO
      SPC 2
BRGAP BSS 1          ENTRY/EXIT LINE
      JSB MTCCR      BACK SPACE 2020 ONE RECORD
      DEF *+3                  RETURN ADDRESS
      DEF ICC20                2020 COMMAND CHANNEL
      DEF IBSR2                2020 BACK SPACE ONE RECORD REQUEST
      JSB MTCCR      WRITE 3 INCH GAP ON 2020
      DEF *+4                  RETURN ADDRESS
      DEF ICC20                2020 COMMAND CHANNEL
      DEF IGAP2                2020 WRITE 3 INCH GAP REQUEST
      DEF IWSTA                2020 STATUS AFTER WRITE
      JMP BRGAP,I    RETURN
      SPC 2
*
*     MESSAGES:
*
MSG01 ASC 22,HP 9 TRACK TO CDC 7 TRACK CONVERSION PROGRAM
MSG02 ASC 23,9 TRACK TAPE IS 3030 AND 7 TRACK TAPE IS 2020
MSG03 ASC 27,SS00 = [0,1] FOR [NO LIST,LIST] IDENTS READ FROM 3030
MSG04 ASC 28,SS01 = [0,1] FOR [NO REWIND,REWIND] 3030 BEFORE SEARCH F
      ASC 04,OR IDENT
MSG05 ASC 20,ENTER 60 CHARACTER IDENT TO BE PROCESSED
MSG06 ASC 28,ENTER [-1,0,+1] FOR [CONTINUE,START=OF=TAPE,END=OF=DATA]
MSG07 ASC 21,FOR POSITIONING 2020 FOR WRITING DATA
MSG08 ASC 28,READ ERROR ON 3030 - SS15 = [0,1] FOR [RE=READ,SKIP] REC
      ASC 02,ORD
MSG09 ASC 16,UT CDC COMPATABLE TAPE COMPLETED
MSG10 ASC 13,END OF PROGRAM - RUN AGAIN
      SPC 2
      ORB
*
*     ERROR MESSAGES:
*
EM01  ASC 10,END=OF=TAPE ON 2020
EM02  ASC 14,WHILE POSITIONING FOR WRITE
EM03  ASC 09,MOUNT ANOTHER TAPE
EM04  ASC 10,END=OF=TAPE ON 3030
EM05  ASC 13,WHILE SEARCHING FOR IDENT
EM06  ASC 12,POSSIBLY NO DATA ON TAPE
EM07  ASC 18,END=OF=FILE ON 3030 AT START=OF=TAPE
EM08  ASC 10,END=OF=DATA ON 3030
EM09  ASC 14,OR INCORRECT IDENT SPECIFIED
EM10  ASC 09,WHILE WRITING DATA
EM11  ASC 11,START=OF=TAPE ON 2020
EM12  ASC 09,WHILE READING DATA
      SPC 2
ADDR1 BSS 1
ADDR2 BSS 1
BLANK OCT 020040
D30   DEC 30
D1500 DEC 1500
DM2   DEC -2
IBSR2 OCT 000101
IBSR3 OCT 000041
ICC20 OCT 000011
ICC30 OCT 000016
ICLR  OCT 000000
ICNT  BSS 1
IEOF  BSS 1
IFSR2 OCT 000003
IFSR3 OCT 000003
IGAP2 OCT 000015
INPUT OCT 000001
IOC20 OCT 000007
IOC30 OCT 000012
IOCKI OCT 000001
IPOS  BSS 1
IREC  BSS 1
IRED  OCT 010100
IREP  OCT 010400
IREW  OCT 030400
IREW2 OCT 000201
IRSTA BSS 1
IRTRL BSS 1
ISOT  BSS 1
ISTA  OCT 040000
IWFM  OCT 030100
IWSTA BSS 1
MEOF  OCT 000200     0/000/000/010/000/000
MEOT  OCT 000040     0/000/000/000/100/000
MERR  OCT 000032     0/000/000/000/011/010
MSOT  OCT 000100     0/000/000/001/000/000
      END HPCDC
```

```
ASMB,R,B             WRTID     HPCDC
      HED WRTID  -  WRITE 60 CHARACTER IDENT ON 2020 IN UT BCD
      NAM WRTID
      ENT WRTID
      EXT GETAP
      EXT LOCAL
      EXT WRING
      EXT ASCUT
      SPC 2
*
***** WRTID - WRITE 60 CHARACTER IDENT ON 2020 IN UT BCD
*
***** FORMAL PARAMETERS:
*     IBUF           BUFFER ADDRESS
*     ICNT           BUFFER LENGTH ADDRESS
*     ISTA           2020 STATUS AFTER WRITE ADDRESS
*
***** HLT CODES:
*     02  WRTID      TIMING ERROR
*
      SPC 2
      ORB
IBUF  BSS 1          BUFFER ADDRESS
ICNT  BSS 1          BUFFER LENGTH ADDRESS
ISTA  BSS 1          2020 STATUS AFTER WRITE ADDRESS
      SPC 2
WRTID BSS 1          ENTRY/EXIT LINE
      JSB GETAP      GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IBUF                 FWA OF FORMAL PARAMETERS
      LDA IBUF       SET JBUF FOR IBUF(1)
      STA JBUF
      LDA ICNT,I     SET JCNT=-ICNT
      CMA,INA
      STA JCNT
LOOP1 LDA JBUF,I     GET NEXT 2 ASCII CHARACTERS
      ALF,ALF        POSITION FOR UPPER ASCII CHARACTER
      JSB ASCUT      CONVERT FROM ASCII TO UT BCD CHARACTER CODE
      ALF,ALF        POSITION FOR UPPER UT BCD CHARACTER
      STA TEMP       STORE FOR LATER USE
      LDA JBUF,I     GET 2 ASCII CHARACTERS AGAIN
      JSB ASCUT      CONVERT FROM ASCII TO UT BCD CHARACTER CODE
      IOR TEMP       OR IN UPPER UT BCD CHARACTER
      STA JBUF,I     SAVE 2 UT BCD CHARACTERS
      ISZ JBUF       SET JBUF FOR IBUF(I+1)
      ISZ JCNT       JCNT=JCNT+1 THEN IF JCNT=0 SKIP NEXT INSTRUCTION
      JMP LOOP1      GO TO LOOP1 AND CONVERT NEXT 2 ASCII CHARACTERS
      LDA IBUF       SET JBUF FOR IBUF(1)
      STA JBUF
      LDA ICNT,I     SET JCNT=-ICNT
      CMA,INA
      STA JCNT
      JSB LOCAL      CHECK FOR 2020 IN LOCAL
      DEF *+2                  RETURN ADDRESS
      DEF C2020                2020 COMMAND CHANNEL
      JSB WRING      CHECK FOR 2020 WRITE WRING
      DEF *+2                  RETURN ADDRESS
      DEF C2020                2020 COMMAND CHANNEL
      SFS 11B        WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      LDA WRO        GET WRITE RECORD ODD PARITY REQUEST
      OTA 11B        SEND TO 2020 COMMAND CHANNEL
      LDA JBUF,I     GET FIRST 2 UT BCD CHARACTERS
      ALF,ALF        POSITION FOR UPPER CHARACTER
      JMP LABL1      GO TO LABL1 AND SEND TO 2020 DATA CHANNEL - 10.0 MIC SEC
LOOP2 LDA JBUF,I     GET NEXT 2 UT BCD CHARACTERS
      ALF,ALF        POSITION FOR UPPER CHARACTER
      SFC 10B        IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 02B
LABL1 SFS 10B        WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C      SEND TO 2020 DATA CHANNEL - FRAME 1 - 2 TO 1 = 18.5 MIC SEC
      LDA JBUF,I     GET 2 UT BCD CHARACTERS AGAIN
      ISZ JBUF       SET JBUF FOR JBUF(I+1)
      SFC 10B        IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 02B
      SFS 10B        WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C      SEND TO 2020 DATA CHANNEL - FRAME 2 - 1 TO 2 = 14.5 MIC SEC
      ISZ JCNT       JCNT=JCNT+1 THEN IF JCNT=0 SKIP NEXT INSTRUCTION
      JMP LOOP2      GO TO LOOP2 AND PROCESS NEXT CHARACTER
      CLC 10B        CLEAR CONTROL BIT FOR 2020 DATA CHANNEL
      LDA CLR        GET CLEAR REQUEST
      OTA 11B        SEND TO 2020 COMMAND CHANNEL
      SFS 11B        WAIT UNTIL 2020 CONTROLLER NOT BUSY
      JMP *-1
      LIA 11B        GET STATUS OF 2020 FROM COMMAND CHANNEL
      STA ISTA,I     STORE STATUS FOR RETURN
      JMP WRTID,I    RETURN
      SPC 2
C2020 OCT 000011
CLR   OCT 000000
JBUF  BSS 1
JCNT  BSS 1
TEMP  BSS 1
WRO   OCT 000071
      END
```

```
ASMB,R,B              ASCUT     HPCDC
      HED ASCUT - CONVERT FROM ASCII TO UT BCD CHARACTER CODE
      NAM ASCUT
      ENT ASCUT
      SPC 2
*
***** ASCUT - CONVERT FROM ASCII TO UT BCD CHARACTER CODE
*
***** ENTRY:
*      A    ASCII CHARACTER CODE IN BØØ-BØ6
*      B    TRASH
*
***** EXIT:
*      A    UT BCD CHARACTER CODE IN BØØ-BØ5
*      B    TRASH
*
      SPC 2
      ORB
A     EQU Ø
B     EQU 1
UTBCD DEF BCD
      SPC 2
ASCUT BSS 1          ENTRY/EXIT LINE
      AND MASK1      MASK OUT ASCII CHARACTER FROM BØØ-BØ6
      ADA DM32       SET FIRST 32 ASCII CODES BLANK
      SSA
      CLA
      LDB A          SET LAST 32 ASCII CODES BLANK
      CMB,INB
      ADB D64
      SSB
      CLA
      ADA UTBCD      ADD FWA OF BCD ARRAY
      LDA A,I        GET UT BCD CHARACTER CODE
      JMP ASCUT,I    RETURN
      SPC 2
BCD   OCT ØØØØ55     ■ ■
      OCT ØØØØ71     ■↓■        EXCLAMATION MARK
      OCT ØØØØ6Ø     ■■■        DOUBLE QUOTE MARK
      OCT ØØØØ64     ■≠■        NUMBER OR POUND SYMBOL
      OCT ØØØØ53     ■$■
      OCT ØØØØ66     ■v■        PERCENT SYMBOL
      OCT ØØØØ67     ■∧■        AND SYMBOL
      OCT ØØØØ65     ■→■        SINGLE QUOTE MARK
      OCT ØØØØ51     ■(■
      OCT ØØØØ52     ■)■
      OCT ØØØØ47     ■*■
      OCT ØØØØ45     ■+■
      OCT ØØØØ56     ■,■
      OCT ØØØØ46     ■-■
      OCT ØØØØ57     ■.■
      OCT ØØØØ5Ø     ■/■
      OCT ØØØØ33     ■Ø■
      OCT ØØØØ34     ■1■
      OCT ØØØØ35     ■2■
      OCT ØØØØ36     ■3■
      OCT ØØØØ37     ■4■
      OCT ØØØØ4Ø     ■5■
      OCT ØØØØ41     ■6■
      OCT ØØØØ42     ■7■
      OCT ØØØØ43     ■8■
      OCT ØØØØ44     ■9■
      OCT ØØØØ63     ■:■
      OCT ØØØØ77     ■;■
      OCT ØØØØ72     ■<■
      OCT ØØØØ54     ■=■
      OCT ØØØØ73     ■>■
      OCT ØØØØ75     ■≥■        QUESTION MARK
      OCT ØØØØ74     ■≤■        AT SYMBOL
      OCT ØØØØØ1     ■A■
      OCT ØØØØØ2     ■B■
      OCT ØØØØØ3     ■C■
      OCT ØØØØØ4     ■D■
      OCT ØØØØØ5     ■E■
      OCT ØØØØØ6     ■F■
      OCT ØØØØØ7     ■G■
      OCT ØØØØ1Ø     ■H■
      OCT ØØØØ11     ■I■
      OCT ØØØØ12     ■J■
      OCT ØØØØ13     ■K■
      OCT ØØØØ14     ■L■
      OCT ØØØØ15     ■M■
      OCT ØØØØ16     ■N■
      OCT ØØØØ17     ■O■
      OCT ØØØØ2Ø     ■P■
      OCT ØØØØ21     ■Q■
      OCT ØØØØ22     ■R■
      OCT ØØØØ23     ■S■
      OCT ØØØØ24     ■T■
      OCT ØØØØ25     ■U■
      OCT ØØØØ26     ■V■
      OCT ØØØØ27     ■W■
      OCT ØØØØ3Ø     ■X■
      OCT ØØØØ31     ■Y■
      OCT ØØØØ32     ■Z■
      OCT ØØØØ61     ■[■
      OCT ØØØØ76     ■¬■        REVERSE SLASH MARK
      OCT ØØØØ62     ■]■
      OCT ØØØØ7Ø     ■↑■
      OCT ØØØØ55     ■ ■        LEFT ARROW SYMBOL
DM32  DEC -32
D64   DEC 64
MASK1 OCT ØØØ177     Ø/ØØØ/ØØØ/ØØ1/111/111
      END
```

```
ASMB,R,B               WRTUT     HPCDC
      HED WRTUT - WRITE DATA ON 2020 IN UT BINARY
      NAM WRTUT
      ENT WRTUT
      EXT GETAP
      EXT LOCAL
      EXT WRING
      SPC 2
*
***** WRTUT - WRITE DATA ON 2020 IN UT BINARY
*
***** TAPE FRAMES:
*     FRAME 01        6/ZEROES
*     FRAME 02        6/ZEROES
*     FRAME 03        6/B06-B11 OF INPUT TAPE NUMBER
*     FRAME 04        6/B00-B05 OF INPUT TAPE NUMBER
*     FRAME 05        4/ZEROES,2/B30-B31 OF 32 BITS OF INFORMATION
*     FRAME 06        6/B24-B29 OF 32 BITS OF INFORMATION
*     FRAME 07        6/B18-B23 OF 32 BITS OF INFORMATION
*     FRAME 08        6/B12-B17 OF 32 BITS OF INFORMATION
*     FRAME 09        6/B06-B11 OF 32 BITS OF INFORMATION
*     FRAME 10        6/B00-B05 OF 32 BITS OF INFORMATION
*
***** CONTENTS OF 60 BIT UT BINARY WORD:
*     B00-B31         32 BITS OF INFORMATION
*     B32-B35          4 BITS OF ZEROES
*     B36-B47         12 BITS OF INPUT TAPE NUMBER
*     B48-B59         12 BITS OF ZEROES
*
***** FORMAL PARAMETERS:
*     IBUF            BUFFER ADDRESS
*     ICNT            BUFFER LENGTH ADDRESS
*     ISTA            2020 STATUS AFTER WRITE ADDRESS
*
***** HLT CODES:
*     03  WRTUT       TIMING ERROR
*     04  WRTUT       BAD ICNT
*
      SPC 2
      ORB
A     EQU 0
B     EQU 1
IBUF  BSS 1           BUFFER ADDRESS
ICNT  BSS 1           BUFFER LENGTH ADDRESS
ISTA  BSS 1           2020 STATUS WORD AFTER WRITE ADDRESS
      SPC 2
WRTUT BSS 1           ENTRY/EXIT LINE
      JSB GETAP       GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IBUF                  FWA OF FORMAL PARAMETERS
      LDA ICNT,I      SET ICNT=-ICNT
      CMA,INA
      STA ICNT
      JSB LOCAL       CHECK FOR 2020 IN LOCAL
      DEF *+2                   RETURN ADDRESS
      DEF C2020                 2020 COMMAND CHANNEL
      JSB WRING       CHECK FOR 2020 WRITE WRING
      DEF *+2                   RETURN ADDRESS
      DEF C2020                 2020 COMMAND CHANNEL
      SFS 11B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      LDA WRO         GET WRITE RECORD ODD PARITY REQUEST
      OTA 11B         SEND TO 2020 COMMAND CHANNEL
      CLA             GET 6 BITS OF ZEROES
      JMP LABL1       GO TO LABL1 AND SEND TO 2020 DATA CHANNEL - 6.0 MIC SEC
LOOP1 CLA             GET 6 BITS OF ZEROES
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
LABL1 SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 1 - 10 TO 1 = 14.5 MIC SEC
      CLA             GET 6 BITS OF ZEROES
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 2 - 1 TO 2 = 8.0 MIC SEC
      LDA IBUF,I      GET TAPE NUMBER
      RAR,RAR         SHIFT B06-B11 INTO B00-B05
      RAR,RAR
      RAR,RAR
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 3 - 2 TO 3 = 16.0 MIC SEC
      LDA IBUF,I      GET TAPE NUMBER
      ISZ IBUF        SET IBUF FOR IBUF(I+1)
      ISZ ICNT        ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
      JMP *+2
      HLT 04B
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 4 - 3 TO 4 = 21.0 MIC SEC
      LDA IBUF,I      GET UPPER 16 BITS OF 32 BITS OF INFO
      RAL,RAL         SHIFT B14-B15 INTO B00-B01
      AND MASK1       MASK B00-B01
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 5 - 4 TO 5 = 16.0 MIC SEC
      LDA IBUF,I      GET UPPER 16 BITS OF 32 BITS OF INFO
      ALF,ALF         SHIFT B08-B13 INTO B00-B05
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 6 - 5 TO 6 = 12.0 MIC SEC
      LDA IBUF,I      GET UPPER 16 BITS OF 32 BITS OF INFO
      RAR,RAR         SHIFT B02-B07 INTO B00-B05
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 7 - 6 TO 7 = 12.0 MIC SEC
      LDA IBUF,I      GET UPPER 16 BITS OF 32 BITS OF INFO
      ISZ IBUF        SET IBUF FOR IBUF(I+1)
      ALF             SHIFT B00-B01 INTO B04-B05
      AND MASK2       MASK B04-B05
      STA B           MOVE TO B REG
      LDA IBUF,I      GET LOWER 16 BITS OF 32 BITS OF INFO
      ALF             SHIFT B12-B15 INTO B00-B03
      AND MASK3       MASK B00-B03
      IOR B           OR TOGETHER A AND B REGISTERS
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
      OTA 10B,C       SEND TO 2020 DATA CHANNEL - FRAME 8 - 7 TO 8 = 38.5 MIC SEC
      ISZ ICNT        ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
      JMP *+2
      HLT 04B
      LDA IBUF,I      GET LOWER 16 BITS OF 32 BITS OF INFO
      RAR,RAR         SHIFT B06-B11 INTO B00-B05
      RAR,RAR
      RAR,RAR
      SFC 10B         IF PREVIOUS WRITE FINISHED THEN HALT
      HLT 03B
      SFS 10B         WAIT UNTIL 2020 AVAILABLE
      JMP *-1
```

```
       OTA 10B,C      SEND TO 2020 DATA CHANNEL - FRAME 9 - 8 TO 9 = 22.5 MIC SEC
       LDA IBUF,I     GET LOWER 16 BITS OF 32 BITS OF INFO
       ISZ IBUF       SET IBUF FOR IBUF(I+1)
       SFC 10B        IF PREVIOUS WRITE FINISHED THEN HALT
       HLT 03B
       SFS 10B        WAIT UNTIL 2020 AVAILABLE
       JMP *-1
       OTA 10B,C      SEND TO 2020 DATA CHANNEL - FRAME 10 - 9 TO 10 = 14.5 MIC SE
       ISZ ICNT       ICNT=ICNT+1 THEN IF ICNT=0 SKIP NEXT INSTRUCTION
       JMP LOOP1      GO TO LOOP1 AND PROCESS MORE DATA
       CLC 10B        CLEAR CONTROL BIT FOR 2020 TAPE
       LDA CLR        GET CLEAR REQUEST
       OTA 11B        SEND TO 2020 COMMAND CHANNEL
       SFS 11B        WAIT UNTIL 2020 CONTROLLER NOT BUSY
       JMP *-1
       LIA 11B        GET STATUS OF 2020 FROM COMMAND CHANNEL
       STA ISTA,I     STORE STATUS FOR RETURN
       JMP WRTUT,I    RETURN
       SPC 2
C2020  OCT 000011
CLR    OCT 000000
MASK1  OCT 000003     0/000/000/000/000/011
MASK2  OCT 000060     0/000/000/000/110/000
MASK3  OCT 000017     0/000/000/000/001/111
WRO    OCT 000071
       END
```

```
ASMB,R,B            IOC       THRSYSP
      HED IOC - PROCESS IOC REQUEST
      NAM IOC
      ENT IOC
      EXT GETAP
      EXT .IOC.
      EXT WAITA
      EXT PMTTY
      SPC 2
*
***** IOC - PROCESS IOC REQUEST
*                   (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     IOCSC         IOC SELECT CODE ADDRESS
*     IOCRQ         IOC REQUEST ADDRESS
* OPT ISTA          IOC STATUS AFTER COMPLETION ADDRESS
* OPT ITRL          IOC TRANSMISSION LOG AFTER COMPLETION ADDRESS
* OPT IBUF          IOC READ/WRITE BUFFER ADDRESS
* OPT ICNT          IOC READ/WRITE BUFFER LENGTH ADDRESS
*
***** STANDARD CALLS:
* 00  CLEAR
*     CALL  IOC     ( IOCSC,000000B )
*     CALL  IOC     ( IOCSC,000000B,ISTA,ITRL )
* 01  READ
*     CALL  IOC     ( IOCSC,01XX00B,ISTA,ITRL,IBUF,ICNT )
* 02  WRITE
*     CALL  IOC     ( IOCSC,02XX00B,ISTA,ITRL,IBUF,ICNT )
* 03  POSITION
*     CALL  IOC     ( IOCSC,03XX00B )
*     CALL  IOC     ( IOCSC,03XX00B,ISTA,ITRL )
* 04  STATUS
*     CALL  IOC     ( IOCSC,040000B )
*     CALL  IOC     ( IOCSC,040000B,ISTA,ITRL )
*
      SPC 2
A     EQU 0
B     EQU 1
IOCSC BSS 1         IOC SELECT CODE ADDRESS
IOCRQ BSS 1         IOC REQUEST ADDRESS
ISTA  BSS 1         IOC STATUS AFTER COMPLETION ADDRESS
ITRL  BSS 1         IOC TRANSMISSION LOG AFTER COMPLETION ADDRESS
IBUF  BSS 1         IOC READ/WRITE BUFFER ADDRESS
ICNT  BSS 1         IOC READ/WRITE BUFFER LENGTH ADDRESS
      SPC 2
IOC   BSS 1         ENTRY/EXIT LINE
      JSB GETAP     GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IOCSC               FWA OF FORMAL PARAMETERS
      ADA DM2       SET NUMAP=NUMBER OF ACTUAL PARAMETERS PASSED - 2
      STA NUMAP     (A REG=NUMBER OF PARAMETERS PROCESSED BY GETAP)
      LDA IOCRQ,I   GET IOC REQUEST
      AND MASK1     MASK B06-B15
      IOR IOCSC,I   OR IN IOC SELECT CODE
      STA OCTI      STORE IOC REQUEST IN .IOC. CALL
      AND MASK2     SET CODE=B12-B15 OF IOC REQUEST WORD
      STA CODE
      LDB JMPC1     SET B REG=JUMP TO CALL1 INSTRUCTION FOR REJECT ADDRESS
      CPA CLEAR     IF CODE EQ CLEAR THEN SET B REG=NOP INSTRUCTION
      CLB
      CPA STAT      IF CODE EQ STATUS THEN SET B REG=NOP INSTRUCTION
      CLB
      STB JMPI      STORE REJECT ADDRESS/NOP IN .IOC. CALL
      CLA           STORE NOP FOR BUFFER ADDRESS AND BUFFER LENGTH
      STA DEFI
      STA DECI
      LDA CODE      IF CODE EQ CLEAR THEN GO TO CALL1 AND CALL .IOC.
      CPA CLEAR
      JMP CALL1
      CPA POS       IF CODE EQ POSITION THEN GO TO CALL1 AND CALL .IOC.
      JMP CALL1
      CPA STAT      IF CODE EQ STATUS THEN GO TO CALL1 AND CALL .IOC.
      JMP CALL1
      LDA IBUF      GET BUFFER ADDRESS
      STA DEFI      STORE BUFFER ADDRESS IN .IOC. CALL
      LDA ICNT,I    GET BUFFER LENGTH
      STA DECI      STORE BUFFER LENGTH IN .IOC. CALL
CALL1 JSB .IOC.     CALL .IOC. AND EXECUTE REQUEST
OCTI  OCT 000000              IOC REQUEST
JMPI  JMP CALL1               IF BUSY THEN RE-SUBMIT
DEFI  DEF 000000              BUFFER ADDRESS
DECI  DEC 0                   BUFFER LENGTH
      LDA NUMAP     IF NOT AT LEAST 4 ACTUAL PARAMETERS THEN RETURN
      ADA DM2
      SSA
      JMP IOC,I
      JSB WAITA     WAIT UNTIL UNIT AVAILABLE AND STORE STATUS
      DEF *+4                 RETURN ADDRESS
      DEF IOCSC,I             IOC SELECT CODE
      DEF ISTA,I              STATUS AFTER AVAILABLE
      DEF ITRL,I              TRANSMISSION LOG AFTER AVAILABLE
      JMP IOC,I     RETURN
      SPC 2
CLEAR OCT 000000
CODE  BSS 1
DM2   DEC -2
D2    DEC 2
D4    DEC 4
JMPC1 JMP CALL1
MASK1 OCT 177700    1/111/111/111/000/000
MASK2 OCT 170000    1/111/000/000/000/000
NUMAP BSS 1
POS   OCT 030000
READ  OCT 010000
STAT  OCT 040000
WRITE OCT 020000
      END
```

```
ASMB,R,B              MTCCR     TWRSYSP
      HED MTCCR = PROCESS MAGNETIC TAPE COMMAND CHANNEL REQUEST
      NAM MTCCR
      ENT MTCCR
      EXT GETAP
      EXT LOCAL
      EXT PMTTY
      SPC 2
*
***** MTCCR = PROCESS MAGNETIC TAPE COMMAND CHANNEL REQUEST
*                  (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     IMTCC              MAGNETIC TAPE COMMAND CHANNEL ADDRESS
*     CCREQ              MAGNETIC TAPE COMMAND CHANNEL REQUEST ADDRESS
* OPT ISTA               MAGNETIC TAPE COMMAND CHANNEL STATUS AFTER COMPLETION ADDR
*
***** STANDARD CALLS:
*     CALL  MTCCR   ( IMTCC,CCREQ )
*     CALL  MTCCR   ( IMTCC,CCREQ,ISTA )
*
      SPC 2
A     EQU 0
B     EQU 1
IMTCC BSS 1              MAGNETIC TAPE COMMAND CHANNEL ADDRESS
CCREQ BSS 1              MAGNETIC TAPE COMMAND CHANNEL REQUEST ADDRESS
ISTA  BSS 1              MAGNETIC TAPE COMMAND CHANNEL STATUS AFTER COMPLETION ADDR
      SPC 2
MTCCR BSS 1              ENTRY/EXIT LINE
      JSB GETAP          GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IMTCC                    FWA OF FORMAL PARAMETERS
      ADA DM2            SET NUMAP=NUMBER OF ACTUAL PARAMETERS PASSED - 2
      STA NUMAP          (A REG=NUMBER OF PARAMETERS PROCESSED BY GETAP)
      LDA IMTCC,I        GET MAGNETIC TAPE COMMAND CHANNEL
      IOR SFSI           OR IN SFS INSTRUCTION SKELETON
      STA ISFS           STORE IN INSTRUCTION STACK
      STA JSFS
      LDA IMTCC,I        GET MAGNETIC TAPE COMMAND CHANNEL
      IOR OTAI           OR IN OTA INSTRUCTION SKELETON
      STA IOTA           STORE IN INSTRUCTION STACK
      LDA IMTCC,I        GET MAGNETIC TAPE COMMAND CHANNEL
      IOR LIAI           OR IN LIA INSTRUCTION SKELETON
      STA ILIA           STORE IN INSTRUCTION STACK
      JSB LOCAL          CHECK MAGNETIC TAPE IN LOCAL
      DEF *+2                      RETURN ADDRESS
      DEF IMTCC,I                  MAGNETIC TAPE COMMAND CHANNEL
ISFS  SFS 00B            WAIT UNTIL MAGNETIC TAPE COMMAND CHANNEL AVAILABLE
      JMP *-1
      LDA CCREQ,I        GET MAGNETIC TAPE COMMAND CHANNEL REQUEST
IOTA  OTA 00B            SEND TO MAGNETIC TAPE COMMAND CHANNEL
      LDA NUMAP          IF NUMAP=0 THEN RETURN
      SZA,RSS
      JMP MTCCR,I
JSFS  SFS 00B            WAIT UNTIL MAGNETIC TAPE COMMAND CHANNEL AVAILABLE
      JMP *-1
ILIA  LIA 00B            GET STATUS FROM MAGNETIC TAPE COMMAND CHANNEL
      STA ISTA,I         STORE MAGNETIC TAPE COMMAND CHANNEL STATUS FOR RETURN
      JMP MTCCR,I        RETURN
      SPC 2
DM2   DEC -2
D2    DEC 2
LIAI  LIA 00B            LIA INSTRUCTION SKELETON
NUMAP BSS 1
OTAI  OTA 00B            OTA INSTRUCTION SKELETON
SFSI  SFS 00B            SFS INSTRUCTION SKELETON
      END
```

```
ASMB,R,B              LOCAL     TWRSYSP
      HED LOCAL = CHECK MAGNETIC TAPE FOR LOCAL STATUS
      NAM LOCAL
      ENT LOCAL
      EXT GETAP
      EXT PMTTY
      EXT PRUN
      SPC 2
*
***** LOCAL = CHECK MAGNETIC TAPE FOR LOCAL STATUS
*                  (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     IMTCC              MAGNETIC TAPE COMMAND CHANNEL ADDRESS
*
***** STANDARD CALL:
*     CALL  LOCAL   ( IMTCC )
*
      SPC 2
IMTCC BSS 1              MAGNETIC TAPE COMMAND CHANNEL ADDRESS
      SPC 2
LOCAL BSS 1              ENTRY/EXIT LINE
      JSB GETAP          GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IMTCC                    FWA OF FORMAL PARAMETERS
      LDA IMTCC,I        GET MAGNETIC TAPE COMMAND CHANNEL
      IOR LIAI           OR IN LIA INSTRUCTION SKELETON
      STA LABL1          STORE IN INSTRUCTION STACK
LABL1 LIA 00B            GET STATUS OF MAGNETIC TAPE FROM COMMAND CHANNEL
      ALF,ALF            IF MAGNETIC TAPE NOT IN LOCAL STATUS THEN RETURN
      SLA,RSS
      JMP LOCAL,I
      JSB PMTTY          =TAPE IN LOCAL (A REG = MAGNETIC TAPE COMMAND CHANNEL)=
      DEF MSG1                     MESSAGE 1 ADDRESS
      DEC 27                       MESSAGE 1 LENGTH - 53 CHARACTERS
      JSB PRUN           =PRESS =PRESET= AND =RUN= WHEN READY=
      LDA IMTCC,I        GET MAGNETIC TAPE COMMAND CHANNEL
      CLB                CLEAR B REG
      HLT 00B            HALT AND WAIT FOR RUN
      JMP LABL1          GO TO LABL1 AND CHECK AGAIN
      SPC 2
LIAI  LIA 00B            LIA INSTRUCTION SKELETON
MSG1  ASC 27,TAPE IN LOCAL (A REG = MAGNETIC TAPE COMMAND CHANNEL)
      END
```

```
ASMB,R,B             WRING      TWRSYSP
      HED WRING - CHECK MAGNETIC TAPE FOR WRITE RING
      NAM WRING
      ENT WRING
      EXT GETAP
      EXT PMTTY
      EXT PRUN
      SPC 2
*
***** WRING - CHECK MAGNETIC TAPE FOR WRITE RING
*                    (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     IMTCC          MAGNETIC TAPE COMMAND CHANNEL ADDRESS
*
***** STANDARD CALL:
*     CALL  WRING    ( IMTCC )
*
      SPC 2
IMTCC BSS 1          MAGNETIC TAPE COMMAND CHANNEL ADDRESS
      SPC 2
WRING BSS 1          ENTRY/EXIT LINE
      JSB GETAP      GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IMTCC                FWA OF FORMAL PARAMETERS
      LDA IMTCC,I    GET MAGNETIC TAPE COMMAND CHANNEL
      IOR LIAI       OR IN LIA INSTRUCTION SKELETON
      STA LABL1      STORE IN INSTRUCTION STACK
LABL1 LIA 00B        GET STATUS OF MAGNETIC TAPE FROM COMMAND CHANNEL
      RAR,RAR        IF WRITE RING ENABLED THEN RETURN
      SLA,RSS
      JMP WRING,I
      JSB PMTTY      "NO WRITE RING (A REG = MAGNETIC TAPE COMMAND CHANNEL)"
      DEF MSG1                 MESSAGE 1 ADDRESS
      DEC 27                   MESSAGE 1 LENGTH = 53 CHARACTERS
      JSB PRUN       "PRESS "PRESET" AND "RUN" WHEN READY"
      LDA IMTCC,I    GET MAGNETIC TAPE COMMAND CHANNEL
      CLB            CLEAR B REG
      HLT 00B        HALT AND WAIT FOR RUN
      JMP LABL1      GO TO LABL1 AND CHECK AGAIN
      SPC 2
LIAI  LIA 00B        LIA INSTRUCTION SKELETON
MSG1  ASC 27,NO WRITE RING (A REG = MAGNETIC TAPE COMMAND CHANNEL)
      END
```

```
ASMB,R,B             PRUN       TWRSYSP
      HED PRUN - PRINT "PRESS "PRESET" AND "RUN" WHEN READY" ON TTY
      NAM PRUN
      ENT PRUN
      EXT PMTTY
      SPC 2
*
***** PRUN - PRINT "PRESS "PRESET" AND "RUN" WHEN READY" ON TTY
*                    (ASSEMBLY CALLABLE SUBROUTINE)
*
***** STANDARD CALL:
*     JSB PRUN       "PRESS "PRESET" AND "RUN" WHEN READY"
*     <NORMAL RETURN>
*
      SPC 2
PRUN  BSS 1          ENTRY/EXIT LINE
      JSB PMTTY      "PRESS "PRESET" AND "RUN" WHEN READY"
      DEF MSG1                 MESSAGE 1 ADDRESS
      DEC 18                   MESSAGE 1 LENGTH = 35 CHARACTERS
      JMP PRUN,I     RETURN
      SPC 2
MSG1  ASC 18,PRESS "PRESET" AND "RUN" WHEN READY
      END
```

```
ASMB,R,B            PMTTY     TWRSYSP
      HED PMTTY - PRINT MESSAGE ON TTY
      NAM PMTTY
      ENT PMTTY
      EXT .IOC.
      EXT WAITA
      SPC 2
*
***** PMTTY - PRINT MESSAGE ON TTY
*                    (ASSEMBLY CALLABLE SUBROUTINE)
*
***** STANDARD CALL:
*     JSB PMTTY      -MESSAGE-
*     DEF -MSGA-     MESSAGE ADDRESS
*     DEC -MSGL-     MESSAGE LENGTH
*     <NORMAL RETURN>
*
      SPC 2
PMTTY BSS 1          ENTRY/EXIT LINE
      LDA PMTTY,I    GET MESSAGE ADDRESS
      ISZ PMTTY      INCREMENT ADDRESS AT PMTTY
      STA MSGA       STORE IN IOC CALL
      LDA PMTTY,I    GET MESSAGE LENGTH
      ISZ PMTTY      INCREMENT ADDRESS AT PMTTY
      STA MSGL       STORE IN IOC CALL
CALL1 JSB .IOC.      CALL IOC AND PRINT MESSAGE ON TTY
      OCT 020002               IOC WRITE ASCII TO TELEPRINTER OUTPUT REQUEST
      JMP CALL1                IF BUSY THEN RE-SUBMIT
MSGA  BSS 1                    MESSAGE ADDRESS
MSGL  BSS 1                    MESSAGE LENGTH
      JSB WAITA      WAIT UNTIL TTY AVAILABLE
      DEF *+2                  RETURN ADDRESS
      DEF STTY                 IOC TELEPRINTER OUTPUT SELECT CODE
      JMP PMTTY,I    RETURN
      SPC 2
STTY  OCT 000002
      END
```

```
ASMB,R,B            WAITA     TWRSYSP
      HED WAITA - WAIT UNTIL SPECIFIED UNIT AVAILABLE AND STORE STATUS
      NAM WAITA
      ENT WAITA
      EXT GETAP
      EXT .IOC.
      SPC 2
*
***** WAITA - WAIT UNTIL SPECIFIED UNIT AVAILABLE AND STORE STATUS
*                    (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     IOCSC          IOC SELECT CODE FOR SPECIFIED UNIT ADDRESS
*OPT  ISTA           IOC STATUS AFTER AVAILABLE ADDRESS
*OPT  ITRL           IOC TRANSMISSION LOG AFTER AVAILABLE ADDRESS
*
***** STANDARD CALLS:
*     CALL  WAITA    ( IOCSC )
*     CALL  WAITA    ( IOCSC,ISTA,ITRL )
*
      SPC 2
IOCSC BSS 1          IOC SELECT CODE FOR SPECIFIED UNIT ADDRESS
ISTA  BSS 1          IOC STATUS AFTER AVAILABLE ADDRESS
ITRL  BSS 1          IOC TRANSMISSION LOG AFTER AVAILABLE ADDRESS
      SPC 2
WAITA BSS 1          ENTRY/EXIT LINE
      JSB GETAP      GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IOCSC                FWA OF FORMAL PARAMETERS
      ADA DM1        SET A REG=NUMBER OF ACTUAL PARAMETERS - 1
      LDB RETJ       GET RETURN INSTRUCTION
      SZA            IF MORE THAN 1 PARAMETER THEN SET B REG FOR NOP
      CLB
      STB RETI       STORE NOP OR RETURN INSTRUCTION IN INSTRUCTION STACK
      LDA IOCSC,I    GET IOC SELECT CODE FOR SPECIFIED UNIT
      IOR STATI      OR IN IOC STATUS REQUEST SKELETON
      STA ISTAT      STORE IN INSTRUCTION STACK
LOOP1 JSB .IOC.      CALL IOC AND GET STATUS OF SPECIFIED UNIT
ISTAT OCT 040000               IOC STATUS REQUEST
      SSA            IF NOT AVAILABLE THEN GO TO LOOP1
      JMP LOOP1
RETI  NOP            NOP OR RETURN
      STA ISTA,I     STORE IOC STATUS AFTER AVAILABLE FOR RETURN
      RBL,CLE,ERB    MASK OFF B15 OF TRANSMISSION LOG
      STB ITRL,I     STORE IOC TRANSMISSION LOG AFTER AVAILABLE FOR RETURN
RETJ  JMP WAITA,I    RETURN
      SPC 2
DM1   DEC -1
STATI OCT 040000     IOC STATUS REQUEST SKELETON
      END
```

```
ASMB,R,B            GETSR     THRSYSP
      HED GETSR - GET SWITCH REGISTER
      NAM GETSR
      ENT GETSR
      EXT GETAP
      SPC 2
*
***** GETSR - GET SWITCH REGISTER
*                   (FORTRAN CALLABLE SUBROUTINE)
*
***** FORMAL PARAMETERS:
*     ISR           SWITCH REGISTER CONTENTS ADDRESS
*
***** STANDARD CALL:
*     CALL  GETSR   ( ISR )
*
      SPC 2
ISR   BSS 1         SWITCH REGISTER CONTENTS ADDRESS
      SPC 2
GETSR BSS 1         ENTRY/EXIT LINE
      JSB GETAP     GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF ISR                 FWA OF FORMAL PARAMETERS
      LIA 01B       GET CONTENTS OF SWITCH REGISTER
      STA ISR,I     STORE FOR RETURN
      JMP GETSR,I   RETURN
      END
```

```
ASMB,R,B            LSHIF     THRSYSP
      HED LSHIF - LEFT SHIFT IWORD ICNT TIMES
      NAM LSHIF
      ENT LSHIF
      EXT GETAP
      EXT SHIFT
      SPC 2
*
***** LSHIF - LEFT SHIFT IWORD ICNT TIMES
*                   (FORTRAN CALLABLE FUNCTION)
*
***** EXIT:
*     A   IWORD LEFT SHIFTED ICNT TIMES
*     B   0
*
***** FORMAL PARAMETERS:
*     IWORD         16 BIT WORD TO BE LEFT SHIFTED ADDRESS
*     ICNT          NUMBER OF TIMES TO LEFT SHIFT IWORD ADDRESS
*                             (-16 LE ICNT LE 16)
*
***** STANDARD CALLS:
*     I = LSHIF( I,5 )
*     J = LSHIF( I,-11 )
*                   IF ( LSHIF( ISR,15 ) )       1010 , 1020
*                   IF ( LSHIF( ISTA,LSEOF ) )   9010 , 2010
*
      SPC 2
IWORD BSS 1         16 BIT WORD TO BE LEFT SHIFTED ADDRESS
ICNT  BSS 1         NUMBER OF TIMES TO LEFT SHIFT IWORD ADDRESS
      SPC 2
LSHIF BSS 1         ENTRY/EXIT LINE
      JSB GETAP     GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      DEF IWORD               FWA OF FORMAL PARAMETERS
      LDA IWORD,I   SET A REG=16 BIT WORD TO BE LEFT SHIFTED
      LDB ICNT,I    SET B REG=NUMBER OF TIMES TO BE LEFT SHIFTED
      JSB SHIFT     SHIFT A REGISTER LEFT B REGISTER TIMES
      JMP LSHIF,I   RETURN
      END
```

```
ASMB,R,B              GETAP     T*RSYSP
      HED GETAP = GET ACTUAL PARAMETERS FROM CALLING ROUTINE
      NAM GETAP
      ENT GETAP
      SPC 2
*
***** GETAP - GET ACTUAL PARAMETERS FROM CALLING ROUTINE
*                   (ASSEMBLY CALLABLE SUBROUTINE)
*
***** ENTRY:
*     A   TRASH
*     B   TRASH
*
***** EXIT:
*     A   NUMBER OF ACTUAL PARAMETERS PROCESSED
*     B   0
*
***** STANDARD CALLING SEQUENCE FOR NO ACTUAL PARAMETERS:
*           JSB     SUB
*           DEF     *+1         APAR
*           <NORMAL RETURN FROM SUB>
*           ...
*           ...
*     SUB   NOP                 FPAR
*           JSB     GETAP
*           DEF     SUB
*           <NORMAL RETURN FROM GETAP>
*
***** STANDARD CALLING SEQUENCE FOR 1 TO M ACTUAL PARAMETERS:
*     APAR1 BSS     1
*     APAR2 BSS     1
*           ...
*     APARM BSS     1           M ACTUAL PARAMETERS
*           ...
*           ...
*           JSB     SUB
*           DEF     *+M+1       APAR
*           DEF     APAR1
*           DEF     APAR2
*           ...
*           DEF     APARM       M ACTUAL PARAMETERS
*           <NORMAL RETURN FROM SUB>
*           ...
*           ...
*     FPAR1 BSS     1           FPAR
*     FPAR2 BSS     1
*           ...
*     FPARN BSS     1           N FORMAL PARAMETERS
*     SUB   NOP
*           JSB     GETAP
*           DEF     FPAR1
*           <NORMAL RETURN FROM GETAP>
*
      SPC 2
A     EQU 0
B     EQU 1
      SPC 2
GETAP BSS 1            ENTRY/EXIT LINE
      LDB GETAP,I      SET FPAR FOR (FPAR1 BSS 1)
      STB FPAR
      INB              SET NUMFP=NUMBER OF FORMAL PARAMETERS
      CMB
      ADB GETAP
      STB NUMFP
      ADB FPAR         SET B REG FOR (SUB NOP)
      LDA B,I          SET APAR FOR (DEF *+M+1)
      STA APAR
      IOR IBIT         OR IN INDIRECT BIT
      STA B,I          STORE INDIRECT RETURN ADDRESS AT (SUB NOP)
      RAL,CLE,ERA      REMOVE INDIRECT BIT FROM APAR
      CMA              SET NUMAP=NUMBER OF ACTUAL PARAMETERS
      ADA APAR,I
      STA NUMAP
      STA B            SET B REG=NUMAP
      CMA,INA          IF NUMAP GT NUMFP THEN SET B REB=NUMFP
      ADA NUMFP
      SSA
      LDB NUMFP
      STB NUMAP        SET NUMAP=NUMBER OF PARAMETERS TO BE PROCESSED
      CMB              SET B REG=-NUMAP-1
      ISZ GETAP        INCREMENT RETURN ADDRESS AT ENTRY POINT
LOOP1 LDA NUMAP        SET A REG=NUMAP FOR RETURN
      INB,SZB,RSS      B REG=B REG+1 THEN IF B REG=0 RETURN
      JMP GETAP,I      RETURN
      ISZ APAR         SET APAR FOR NEXT ACTUAL PARAMETER
      LDA APAR,I       GET ADDRESS OF NEXT ACTUAL PARAMETER
DIRAD RAL,CLE,ERA      REMOVE INDIRECT BIT
      SEZ              IF INDIRECT THEN GO TO INDIR
      JMP INDIR
      STA FPAR,I       STORE DIRECT ADDRESS IN FORMAL PARAMETER LIST
      ISZ FPAR         SET FPAR FOR NEXT FORMAL PARAMETER
      JMP LOOP1        GO TO LOOP1 AND PROCESS NEXT ACTUAL PARAMETER
INDIR LDA A,I          GET ADDRESS
      JMP DIRAD        GO TO DIRAD AND CHECK FOR DIRECT ADDRESS
      SPC 2
APAR  BSS 1
FPAR  BSS 1
IBIT  OCT 100000
NUMAP BSS 1
NUMFP BSS 1
      END
```

```
ASMB,R,B              SHIFT      THRSYSP
      HED SHIFT - SHIFT A REGISTER LEFT B REGISTER TIMES
      NAM SHIFT
      ENT SHIFT
      SPC 2
*
***** SHIFT - SHIFT A REGISTER LEFT B REGISTER TIMES
*                 (ASSEMBLY CALLABLE SUBROUTINE)
*
***** ENTRY:
*     A   16 BIT WORD TO BE LEFT SHIFTED B REGISTER TIMES
*     B   NUMBER OF TIMES TO LEFT SHIFT A REGISTER
*                 (-16 LE B REGISTER LE 16)
*
***** EXIT:
*     A   ORIGINAL A REGISTER LEFT SHIFTED B REGISTER TIMES
*     B   0
*
***** STANDARD CALLS:
*     LDA WORD
*     LDB =D5
*     JSB SHIFT
*     <NORMAL RETURN>
*     ...
*     ...
*     LDA WORD
*     LDB =D-11
*     JSB SHIFT
*     <NORMAL RETURN>
*
      SPC 2
SHIFT BSS 1            ENTRY/EXIT LINE
      SSB              IF B REG LT 0 THEN B REG=B REG+16
      ADB D16
      SZB,RSS          IF B REG=0 THEN RETURN
      JMP SHIFT,I
      CMB,INB          B REG=-B REG
LOOP1 RAL              ROTATE A REG LEFT 1 BIT
      INB,SZB          B REG=B REG+1 THEN IF B REG=0 SKIP NEXT INSTRUCTION
      JMP LOOP1        GO TO LOOP1
      JMP SHIFT,I      RETURN
      SPC 2
D16   DEC 16
      END
```

```
      PROGRAM    DVHPRO ( INPUT=65,OUTPUT=513,TAPE77=513,TAPE66=513,
     *                    TAPE1=513,TAPE2=513,TAPE3=513,TAPE4=513,
     *                    TAPE5=513 )
C
C-----COMPILE(MNF)       FL = 55000
C-----LOAD(MAP=PART)     FL = 34000
C-----LOAD(MAP=ON)       FL = 36000
C-----EXECUTE            FL = 22000
C
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),ITEST(501)
      DIMENSION         ISTART(5),LINE(8),NUMW(5)
      EQUIVALENCE       (IEOF,IEOR),(IREWIN,IT),(IWHAT,IWORD),
     *                  (ISTART,IDEN),(LINE,ITEST),(NUMW,IDIGIT)
C
C-----STOP 801 = NOTHING ON INPUT
C-----STOP 802 = NOTHING ON TAPE 77
C-----STOP 803 = MORE THAN 6 WORD IDENT ON TAPE 77
C-----STOP 804 = NO EOR ENCOUNTERED AFTER IDENT ON TAPE 77
C-----STOP 805 = NO END-OF-RECORD ENCOUNTERED AFTER 501 WORDS
C
C-----STOP 901 = INCORRECT IDENT ON TAPE 77 OR INPUT
C-----STOP 902 = UNABLE TO RENAME HEADWAY FILE IN CHGFILE
C-----STOP 903 = ILLEGAL TIME VALUE IN REPACK
C
      DATA      IAVGDY / 8HAVGDELAY /
      DATA      IHEADW / 8HHEADWAY  /
      DATA      INPUT  / 5LINPUT    /
      DATA      ISIGNL / 8HSIGNAL   /
      DATA      ITOTDY / 8HTOTDELAY /
      DATA      IUSE66 / 0          /
      DATA      IVOLUM / 8HVOLUME   /
      DATA      M01    / 000000001B /
      DATA      M03    / 000000007B /
      DATA      M04    / 000000017B /
      DATA      M12    / 000007777B /
      DATA      M13    / 000017777B /
      DATA      ODD    / 3HODD      /
      DATA      RB     / 2HRB       /
      DATA      RC     / 2HRC       /
      DATA      RET    / 3HRET      /
      DATA      REW    / 3HREW      /
      DATA      RR     / 2HRR       /
      DATA      WB     / 2HWB       /
      DATA      WC     / 2HWC       /
      DATA      WF     / 2HWF       /
  501 FORMAT(6A10,2I1,I3,5I2)
  502 FORMAT(A10,4I1,6A10)
  601 FORMAT(1H1,19X,21HLOOKING FOR IDENT = [,6A10,1H]/)
  602 FORMAT(27X,14HTAPE IDENT = [,6A10,1H]//)
  603 FORMAT(27X,8HISTART =,I5,7H NUMW =,I6,9H FOR TAPE,I2)          *DEBUG*
  604 FORMAT(/27X,8HITSTRT =,I5/27X,8HITSTOP =,I5,/27X,8HMNUMW  =,I5,
     *       /1H1)
  801 FORMAT(//20X,22HINCORRECT TAPE NUMBER(,I3,1H))
  802 FORMAT(//20X,25HINCORRECT CHANNEL NUMBER(,I3,1H))
  803 FORMAT(//20X,17HINCORRECT INPUT :,A10)
  804 FORMAT(//20X,40HINCORRECT NUMBER OF SIGNAL INDICATIONS =,I3)
  901 FORMAT(//20X,31HCORRECT IDENT NOT FOUND ON TAPE//)
      CALL  IOP      ( ODD,77 )
C1010 CONTINUE
      ITSTRT = -1
      ITSTOP = 8192
      DO 1020  IT = 1 , 5
      CALL  IOP      ( REW,IT )
      NUMW(IT) = 0
      DO 1020  IC = 1 , 4
      DO 1020  ID = 1 , 4
      STAT(IT,IC,ID) = -1.0
 1020 CONTINUE
      IEOR = IOP     ( RC,INPUT,LINE,8 )
      NLINE = 1
                     IF ( IEOR . NE . 0 )           GO TO 8010
            DECODE   ( 75,501,LINE )
     *               IDEN,NINPUT,IREWIN,INTVL,IBAD
                     IF ( IREWIN . EQ . 1 )         CALL IOP ( REW,77 )
      INTVL = INTVL * 60
      PRINT 601 , IDEN
      IEOR = IOP     ( RB,77,IDENT,6 )
                     IF ( IEOR . NE . 0 )           STOP 802
C1030 CONTINUE
      IEOR = IOP     ( RB,77,ITEST,1 )
                     IF ( IEOR . EQ . 0 )           STOP 803
      IEOR = IOP     ( RR,77 )
                     IF ( IEOR . EQ . 1 )           STOP 804
      PRINT 602 , IDENT
C-----FIND CORRECT RECORD ON TAPE
      DO 1040  ID = 1 , 6
                     IF ( IDEN(ID).NE.IDENT(ID) ) GO TO 1050
 1040 CONTINUE
C-----CORRECT RECORD LOCATED
      GO TO 2010
 1050 CONTINUE
      GO TO 9010
C-----WRONG IDENT - READ ENTIRE RECORD + 1
C     NEOF = 0
C     IEOR = IOP     ( RB,77,ITEST,501 )
C                    IF ( IEOR . EQ . 0 )           STOP 805
C1060 CONTINUE
C     IEOF = IOP     ( RR,77 )
C-----CHECK FOR EOF CONDITION
C                    IF ( IEOF . EQ . 0 )           GO TO 1050
C-----1050: NO END OF FILE ENCOUNTERED
C     NEOF = NEOF + 1
C                    IF ( NEOF . EQ . 2 )           GO TO 9010
C-----9010: CORRECT IDENT NOT FOUND ANYWHERE ON TAPE
C     IEOR = IOP     ( RB,77,IDENT,6 )
C                    IF ( IEOR . EQ . 0 )           GO TO 1030
C     GO TO 1060
 2010 CONTINUE
C-----CORRECT RECORD LOCATED
      IEOR = IOP     ( RB,77,ITEST,501 )
C-----READ 501 WORDS
                     IF ( IEOR . EQ . 0 )           STOP 805
      NUM = 501 - IEOR
C-----NUM = NUMBER OF WORDS ACTUALLY READ
                     IF ( NUM . EQ . 0 )            GO TO 2030
      DO 2020  IN = 1 , NUM
      IWORD = ITEST(IN)
      IT = LSHIFT(IWORD,60-36) . AND . M03
                     IF ( IT . LE . 0 )             GO TO 2020
                     IF ( IT . GT . NINPUT )        GO TO 2020
                     IF ( IBAD(IT) . NE . 0 )       GO TO 2020
C-----WRITE WORD ONTO CORRECT TAPE
      CALL  IOP      ( WB,IT,IWORD,1 )
      NUMW(IT) = NUMW(IT) + 1
                     IF ( NUMW(IT) . EQ . 61 )
     *               ISTART(IT) = LSHIFT(IWORD,60-16) . AND . M13
 2020 CONTINUE
C-----FINAL WORD SORTED
 2030 CONTINUE
      IEOF = IOP     ( RR,77 )
                     IF ( IEOF . EQ . 0 )           GO TO 2010
C-----WRITE END OF FILE ON AND REWIND TAPES 1 THRU 6
      DO 2040  IT = 1 , NINPUT
      CALL  IOP      ( WF,IT )
      CALL  IOP      ( REW,IT )
 2040 CONTINUE
      NUM = 0
```

```
 2050 CONTINUE
                      IF ( ITSTRT . NE . -1 )       GO TO 2070
      NUM = NUM + 300
      DO 2060  IT = 1 , NINPUT
                      IF ( IBAD(IT) . NE . 0 )      GO TO 2060
                      IF ( ISTART(IT) . GT . ITSTRT . AND .
     *                     ISTART(IT) . LT . NUM ) ITSTRT = ISTART(IT)
 2060 CONTINUE
      GO TO 2050
 2070 CONTINUE
      MNUMW = 100000
      DO 2080  IT = 1 , NINPUT
                      IF ( IBAD(IT) . NE . 0 )      GO TO 2080
                      IF ( NUMW(IT) . LT . MNUMW ) MNUMW = NUMW(IT)
      PRINT 603 , ISTART(IT),NUMW(IT),IT                                  *DEBUG*
 2080 CONTINUE
      DO 2090  IT = 1 , NINPUT
                      IF ( IBAD(IT) . NE . 0 )      GO TO 2090
                      IF ( ITSTRT+MNUMW . GT . ISTART(IT)+NUMW(IT) )
     *                MNUMW = NUMW(IT) - (ITSTRT-ISTART(IT))
 2090 CONTINUE
      ITSTOP = ITSTRT + MNUMW - 20
      PRINT 604 , ITSTRT,ITSTOP,MNUMW
C-----TAPES 1 THRU 6 NOW CONTAIN NUMW WORDS OF DATA
 3010 CONTINUE
      IEOR = IOP     ( RC,INPUT,LINE,8 )
                      IF ( IEOR . NE . 0 )          GO TO 8010
      NLINE = NLINE + 1
C-----GO TO 8010 IF EOR ENCOUNTERED
             DECODE  ( 74,502,LINE )               IWHAT,IT,IC,ID,IN,IDEN
                      IF ( IT . LT . 1 )            GO TO 3060
                      IF ( IT . GT . NINPUT )       GO TO 3060
                      IF ( IBAD(IT) . NE . 0 )      GO TO 3010
                      IF ( IWHAT . NE . IAVGDY )    GO TO 3020
                      IF ( IBAD(ID) . NE . 0 )      GO TO 3010
C-----IT,IC = LOCATION OF VOLUME STATISTICS IN STAT( , ,1)
C-----ID,IN = LOCATION OF DELAY  STATISTICS IN STAT( , ,2)
      CALL  DELAYA   ( IT,IC,ID,IN )
      GO TO 3010
 3020 CONTINUE
                      IF ( IWHAT . EQ . ISIGNL )    GO TO 3030
                      IF ( IC.NE.1.AND.IC.NE.2 )    GO TO 3070
      ITYPE(IC) = IN
                      IF ( IWHAT . NE . IHEADW )    GO TO 3030
      CALL  HEADWA   ( IT,IC,ID )
      IUSE66 = 1
      GO TO 3010
 3030 CONTINUE
      PRINT 602 , IDENT
                      IF ( IWHAT . NE . IVOLUM )    GO TO 3040
      CALL  VOLUM    ( IT,IC,ID )
      GO TO 3010
 3040 CONTINUE
                      IF ( IWHAT . NE . ITOTDY )    GO TO 3050
      CALL  DELAYT   ( IT,IC )
      GO TO 3010
 3050 CONTINUE
                      IF ( IWHAT . NE . ISIGNL )    GO TO 3080
      NUM = IC*10 + ID
                      IF ( NUM . LT .  1 )          GO TO 3090
                      IF ( NUM . GT . 12 )          GO TO 3090
      CALL  SIGNAL   ( IT,NUM )
      GO TO 3010
 3060 CONTINUE
      PRINT 801 , IT
      GO TO 3010
 3070 CONTINUE
      PRINT 802 , IC
      GO TO 3010
 3080 CONTINUE
      PRINT 803 , IWHAT
      GO TO 3010
 3090 CONTINUE
      PRINT 804 , NUM
      GO TO 3010
 8010 CONTINUE
      DO 8020  IT = 1 , 5
      CALL  IOP      ( RET , IT )
 8020 CONTINUE
                      IF ( IUSE66 . EQ . 1 )        GO TO 8030
      CALL  IOP      ( RET , 66 )
 8030 CONTINUE
                      IF ( NLINE . GE . 2 )         CALL  EXIT
      STOP  801
 9010 CONTINUE
      PRINT 901
C     GO TO 1010
      STOP  901
      END
                                                                  DVHPRO
```

```
      SUBROUTINE VOLUM  ( ITT,ICC,IDD )
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,IONESL,IVOL,JJ,JONES,JTIME,
     *                  VOLEQ,VOLUME,ZTEMPD(419)
  601 FORMAT(//40X,21HINFORMATION FROM TAPE,I2,8H CHANNEL,I2,6H DIGIT,
     *       I2//40X,6A10//)
  602 FORMAT(30X,8HVOLUME =,I5,16H VEHICLES AFTER ,I2,6H HOUR,,I3,
     *       9H MINUTES,,I3,8H SECONDS)
  603 FORMAT(//30X,20HEQUIVALENT VOLUME = ,F4.0,18H VEHICLES PER HOUR,/
     *       30X,I5,41H MISSING DATA POINTS WERE SYNTHESIZED   (,
     *       F5.2,9H PERCENT),/1H1)
  901 FORMAT(//5X,29HNO DATA : 9010 ERROR IN VOLUM)
      IT = ITT
      IC = ICC
      ID = IDD
C-----IT = TAPE NUMBER
C-----IC = CHANNEL NUMBER
C-----ID = DIGIT(1 = ONES  :  2 = TENS)
      PRINT 601 , IT,IC,ID,IDEN
      CALL  POSITON
                    IF ( IEOR . NE . 0 )         GO TO 9010
      IVOL = 0
      IONESL = IDIGIT(ID,IC)
 1010 CONTINUE
      IEOR = IOP     ( RB,IT,IWORD,1 )
                    IF ( IEOR . NE . 0 )         GO TO 2010
      CALL  UNPACK
                    IF ( IDIGIT(ID,IC) . GT . 9 )GO TO 1010
                    IF ( (ITHOLD-ITIME).LE.100 ) GO TO 1020
C-----ITMAX IS THE CUMULATIVE VALUE OF ANY BACKWARD SKIP ON THE CLOCK
      ITMAX = ITMAX + ITHOLD - ITIME + 1
 1020 CONTINUE
                    IF ( (ITIME-ITHOLD).LT.60 )  GO TO 1030
C-----JUMP IS THE CUMULATIVE VALUE OF A FORWARD SKIP ON THE CLOCK
C-----GREATER THAN 60 SECONDS
      JUMP = JUMP + ITIME - ITHOLD - 1
 1030 CONTINUE
      ITHOLD = ITIME
      ITIME = ITIME + ITMAX - JUMP
                    IF ( ITIME . LE . ITIMEL )   GO TO 1010
                    IF ( ITIMEL . LT . INTVLC )  GO TO 1040
      CALL  DELTA    ( ITSTRT,INTVLC )
      PRINT 602 , IVOL,ITHOUR,ITMIN,ITSEC
      INTVLC = INTVLC + INTVL
 1040 CONTINUE
C-----1040 TO 1050 - MISSING DATA ROUTINE
      JONES = IDIGIT(ID,IC)
      JTIME = ITIME
                    IF ( JTIME . EQ . ITIMEL+1 ) GO TO 1050
      MISS = MISS + 1
                    IF ( JONES . LT . IONESL-5 ) JONES = JONES + 10       8,9, ,0
                    IF ( IONESL . LT . JONES-5 ) JONES = JONES - 10       1,0, ,9
      JONES = IONESL + (FLOAT(JONES-IONESL)/FLOAT(JTIME-ITIMEL)+0.499)
      JTIME = ITIMEL + 1
 1050 CONTINUE
C$                  IF ( JTIME . GT . ITSTOP )   GO TO 2010               ALLDATA
                    IF ( JONES-IONESL . LT . -6 )JONES = JONES + 10       9,0,1
           IF ( IABS(JONES-IONESL) . GT . 3 )    JONES = IONESL           1,1,8,2
      IF ( IONESL . NE . JONES )
     *              IVOL = IVOL + IABS(JONES-IONESL)
                    IF ( JONES . GE . 10 )       JONES = JONES - 10
                    IF ( JONES . LT . 0 )        JONES = JONES + 10
      IONESL = JONES
      ITIMEL = JTIME
                    IF ( ITIME . EQ . ITIMEL )   GO TO 1010
      GO TO 1040
 2010 CONTINUE
C-----PRINT VOLUME STATISTICS
      CALL  DELTA    ( INITIAL,JTIME )
      TIME = JTIME - INITIAL
      VOLUME = IVOL
      VOLEQ = 3600.0*VOLUME/TIME
      PMISS = 100.0*FLOAT(MISS)/TIME
      JJ = (IC-1)*2 + ID
      STAT(IT,JJ,1) = VOLUME
      STAT(IT,JJ,3) = TIME
      PRINT 602 , IVOL,ITHOUR,ITMIN,ITSEC
      PRINT 603 , VOLEQ,MISS,PMISS
      RETURN
 9010 CONTINUE
      PRINT 901
      RETURN
      END
```

VOLUM

```
      SUBROUTINE DELAYT ( ITT,ICC )
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,CKOUNT,DELAY,DELTEQ,IDELT,
     *                  IKOUNT,IKOUNTL,ZTEMPD(420)
  601 FORMAT(//40X,21HINFORMATION FROM TAPE,I2,8H CHANNEL,I2,
     *       //40X,6A10,//)
  602 FORMAT(30X,13HQUEUE DELAY =,I6,23H VEHICLE-SECONDS AFTER ,I2
     *       ,6H HOUR,,I3,9H MINUTES,,I3,8H SECONDS)
  603 FORMAT(//30X,25HEQUIVALENT QUEUE DELAY = ,F6.0,
     *        25H VEHICLE-SECONDS PER HOUR,/30X,I5,1X,
     *        34HMISSING DATA POINTS ENCOUNTERED  (,F5.2,9H PERCENT)/1H1)
  901 FORMAT(//5X,30HNO DATA : 9010 ERROR IN DELAYT)
      IT = ITT
      IC = ICC
C-----IT = TAPE NUMBER
C-----IC = CHANNEL NUMBER
      PRINT 601 , IT,IC,IDEN
      CALL  POSITON
                     IF ( IEOR . NE . 0 )          GO TO 9010
      IDELT = 0
      IKOUNT = IDIGIT(2,IC)*10 + IDIGIT(1,IC)
                     IF ( IKOUNT . GT . 50 )       IKOUNT = 0
      IKOUNTL = IKOUNT
 1010 CONTINUE
      IEOR = IOP     ( RB,IT,IWORD,1 )
                     IF ( IEOR . NE . 0 )          GO TO 2010
      CALL  UNPACK
                     IF ( ITHOLD-ITIME.LE.100 )    GO TO 1020
C-----ITMAX IS THE CUMULATIVE VALUE OF A BACKWARD SKIP ON THE CLOCK
C-----GREATER THAN 100 SECONDS
      ITMAX = ITMAX + ITHOLD - ITIME + 1
 1020 CONTINUE
                     IF ( ITIME-ITHOLD . LT . 60 )GO TO 1030
C-----JUMP IS THE CUMULATIVE VALUE OF A FORWARD SKIP ON THE CLOCK
C-----GREATER THAN 60 SECONDS
      JUMP = JUMP + ITIME - ITHOLD - 1
 1030 CONTINUE
      ITHOLD = ITIME
      ITIME = ITIME + ITMAX - JUMP
CS                   IF ( ITIME . GT . ITSTOP )    GO TO 2010              ALLDATA
                     IF ( ITIME . LE . ITIMEL )    GO TO 1010
                     IF ( ITIMEL . LT . INTVLC )   GO TO 1040
      CALL  DELTA    ( ITSTRT,INTVLC )
      PRINT 602 , IDELT,ITHOUR,ITMIN,ITSEC
      INTVLC = INTVLC + INTVL
 1040 CONTINUE
      IKOUNT = IDIGIT(2,IC)*10 + IDIGIT(1,IC)
               IF ( IABS(IKOUNT-IKOUNTL) . GT . 6 ) IKOUNT = IKOUNTL
      CKOUNT = IKOUNT
                     IF ( ITIME . NE . ITIMEL+1 )
     *               CKOUNT = 0.5*(CKOUNT+IKOUNTL)
                     IF ( ITIME . NE . ITIMEL+1 )
     *               MISS = MISS + ITIME - ITIMEL - 1
      IDELT = IDELT + CKOUNT*(ITIME-ITIMEL) + 0.5
      ITIMEL = ITIME
      IKOUNTL = IKOUNT
      GO TO 1010
 2010 CONTINUE
      CALL  DELTA    ( INITIAL,ITIME )
      TIME = ITIME - INITIAL
      DELAY = IDELT
      DELTEQ = DELAY*3600.0/TIME
      PMISS = 100.0*FLOAT(MISS)/TIME
      STAT(IT,IC,2) = DELAY
      STAT(IT,IC,4) = TIME
      PRINT 602 , IDELT,ITHOUR,ITMIN,ITSEC
      PRINT 603 , DELTEQ,MISS,PMISS
      RETURN
 9010 CONTINUE
      PRINT 901
      RETURN
      END                                                                  DELAYT
```

```
      SUBROUTINE DELAYA ( IV1,IV2,ID1,ID2 )
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,AVDELA,DELAY,IDELT,TDELTD,
     *                  TDELTV,VOLUME,ZTEMPD(420)
  601 FORMAT(//40X,19HCOMPUTED STATISTICS,//40X,6A10//40X,8H AVERAGE,
     *       27H QUEUE DELAY PER VEHICLE = ,F5.1,8H SECONDS,//
     *       40X,13H TIME PERIOD:,I3,6H HOUR,,I3,9H MINUTES,,I3,
     *       8H SECONDS///)
  901 FORMAT(//40X,33HNOT ENOUGH INFO FOR AVERAGE DELAY,//40X,6A10//40X
     *       ,10H VOLUME = ,G10.3,5X,9H DELAY = ,G10.3)
      VOLUME = STAT(IV1,IV2,1)
      DELAY  = STAT(ID1,ID2,2)
                    IF ( VOLUME . LE . 0.0 )      GO TO 9010
                    IF ( DELAY  . LE . 0.0 )      GO TO 9010
      TDELTV = STAT(IV1,IV2,3)
      TDELTD = STAT(ID1,ID2,4)
      AVDELA = DELAY*TDELTV/(VOLUME*TDELTD)
      IDELT  = TDELTV
                    IF ( TDELTV . GT . TDELTD )  IDELT = TDELTD
      CALL  DELTA   ( 0,IDELT )
      PRINT 601 , IDEN,AVDELA,ITHOUR,ITMIN,ITSEC
      RETURN
 9010 CONTINUE
      PRINT 901 , IDEN,VOLUME,DELAY
      RETURN
      END
```

DELAYA

```
      SUBROUTINE SIGNAL ( ITT,NUMM )
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,I,ISIG(12),NUM,ZTEMPD(412)
  601 FORMAT(//40X,21HINFORMATION FROM TAPE,I2,//40X,6A10//)
  602 FORMAT(63X,24HSIGNAL CONNECTOR LETTERS//62X,12R6)
  603 FORMAT(10X,I2,7H HOURS,,I3,9H MINUTES,,I3,
     *       29H SECONDS - SUMMATION (SECS) =,12I6)
  604 FORMAT(/40X,I5,36H MISSING DATA POINTS ENCOUNTERED   (,F5.2,
     *       9H PERCENT)/1H1)
  901 FORMAT(//5X,30HNO DATA : 9010 ERROR IN SIGNAL)
      IT = ITT
      NUM = NUMM
      PRINT 601 , IT,IDEN
      CALL  POSITON
                    IF ( IEOR . NE . 0 )          GO TO 9010
      PRINT 602 , (I,I=1,NUM)
      PRINT
      DO 1010  I = 1 , NUM
      ISIG(I) = 0
 1010 CONTINUE
 1020 CONTINUE
      ITIMEL = ITIME
      DO 1030  I = 1 , NUM
      IF ( (LSHIFT(IWORD,60-(I-1)).A.M01).NE.0 ) ISIG(I) = ISIG(I) + 1
 1030 CONTINUE
 1040 CONTINUE
      IEOR = IOP     ( RB,IT,IWORD,1 )
                    IF ( IEOR . NE . 0 )          GO TO 2010
      ITIME = LSHIFT(IWORD,60-16) . AND . M13
                    IF ( (ITHOLD-ITIME).LE.100 ) GO TO 1050
      ITMAX = ITMAX + ITHOLD - ITIME + 1
 1050 CONTINUE
                    IF ( (ITIME-ITHOLD).LT.60 )  GO TO 1060
      JUMP = JUMP + ITIME - ITHOLD - 1
 1060 CONTINUE
      ITHOLD = ITIME
      ITIME = ITIME + ITMAX - JUMP
                    IF ( ITIME . LE . ITIMEL )   GO TO 1040
CS                  IF ( ITIMEL . GT . ITSTOP )  GO TO 2010
                    IF ( ITIME . GT . ITIMEL+1 )
     *              MISS = MISS + ITIME - ITIMEL - 1
                    IF ( ITIMEL . LT . INTVLC )  GO TO 1020
      CALL  DELTA   ( ITSTRT,INTVLC )
      PRINT 603 , ITHOUR,ITMIN,ITSEC,(ISIG(I),I=1,NUM)
      INTVLC = INTVLC + INTVL
      GO TO 1020
 2010 CONTINUE
      CALL  DELTA   ( INITIAL,ITIMEL )
      TIME = ITIMEL - INITIAL
      PMISS = 100.0*FLOAT(MISS)/TIME
      PRINT 603 , ITHOUR,ITMIN,ITSEC,(ISIG(I),I=1,NUM)
      PRINT 604 , MISS,PMISS
      RETURN
 9010 CONTINUE
      PRINT 901
      RETURN
      END
```

SIGNAL

```
      SUBROUTINE HEADWA ( ITT,ICC,IDD )
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,IDIGTL,IDNUM1,IDNUM2,IDNUM3,
     *                  INAME,IRET,ITFIRST,NUMRED,NUMRIT,RHEAD,
     *                  ZTEMPD(416)
      DATA      TAPE66 / 6LTAPE66 /
  601 FORMAT(3I1)
  602 FORMAT(1HH,6I1)
  603 FORMAT(///20X,45HHEADWAYS WRITTEN ON TAPE 66 IN BINARY FORMAT ,
     *       /20X,50HTAPE 66 WILL BE FOUND IN THE LOCAL FILE TABLE AS& ,
     *       /20X,A7,5X,I5,11H WORDS READ,I5,16H HEADWAYS STORED,
     *       /20X,1H(,I4,1H,,I4,1H,,I4,1H))
  901 FORMAT(////5X,30HNO DATA : 9010 ERROR IN HEADWA)
      IT = ITT
      IC = ICC
      ID = IDD
      CALL  IOP     ( REW,IT )
      CALL  IOP     ( RB,IT,DISCRD,60 )
            DECODE  ( 3,601,IDENT )              IDNUM1,IDNUM2,IDNUM3
                    IF ( IDNUM1 . LE . 0 )       IDNUM1 = 0
                    IF ( IDNUM2 . LE . 0 )       IDNUM2 = 0
                    IF ( IDNUM3 . LE . 0 )       IDNUM3 = 0
            ENCODE  ( 7,602,INAME )
     *              IDNUM1,IDNUM2,IDNUM3,IT,IC,ID
      INAME = INAME . AND . 7L;;;;;;;
      CALL  CHGFILE ( TAPE66,INAME,IRET )
                    IF ( IRET . NE . 0 )         STOP 902
      CALL  IOP     ( REW,66 )
      CALL  IOP     ( WC,66,IDEN,6 )
      NUMRED = NUMRIT = 0
      IEOR = IOP    ( RB,IT,IWORD,1 )
                    IF ( IEOR . NE . 0 )         GO TO 9010
      CALL  UNPACK
      ITFIRST = ITIME
 1010 CONTINUE
      ITHOLD = ITIME
 1020 CONTINUE
      ITIMEL = ITIME
      IDIGTL = IDIGIT(ID,IC)
      IEOR = IOP    ( RB,IT,IWORD,1 )
                    IF ( IEOR . NE . 0 )         GO TO 2010
      NUMRED = NUMRED + 1
                    IF ( NUMRED . GT . MNUMW )   GO TO 2010
      CALL  UNPACK
                    IF ( ITIME . NE . ITIMEL+1 ) GO TO 1010
            IF ( IDIGIT(ID,IC) . EQ . IDIGTL )   GO TO 1020
      RHEAD = ITIME - ITHOLD
      NUMRIT = NUMRIT + 1
      CALL  IOP     ( WB,66,RHEAD,1 )
      GO TO 1010
 2010 CONTINUE
      PRINT 603 , INAME,NUMRED,NUMRIT,ITIME,ITHOLD,ITFIRST
      CALL  IOP     ( WF,66 )
      RETURN
 9010 CONTINUE
      PRINT 901
      RETURN
      END
```

HEADWA

```
      SUBROUTINE POSITON
      COMMON / IOPLIT / IEOR,ODD,RB,RC,RET,REW,RR,WB,WC,WF
      COMMON / STAT   / IDEN(6),IDENT(6),INTVL,ITSTOP,ITSTRT,
     *                  MNUMW,STAT(5,4,4)
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,TIME,ZTEMPD(426)
      CALL  IOP     ( REW,IT )
      CALL  IOP     ( RB,IT,DISCRD,60 )
 1010 CONTINUE
      IEOR = IOP    ( RB,IT,IWORD,1 )
                    IF ( IEOR . NE . 0 )         RETURN
      ITIME = LSHIFT(IWORD,60-16) . AND . M13
            IF ( IABS(ITIME-ITSTRT) . GT . 200 )ITIME = ITSTRT
                    IF ( ITIME . LT . ITSTRT )   GO TO 1010
      CALL  UNPACK
      ITMAX = 0
      JUMP = 0
      MISS = 0
      ITIMEL = ITIME
      ITHOLD = ITIME
      INITIAL = ITIME
      INTVLC = INTVL + INITIAL
      RETURN
      END
```

POSITON

```
      SUBROUTINE UNPACK
      COMMON / UNPACK / IDIGIT(2,2),ITIME,ITYPE(2),IWORD,M01,M03,M04,
     *                  M12,M13
      EQUIVALENCE       (IDIG11,IDIGIT(1,1)),(IDIG12,IDIGIT(1,2)),
     *                  (IDIG21,IDIGIT(2,1)),(IDIG22,IDIGIT(2,2)),
     *                  (ITYPE1,ITYPE(1)),(ITYPE2,ITYPE(2))
C
      INVERT(I) = M04 . AND . .NOT. I
C
      IDIG11 = IWORD . AND . M04
                IF ( ITYPE1 . NE . 0 )       IDIG11 = INVERT(IDIG11)
      IDIG21 = LSHIFT(IWORD,60-4) . AND . M04
                IF ( ITYPE1 . NE . 0 )       IDIG21 = INVERT(IDIG21)
      IDIG12 = LSHIFT(IWORD,60-8) . AND . M04
                IF ( ITYPE2 . NE . 0 )       IDIG12 = INVERT(IDIG12)
      IDIG22 = LSHIFT(IWORD,60-12) . AND . M04
                IF ( ITYPE2 . NE . 0 )       IDIG22 = INVERT(IDIG22)
      ITIME  = LSHIFT(IWORD,60-16) . AND . M13
C     IUNIT  = LSHIFT(IWORD,60-29) . AND . M03
C     ICTAPE = LSHIFT(IWORD,60-36) . AND . M12
      RETURN
      END                                                   UNPACK
```

```
C     SUBROUTINE REPACK
C     DATA MTIME / 77777777777400001777777B /
C               IF ( ITIME . GT . 8191 )       STOP 903
C     IWORD = IWORD . AND . MTIME
C     RETURN
C     END
      SUBROUTINE DELTA ( IBEGIN,IEND )
      COMMON / ZTEMPD / IBAD(5),DISCRD(60),IC,ID,INITIAL,INTVLC,IT,
     *                  ITHOLD,ITHOUR,ITIMEL,ITMAX,ITMIN,ITSEC,JUMP,
     *                  MISS,PMISS,NTERVL,ZTEMPD(426)
      NTERVL = IEND - IBEGIN
      ITHOUR = NTERVL/3600
      ITMIN = NTERVL/60 - ITHOUR*60
      ITSEC = NTERVL - ITHOUR*3600 - ITMIN*60
      RETURN
      END                                                   DELTA
```

EE M

```
          IDENT    CHGFILE
          ENTRY    CHGFILE
          TITLE    CHGFILE  ( INTFILE,EXTFILE,IRET )
          SPACE    2
*
*****     CHANGE EXTERNAL FILE NAME FOR SPECIFIED INTERNAL FILE NAME
*
*****     INTFILE  = INTERNAL FILE NAME ADDRESS
*                  = ØLFILENAME
*         EXTFILE  = EXTERNAL FILE NAME ADDRESS
*                  = ØLFILENAME
*         IRET     = RETURN FLAG ADDRESS
*                  = Ø = OK
*                  = 1 = IFILE NAME NOT FOUND
*
          SPACE    2
          VFD      42/7LCHGFILE,18/3
CHGFILE   BSS      1                  ENTRY/EXIT LINE
          MXØ      42                 MAKE NAME MASK
          SA1      B1                 SET X1 = INTFILE NAME
          SB1      1                  SET B1 = 1
          SB6      6Ø                 SET B6 = 6Ø
          SX6      B1                 SET INTFILE NAME NOT FOUND FLAG
          SA2      B1                 INITIALIZE A2 = 1
LOOPLC    SA2      A2+B1              GET LOW CORE FILE NAME AND FET AD
          SB5      A2                 CHECK FOR END OF LOW CORE POINTER
          GT       B5,B6,NOFILE       GO TO NOFILE IF A2 GT 6Ø
          ZR       X2,NOFILE          GO TO NOFILE IF LOW CORE WORD = Ø
          BX3      X2*XØ              MASK OUT LOW CORE FILE NAME
          BX4      X3-X1              CHECK INTFILE NAME = LC FILE NAME
          NZ       X4,LOOPLC          GO TO LOOPLC IF NAMES NOT EQUAL
          SX6      BØ                 SET IRET OK
NOFILE    SA6      B3                 STORE IRET
          NZ       X6,CHGFILE         RETURN IF IFILE NAME NOT FOUND
          SA3      X2                 GET FIRST WORD OF INTFILE FET
          SX4      X3                 SET X4 = LOWER 18 BITS OF X3
          ZR       X4,QUIET           GO TO QUIET IF INTFILE NOT USED
          LX4      59                 PUT BIT 1 IN SIGN BIT
          NG       X4,QUIET           GO TO QUIET IF X4 ODD
          SYSTEM   RCL,R,A3           WAIT FOR INTFILE FET QUIET
QUIET     SA1      B2                 GET EXTFILE NAME
          BX7      X1                 TRANSMIT EXTFILE TO X7
          SA7      A3                 STORE EXTFILE IN INTFILE FET
          SYSTEM   OPE,R,A7           OPEN EXTFILE
          EQ       CHGFILE
          END
```

```
C¬    PROGRAM    DISFIT ( INPUT=513,OUTPUT=513,TAPE5=INPUT,
C¬   *                    TAPE6=OUTPUT )
C
C-----DISTRIBUTION FITTING PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION
C-----PACKAGE
C
C-----C¬ = CDC ONLY CODE
C-----C; = IBM ONLY CODE
      COMMON / DISVAL / DØPØ,PE(5Ø,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  DØPØ,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  SUMX,SUMXX,DHEAD
      DIMENSION         IDWN(2),IUPP(2)
      DATA      IDWN    / 4H DOW,4HN) - /
      DATA      IUPP    / 4H UP),4H --- /
      DATA      HMAX    /-1,ØE+99 /
      DATA      HMIN    / 1,ØE+99 /
      DATA      SUMX    / Ø,ØD+ØØ /
      DATA      SUMXX   / Ø,ØD+ØØ /
      DATA      TIME    / Ø,Ø /
  5Ø1 FORMAT(15A4)
  5Ø2 FORMAT(F8,3)
  6Ø1 FORMAT(1H1,39X,15A4)
  6Ø2 FORMAT(/4ØX,38HNUMBER OF HEADWAYS READ --------------,I6    ,//,
     *          4ØX,38HTIME (HR) ---------------------------,F12,5,//,
     *          4ØX,38HVOLUME (VEH/HR) ---------------------,F12,5,//,
     *          4ØX,38HMINIMUM HEADWAY (SEC) ---------------,F12,5,//,
     *          4ØX,38HMAXIMUM HEADWAY (SEC) ---------------,F12,5,//,
     *          4ØX,38HRANGE (SEC) -------------------------,F12,5,//,
     *          4ØX,38HMEAN (SEC/VEH) ----------------------,F12,5,//,
     *          4ØX,38HVARIANCE (SQ, SEC) ------------------,F12,5,//,
     *          4ØX,38HSTANDARD DEVIATION (SEC) ------------,F12,5,//)
  6Ø3 FORMAT(/,1X,8HUNIFORM ,22(1H-),8H S, D, =,F6,2,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  6Ø4 FORMAT(/,1X,11HLOG NORMAL ,19(1H-),8H S, D, =,F6,2,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  6Ø5 FORMAT(/,1X,21HNEGATIVE EXPONENTIAL ,9(1H-),14H NO PARAMETER ,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  6Ø6 FORMAT(/,1X,38HSHIFTED NEGATIVE EXPONENTIAL - TAU   =,F6,2,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  6Ø7 FORMAT(/,1X,6HGAMMA ,24(1H-),8H A      =,F6,2,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  6Ø8 FORMAT(/,1X,15HERLANG (ROUNDED,2A4,7(1H-),1X,7HK      =,I3,3X,
     *          12H    CHI SQ =,F7,2,6H  DF =,I3,9H  ALPHA =,F8,4,
     *          8H  CONF =,F8,4,18H    MAX CUM DIFF =,F7,5)
  9Ø1 FORMAT(//,48HØNO HEADWAYS READ BEFORE END-OF-FILE ENCOUNTERED)
  9Ø2 FORMAT(1X,3F15,8,I1Ø)
C¬    ASSIGN 1Ø1 TO NRECAD
C¬    CALL  XMIT    ( NRECAD )
      DØPØ = Ø,ØD+ØØ
      PI = 4,ØD+ØØ*DATAN(1,ØD+ØØ)
      READ (5,5Ø1) ITITLE
      N = Ø
      DO 1Ø1Ø  IZ = 1 , 7
      CHICHI(IZ) = -1,Ø
 1Ø1Ø CONTINUE
      DO 1Ø2Ø  IZ = 1 , 51
      PO(IZ) = DØPØ
 1Ø2Ø CONTINUE
 1Ø3Ø CONTINUE
      READ (5,5Ø2,END=2Ø1Ø) HEAD
      DHEAD = DBLE(HEAD)
      HMIN = AMIN1(HMIN,HEAD)
      HMAX = AMAX1(HMAX,HEAD)
      N = N + 1
      SUMX = SUMX + DHEAD
      SUMXX = SUMXX + DHEAD**2
      IGROUP = AMINØ(IFIX(HEAD+1,Ø),51)
      PO(IGROUP) = PO(IGROUP) + 1,ØD+ØØ
      GO TO 1Ø3Ø
 2Ø1Ø CONTINUE
                      IF ( N , LE , Ø )             GO TO 9Ø1Ø
      TIME = SUMX/36ØØ,ØD+ØØ
      VOLUME = N/TIME
      DO 2Ø2Ø  I = 1 , 51
      PO(I) = PO(I)/N
 2Ø2Ø CONTINUE
      RANGE = HMAX - HMIN
      XMEAN = SUMX/N
      VAR = (SUMXX - N*(XMEAN**2))/(N-1,ØØD+ØØ)
      SD = DSQRT(VAR)
      PRINT 6Ø1 , ITITLE
      SXMEAN = XMEAN
      SVAR = VAR
      SSD = SD
      PRINT 6Ø2 , N,TIME,VOLUME,HMIN,HMAX,RANGE,SXMEAN,SVAR,SSD
      MP3SD = XMEAN + 3,ØD+ØØ*SD + Ø,5D+ØØ
                      IF ( XMEAN , LE , Ø,ØD+ØØ )  GO TO 9Ø2Ø
      CALL  UNIFRM
                      IF ( CHICHI(1) , LT , Ø,Ø )  GO TO 3Ø1Ø
      PRINT 6Ø3 , PARAM(1),CHICHI(1),NDF(1),ALPHA(1),CONFL(1),XKSMCD(1)
 3Ø1Ø CONTINUE
      CALL  LOGNRM
                      IF ( CHICHI(2) , LT , Ø,Ø )  GO TO 3Ø2Ø
      PRINT 6Ø4 , PARAM(2),CHICHI(2),NDF(2),ALPHA(2),CONFL(2),XKSMCD(2)
 3Ø2Ø CONTINUE
      CALL  NEGEXP
                      IF ( CHICHI(3) , LT , Ø,Ø )  GO TO 3Ø3Ø
      PRINT 6Ø5 , CHICHI(3),NDF(3),ALPHA(3),CONFL(3),XKSMCD(3)
 3Ø3Ø CONTINUE
      CALL  SNEGEX
                      IF ( CHICHI(4) , LT , Ø,Ø )  GO TO 3Ø4Ø
      PRINT 6Ø6 , PARAM(4),CHICHI(4),NDF(4),ALPHA(4),CONFL(4),XKSMCD(4)
 3Ø4Ø CONTINUE
      CALL  GAMMA
                      IF ( CHICHI(5) , LT , Ø,Ø )  GO TO 3Ø5Ø
      PRINT 6Ø7 , PARAM(5),CHICHI(5),NDF(5),ALPHA(5),CONFL(5),XKSMCD(5)
 3Ø5Ø CONTINUE
      CALL  ERLANG    ( 6,DØPØ )
                      IF ( CHICHI(6) , LT , Ø,Ø )  GO TO 3Ø6Ø
      K = PARAM(6)
      PRINT 6Ø8 , IDWN,K,CHICHI(6),NDF(6),ALPHA(6),CONFL(6),XKSMCD(6)
 3Ø6Ø CONTINUE
      CALL  ERLANG    ( 7,1,ØD+ØØ )
                      IF ( CHICHI(7) , LT , Ø,Ø )  GO TO 3Ø7Ø
      K = PARAM(7)
      PRINT 6Ø8 , IUPP,K,CHICHI(7),NDF(7),ALPHA(7),CONFL(7),XKSMCD(7)
 3Ø7Ø CONTINUE
                      IF ( CHICHI(1) , LT , Ø,Ø )  GO TO 4Ø1Ø
      PRINT 6Ø1 , ITITLE
      PRINT 6Ø3 , PARAM(1),CHICHI(1),NDF(1),ALPHA(1),CONFL(1),XKSMCD(1)
      CALL  PAGPLT    ( 1 )
 4Ø1Ø CONTINUE
                      IF ( CHICHI(2) , LT , Ø,Ø )  GO TO 4Ø2Ø
      PRINT 6Ø1 , ITITLE
      PRINT 6Ø4 , PARAM(2),CHICHI(2),NDF(2),ALPHA(2),CONFL(2),XKSMCD(2)
      CALL  PAGPLT    ( 2 )
 4Ø2Ø CONTINUE
                      IF ( CHICHI(3) , LT , Ø,Ø )  GO TO 4Ø3Ø
      PRINT 6Ø1 , ITITLE
      PRINT 6Ø5 , CHICHI(3),NDF(3),ALPHA(3),CONFL(3),XKSMCD(3)
      CALL  PAGPLT    ( 3 )
 4Ø3Ø CONTINUE
                      IF ( CHICHI(4) , LT , Ø,Ø )  GO TO 4Ø4Ø
      PRINT 6Ø1 , ITITLE
      PRINT 6Ø6 , PARAM(4),CHICHI(4),NDF(4),ALPHA(4),CONFL(4),XKSMCD(4)
```

```
      CALL  PAGPLT  ( 4 )
 4040 CONTINUE
                    IF ( CHICHI(5) . LT . 0.0 )  GO TO 4050
      PRINT 601 , ITITLE
      PRINT 607 , PARAM(5),CHICHI(5),NDF(5),ALPHA(5),CONFL(5),XKSMCD(5)
      CALL  PAGPLT  ( 5 )
 4050 CONTINUE
                    IF ( CHICHI(6) . LT . 0.0 )  GO TO 4060
      K = PARAM(6)
      PRINT 601 , ITITLE
      PRINT 608 , IDWN,K,CHICHI(6),NDF(6),ALPHA(6),CONFL(6),XKSMCD(6)
      CALL  PAGPLT  ( 6 )
 4060 CONTINUE
                    IF ( CHICHI(7) . LT . 0.0 )  GO TO 4070
      K = PARAM(7)
      PRINT 601 , ITITLE
      PRINT 608 , IUPP,K,CHICHI(7),NDF(7),ALPHA(7),CONFL(7),XKSMCD(7)
      CALL  PAGPLT  ( 7 )
 4070 CONTINUE
C-    ENDFILE 6
      CALL  EXIT
 9010 CONTINUE
      PRINT 901
      STOP  901
 9020 CONTINUE
      SXMEAN = XMEAN
      PRINT 902 , XMEAN
      STOP  902
C-101 CONTINUE
C-    CALL  XMIT    ( 0 )
C-    DO 102  I = 1 , 51
C-    SFO = N*PO(I)
C-    SPO = PO(I)
C-    PRINT 902 , SFO , SPO
C-102 CONTINUE
C-    DO 103  J = 1 , 7
C-    PRINT 902 , XKSMCD(J),PARAM(J),CHICHI(J),NDF(J)
C-103 CONTINUE
C-    STOP 903
C-104 GO TO NRECAD                                                      *DEBUG*
      END                                                               DISFIT
```

```
      SUBROUTINE CHISUM ( NDIST )
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  FO,FE,FOL,FEL,CHISQD,CSUMFO,CSUMFE,DIFMAX
      CHISQD = D0P0
      NDF(NDIST) = -1
      FO = D0P0
      FE = D0P0
      FOL = D0P0
      FEL = D0P0
      CSUMFO = D0P0
      CSUMFE = D0P0
      DIFMAX = D0P0
      DO 1010  I = 1 , 50
      FO = FO + N*PO(I)
      FE = FE + N*PE(I,NDIST)
      CSUMFO = CSUMFO + PO(I)
      CSUMFE = CSUMFE + PE(I,NDIST)
      DIFMAX = DMAX1(DIFMAX,DABS(CSUMFO-CSUMFE))
                    IF ( FE . LT . 5.0D+00 )     GO TO 1010
      FOL = FO
      FEL = FE
      CHISQD = CHISQD + (FO-FE)**2/FE
      NDF(NDIST) = NDF(NDIST) + 1
      FO = D0P0
      FE = D0P0
 1010 CONTINUE
      FO = FO + FOL
      FE = FE + FEL
      CHICHI(NDIST) = CHISQD - (FOL-FEL)**2/FEL + (FO-FE)**2/FE
      XKSMCD(NDIST) = DIFMAX
      RETURN
      END                                                               CHISUM
```

```
      SUBROUTINE CHIVAL ( NDIST )
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  DFD2,CONST,X,XSTEP,SUM,F,FLAST,GAMMAF,POWER
C
C-----EQUATIONS FROM *STATISTICAL PRINCIPLES IN EXPERIMENTAL DESIGN*
C-----BY B. J. WINER, PAGE 825.
C
      SUM = 1.0D+00
            IF ( CHICHI(NDIST) . LE . 0.0 )       GO TO 1020
                    IF ( NDF(NDIST) . LE . 0 )   GO TO 1020
      DFD2 = NDF(NDIST)/2.0D+00
      CONST = (0.5D+00**DFD2)/GAMMAF(DFD2)
      XSTEP = NDF(NDIST)/100.0D+00
      NSTEP = AMAX0(10,IDINT(DBLE(CHICHI(NDIST))/XSTEP+0.5D+00))
      XSTEP = CHICHI(NDIST)/NSTEP
      POWER = DFD2 - 1.0D+00
      SUM = D0P0
      X = D0P0
      FLAST = D0P0
      DO 1010  I = 1 , NSTEP
      X = X + XSTEP
      F = CONST*DEXP(-0.5D+00*X)*(X**POWER)
      SUM = SUM + XSTEP*0.5D+00*(F+FLAST)
      FLAST = F
 1010 CONTINUE
 1020 CONTINUE
      ALPHA(NDIST) = SUM
      CONFL(NDIST) = 100.0D+00*(1.0D+00-SUM)
      RETURN
      END                                                                CHIVAL
```

```
      SUBROUTINE UNIFRM
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  SQRT3,ALAST,AREA,CONST,T,A,B
      DATA     SQRT3  / 1.73205080757D+00 /
  904 FORMAT(/,4X,7HUNIFORM,6X,7HT MIN =,F6.2,
     *       48HDISTRIBUTION NOT POSSIBLE - MINIMUM VALUE LT 0.0)
      A = XMEAN - SD*SQRT3
                  IF( A . LT . D0P0 )       GO TO 9040
      B = XMEAN + SD*SQRT3
      PARAM(1) = SD
      ALAST = D0P0
      AREA  = D0P0
      CONST = 1.0D+00/(B-A)
      T = D0P0
      DO 1010  I = 1 , 50
      T = T + 1.0D+00
                  IF ( T . GT . A )          AREA = CONST*(T-A)
                  IF ( T . GT . B )          AREA = 1.0D+00
      PE(I,1) = AREA - ALAST
      ALAST = AREA
 1010 CONTINUE
      CALL  CHISUM  ( 1 )
      NDF(1) = NDF(1) - 2
      CALL  CHIVAL  ( 1 )
      RETURN
 9040 CONTINUE
      SA = A
      PRINT 904 , SA
      RETURN
      END                                                                UNIFRM
```

```
      SUBROUTINE LOGNRM
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  XEXP,ALAST,AREA,CONST,DT,F,FLAST,T,YMEAN,YVAR
  905 FORMAT(/,4X,10HLOG NORMAL,21X,7HEXPNT =,F6.2,5X,
C¬   *        45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 741.0)
C;   *        45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 150.0)
C
C-----EQUATIONS FROM *PROBABILITY AND STATISTICS FOR ENGINEERS* BY
C-----IRWIN MILLER AND JOHN E. FREUND, P.76-78.
C
      YMEAN = DLOG(XMEAN) - 0.5D+00*DLOG((VAR/(XMEAN**2))+1.0D+00)
      YVAR = DLOG((VAR/(XMEAN**2))+1.0D+00)
      PARAM(2) = SD
      CONST = 1.0D+00/DSQRT(2.0D+00*PI*YVAR)
      AREA = D0P0
      ALAST = D0P0
      FLAST = D0P0
      T = D0P0
      DT = 0.01D+00
      DO 1020  I = 1 , 50
                     IF ( I . GT . MP3SD )         DT = 0.1D+00
      NUM = 1.0D+00/DT + 0.5D+00
      DO 1010  J = 1 , NUM
      T = T + DT
      XEXP = -0.5D+00*((DLOG(T)-YMEAN)**2)/YVAR
C¬                   IF( DABS(XEXP).GT.741.0D+00 )GO TO 9050
C;                   IF( DABS(XEXP).GT.150.0D+00 )GO TO 9050
      F = CONST*DEXP(XEXP)/T
      AREA = AREA + 0.5D+00*DT*(FLAST+F)
      FLAST = F
 1010 CONTINUE
      PE(I,2) = AREA - ALAST
      ALAST = AREA
 1020 CONTINUE
      CALL  CHISUM   ( 2 )
      NDF(2) = NDF(2) - 2
      CALL  CHIVAL   ( 2 )
      RETURN
 9050 CONTINUE
      SXEXP = XEXP
      PRINT 905 , SXEXP
      RETURN
      END
```

LOGNRM

```
      SUBROUTINE NEGEXP
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  T,ALAST,AREA,XEXP
  906 FORMAT(/,4X,20HNEGATIVE EXPONENTIAL,11X,7HEXPNT =,F6.2,5X,
C¬   *        45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 741.0)
C;   *        45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 150.0)
      PARAM(3) = D0P0
      T = D0P0
      ALAST = D0P0
      DO 1010  I = 1 , 50
      T = T + 1.0D+00
      XEXP = -T/XMEAN
C¬                   IF( DABS(XEXP).GT.741.0D+00 )GO TO 9060
C;                   IF( DABS(XEXP).GT.150.0D+00 )GO TO 9060
      AREA = 1.0D+00 - DEXP(XEXP)
      PE(I,3) = AREA - ALAST
      ALAST = AREA
 1010 CONTINUE
      CALL  CHISUM   ( 3 )
      NDF(3) = NDF(3) - 1
      CALL  CHIVAL   ( 3 )
      RETURN
 9060 CONTINUE
      SXEXP = XEXP
      PRINT 906 , SXEXP
      RETURN
      END
```

NEGEXP

```
      SUBROUTINE SNEGEX
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  T,ALAST,AREA,XEXP,TAU,CONST
  907 FORMAT(/,4X,38HSHIFTED NEGATIVE EXPONENTIAL   TAU   =,F6.2,5X,
     *       38HDISTRIBUTION NOT POSSIBLE - TAU LT 0.0)
  908 FORMAT(/,4X,38HSHIFTED NEGATIVE EXPONENTIAL   MEAN  =,F6.2,5X,
     *       39HDISTRIBUTION NOT POSSIBLE - MEAN LE 0.0)
  909 FORMAT(/,4X,38HSHIFTED NEGATIVE EXPONENTIAL   EXPNT =,F6.2,5X,
C¬   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 741.0)
C;   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 150.0)
      TAU = XMEAN - SD
      PARAM(4) = TAU
                 IF ( TAU . LT . D0P0 )        GO TO 9070
                 IF ( XMEAN . LE . D0P0 )      GO TO 9080
      CONST = 1.0D+00/(XMEAN-TAU)
      T = D0P0
      ALAST = D0P0
      AREA = D0P0
      DO 1020  I = 1 , 50
      T = T + 1.0D+00
                 IF ( T . LE . TAU )          GO TO 1010
      XEXP = -CONST*(T-TAU)
C¬               IF( DABS(XEXP).GT.741.0D+00 )GO TO 9090
C;               IF( DABS(XEXP).GT.150.0D+00 )GO TO 9090
      AREA = 1.0D+00 - DEXP(XEXP)
 1010 CONTINUE
      PE(I,4) = AREA - ALAST
      ALAST = AREA
 1020 CONTINUE
      CALL  CHISUM  ( 4 )
      NDF(4) = NDF(4) - 2
      CALL  CHIVAL  ( 4 )
      RETURN
 9070 CONTINUE
      PRINT 907 , PARAM(4)
      RETURN
 9080 CONTINUE
      SXMEAN = XMEAN
      PRINT 908 , SXMEAN
      RETURN
 9090 CONTINUE
      SXEXP = XEXP
      PRINT 909 , SXEXP
      RETURN
      END                                                              SNEGEX
```

```
      SUBROUTINE GAMMA
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  A,ALAST,ALPHAG,AREA,CONST,DT,F,FLAST,GAMMAF,T,
     *                  XEXP
  910 FORMAT(/,4X,5HGAMMA,26X,7HA     =,F6.2,5X,
     *         38HDISTRIBUTION NOT POSSIBLE - A GT 150.0)
  911 FORMAT(/,4X,5HGAMMA,26X,7HEXPNT =,F6.2,5X,
C¬   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 741.0)
C;   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 150.0)
      ALPHAG = XMEAN/VAR
      A = XMEAN**2/VAR
      PARAM(5) = A
                 IF ( A . GT . 150.0D+00 ) GO TO 9100
      AREA = D0P0
      ALAST = D0P0
      FLAST = D0P0
      T = D0P0
      CONST = ALPHAG/GAMMAF(A)
      DT = 0.01D+00
      DO 1020  I = 1 , 50
                 IF ( I . GT . MP3SD )        DT = 0.1D+00
      NUM = 1.0D+00/DT + 0.5D+00
      DO 1010  J = 1 , NUM
      T = T + DT
      XEXP = -ALPHAG*T
C¬               IF( DABS(XEXP).GT.741.0D+00 )GO TO 9110
C;               IF( DABS(XEXP).GT.150.0D+00 )GO TO 9110
      F = CONST*((ALPHAG*T)**(A-1.0D+00))*DEXP(XEXP)
      AREA = AREA + 0.5D+00*DT*(FLAST+F)
      FLAST = F
 1010 CONTINUE
      PE(I,5) = AREA - ALAST
      ALAST = AREA
 1020 CONTINUE
      CALL  CHISUM  ( 5 )
      NDF(5) = NDF(5) - 2
      CALL  CHIVAL  ( 5 )
      RETURN
 9100 CONTINUE
      PRINT 910 , PARAM(5)
      RETURN
 9110 CONTINUE
      SXEXP = XEXP
      PRINT 911 , SXEXP
      RETURN
      END                                                              GAMMA
```

```
      SUBROUTINE ERLANG ( NDIST,XROUND )
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DOUBLE PRECISION  ALPHAE,AREA,ALAST,DT,F,FACT,FLAST,CONST,T,XEXP,
     *                  XROUND
  912 FORMAT(/,4X,6HERLANG,25X,7HK      =,I6,5X,
     *       48HDISTRIBUTION NOT POSSIBLE - K LT 1.0 OR GT 150.0)
  913 FORMAT(/,4X,6HERLANG,25X,7HEXPNT =,F6.2,5X,
C=   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 741.0)
C;   *       45HDISTRIBUTION NOT POSSIBLE - EXPONENT GT 150.0)
  914 FORMAT(1H+,10X,14H(ROUNDED DOWN))
  915 FORMAT(1H+,10X,12H(ROUNDED UP))
      ALPHAE = XMEAN/VAR
      K = XMEAN**2/VAR + XROUND
      PARAM(NDIST) = K
                    IF ( K . LT . 1 )              GO TO 9120
                    IF ( K . GT . 150 )            GO TO 9120
      AREA = D0P0
      ALAST = D0P0
      T = D0P0
      KM1 = K - 1
      IFACT = 1
      DO 1010  I = 1 , KM1
      IFACT = IFACT*I
 1010 CONTINUE
      CONST = ALPHAE/IFACT
      FLAST = D0P0
                    IF ( K . EQ . 1 )              FLAST = CONST
      DT = 0.01D+00
      DO 1030  I = 1 , 50
                    IF ( I . GT . MP3SD )          DT = 0.1D+00
      NUM = 1.0D+00/DT + 0.5D+00
      DO 1020  J = 1 , NUM
      T = T + DT
      XEXP = -ALPHAE*T
C=                  IF( DABS(XEXP).GT.741.0D+00 )GO TO 9130
C;                  IF( DABS(XEXP).GT.150.0D+00 )GO TO 9130
      F = CONST*((ALPHAE*T)**KM1)*DEXP(XEXP)
      AREA = AREA + 0.5D+00*DT*(FLAST+F)
      FLAST = F
 1020 CONTINUE
      PE(I,NDIST) = AREA - ALAST
      ALAST = AREA
 1030 CONTINUE
      CALL  CHISUM  ( NDIST )
      NDF(NDIST) = NDF(NDIST) - 2
      CALL  CHIVAL  ( NDIST )
      RETURN
 9120 CONTINUE
      PRINT 912 , K
      GO TO 9140
 9130 CONTINUE
      SXEXP = XEXP
      PRINT 913 , SXEXP
 9140 CONTINUE
                    IF ( XROUND . LE . D0P0 )     PRINT 914
                    IF ( XROUND . GT . D0P0 )     PRINT 915
      RETURN
      END
```

ERLANG

```
      SUBROUTINE PAGPLT ( I )
      COMMON / DISVAL / D0P0,PE(50,7),PI,PO(51),SD,VAR,XMEAN,
     *                  ALPHA(7),CHICHI(7),CONFL(7),PARAM(7),XKSMCD(7),
     *                  ITITLE(15),MP3SD,N,NDF(7)
      DOUBLE PRECISION  D0P0,PE,PI,PO,SD,VAR,XMEAN
      DIMENSION         LINE(120)
      DATA      NCLOSE / 1H) /
      DATA      NEQUAL / 1H= /
      DATA      NSTAR  / 1H* /
      DATA      NPLUS  / 1H+ /
      DATA      NBLANK / 1H  /
  601 FORMAT(/,66X,7HPERCENT,//,3X,4HTIME,40X,
     *       10(4H   1),10(4H   2),4H   3,/,3X,5H(SEC),3X,
     *       3(40H   1   2   3   4   5   6   7   8   9   0),/,
     *       10X,122(1H.))
  602 FORMAT(1X,I2,3H - ,I2,3H  .,120A1,1H.)
  603 FORMAT(1X,10HT GT 50  .,120A1,1H.,/,10X,122(1H.))
      PRINT 601
      DO 1050  J = 1 , 51
      NB = IDINT(400.0D+00*PO(J)) + 1
      ICOL = MIN0(120,NB)
      DO 1010  K = 1 , ICOL
      LINE(K) = NPLUS
 1010 CONTINUE
                    IF ( NB . GT . 120 )          LINE(120) = NCLOSE
            IF ( DABS(PO(J)).LE.0.000001D+00 )    LINE(1) = NBLANK
      K = NB + 1
                    IF ( K . GT . 100 )           GO TO 1030
      DO 1020  L = K , 120
      LINE(L) = NBLANK
 1020 CONTINUE
 1030 CONTINUE
                    IF ( J . EQ . 51 )            GO TO 1050
      NB = IDINT(400.0D+00*PE(J,I)) + 1
      ICOL = MIN0(120,NB)
                    IF ( NB . LT . 1 )            GO TO 1040
      NCHAR = NSTAR
                    IF ( NB . LE . NB )           NCHAR = NEQUAL
                    IF ( NB . GT . 120 )          NCHAR = NCLOSE
      IF ( NCHAR.EQ.NEQUAL . AND . NB.EQ.1 )      NCHAR = NBLANK
      LINE(ICOL) = NCHAR
 1040 CONTINUE
      JM1 = J - 1
      PRINT 602 , JM1,J,LINE
 1050 CONTINUE
      PRINT 603 , LINE
      RETURN
      END
```

PAGPLT

```
      DOUBLE PRECISION
     *FUNCTION   GAMMAF ( X )
C
C=====ALGORITHM 221 FROM COLLECTED ALGORITHMS FROM CACM
C=====BY WALTER GAUTSCHI 10 AUG 63
C
C=====ADAPTED FROM CHEBYSHEV APPROXIMATIONS TO THE GAMMA FUNCTION
C=====BY HELMUT WERNER AND ROBERT COLLINGE
C=====MATHEMATICS OF COMPUTATION, VOL 15, 1965, PG 195=197
C
C=====COEFFICIENTS FOR MAXIMUM ERROR OF 0,96E=14
C
      DOUBLE PRECISION  Z,X,T,P,
     *                  A00,A01,A02,A03,A04,A05,A06,A07,A08,A09,
     *                  A10,A11,A12,A13
      DATA      A00    /  0,99999999999999044D+00 /
      DATA      A01    /  0,42278433510233479D+00 /
      DATA      A02    /  0,41184033016678129D+00 /
      DATA      A03    /  0,08157692612415546D+00 /
      DATA      A04    /  0,07424891541944474D+00 /
      DATA      A05    / =0,00026618659495306D+00 /
      DATA      A06    /  0,01114971433577893D+00 /
      DATA      A07    / =0,00283646252037282D+00 /
      DATA      A08    /  0,00206109185022554D+00 /
      DATA      A09    / =0,00083756468513517D+00 /
      DATA      A10    /  0,00037536505226307D+00 /
      DATA      A11    / =0,00012141734870632D+00 /
      DATA      A12    /  0,00002798328899383D+00 /
      DATA      A13    / =0,00000303019081028D+00 /
  916 FORMAT(25H BAD ARGUMENT FOR GAMMAF(,F10,3,1H))
      Z = X
                    IF ( Z , LE ,   0,0D+00 )     GO TO 9160
                    IF ( Z , GT , 150,0D+00 )     GO TO 9160
      GAMMAF = 1,0D+00
                    IF ( Z , EQ , 1,0D+00 )       RETURN
                    IF ( Z , EQ , 2,0D+00 )       RETURN
                    IF ( Z , GT , 3,0D+00 )       GO TO 1020
                    IF ( Z , GT , 2,0D+00 )       GO TO 1030
 1010 CONTINUE
      GAMMAF = GAMMAF/Z
      Z = Z + 1,0D+00
                    IF ( Z , LT , 2,0D+00 )       GO TO 1010
      GO TO 1030
 1020 CONTINUE
      Z = Z = 1,0D+00
      GAMMAF = GAMMAF*Z
                    IF ( Z , GT , 3,0D+00 )       GO TO 1020
 1030 CONTINUE
      T = Z = 2,0D+00
      P = ((((((A13*T+A12)*T+A11)*T+A10)*T+A09)*T+A08)*T+A07)*T+A06
      P = (((((P*T+A05)*T+A04)*T+A03)*T+A02)*T+A01)*T+A00
      GAMMAF = GAMMAF*P
      RETURN
 9160 CONTINUE
      SX = Z
      PRINT 916 , SX
      STOP 916
      END                                                       GAMMAF
```

PROGRAMMERS DOCUMENTATION

DISTRIBUTION FITTING PROCESSOR FOR THE TEXAS TRAFFIC SIMULATION PACKAGE

LATEST UPDATE: 03 NOV 77

THIS DOCUMENTATION IS DIVIDED INTO THE FOLLOWING SECTIONS:

1. DISTRIBUTION FITTING PROCESSOR LIMITATIONS
2. EXPLANATION OF EXECUTION ERRORS
3. DEFINITION OF THE VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED
4. DEFINITION OF LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL
5. ALPHABETICAL LISTING OF ALL ROUTINES AND THE ROUTINES WHICH CAN CALL THEM
6. GENERALIZED CALLING SEQUENCE DIAGRAM
7. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED

1. DISTRIBUTION FITTING PROCESSOR LIMITATIONS

MAXIMUM EXPONENT FOR IBM -------------------------------------- 150.0
MAXIMUM EXPONENT FOR CDC -------------------------------------- 741.0

DISTRIBUTIONS FOR FITTING DATA:

UNIFORM	LOG NORMAL	NEGATIVE EXPONENTIAL
GAMMA	ERLANG	SHIFTED NEGATIVE EXPONENTIAL

2. EXPLANATION OF EXECUTION ERRORS

STOP 901 IN DISFIT = NO HEADWAYS READ BEFORE END-OF-FILE ENCOUNTERED (NO INPUT DATA)
STOP 902 IN DISFIT = MEAN LESS THAN OR EQUAL TO ZERO (HEADWAYS CANNOT BE NEGATIVE)
STOP 916 IN GAMMAF = BAD ARGUMENT FOR GAMMAF (0.0 LE Z GT 150.0)

3. DEFINITION OF THE VARIABLES IN EACH COMMON BLOCK AND THE ROUTINES IN WHICH EACH COMMON BLOCK IS USED

COMMON / DISVAL / VALUES USED IN FITTING EACH DISTRIBUTION
CHISUM CHIVAL DISFIT ELANG GAMMA LOGNRM NEGEXP PAGPLT SNEGEX UNIFRM

ALPHA(7)	AREA OF INTEGRATION UNDER THE CHI SQUARED DISTRIBUTION
CHICHI(7)	CHI SQUARE VALUE FOR EACH DISTRIBUTION
CONFL(7)	CONFIDENCE LEVEL OF CHI SQUARE TEST WITH <NDF> DEGREES OF FREEDOM
D0P0	DOUBLE PRECISION ZERO
ITITLE(15)	60 CHARACTER TITLE FOR DISTRIBUTION FITTING PROCESSOR
MP3SD	MEAN PLUS 3 STANDARD DEVIATIONS
NDF(7)	NUMBER OF DEGREES OF FREEDOM FOR EACH DISTRIBUTION
N	NUMBER OF HEADWAYS READ
PARAM(7)	REQUIRED PARAMETER FOR EACH DISTRIBUTION
PE(50,7)	EXPECTED VALUE OF HEADWAY FOR EACH DISTRIBUTION
PI	3.1415926535898
PO(51)	OBSERVED VALUE OF HEADWAY
SD	STANDARD DEVIATION OF OBSERVRED HEADWAYS (FOR STATISTICS)
VAR	VARIANCE OF OBSERVED HEADWAYS (FOR STATISTCS)
XKSMCD(7)	CUMULATIVE DIFFERENCE BETWEEN EACH FITTED DISTRIBUTION AND OBSERVED DISTRIBUTION
XMEAN	MEAN OF OBSERVED HEADWAYS

4. DEFINITION OF LOCAL VARIABLES USED IN EACH SUBROUTINE, THE ROUTINES WHICH CAN CALL THEM, AND THE ROUTINES THEY CALL

VARIABLES THAT ARE LOCAL WITHIN SUBROUTINES ARE LISTED BELOW, EXCEPT FOR MOST DO-LOOP INDICES

SUBROUTINE CHISUM

FINDS THE DIFFERENCE BETWEEN OBSERVED AND EXPECTED HEADWAYS AND THE CUMULATIVE DIFFERENCE
(CALLED FROM ERLANG GAMMA LOGNRM NEGEXP SNEGEX UNIFRM)

CHISQD	LOCAL VALUE TO BE STORED INTO CHICHI FOR NDIST DISTRIBUTION
CSUMFE	SUM OF EXPECTED HEADWAYS
CSUMFO	SUM OF OBSERVED HEADWAYS
DIFMAX	MAXIMUM DIFFERENCE BETWEEN CSUMFE AND CSUMFO
FE	EXPECTED FREQUENCY AT A PARTICULAR VALUE OF HEADWAY
FEL	EXPECTED FREQUENCY AT LAST VALUE OF HEADWAY
FO	OBSERVED FREQUENCY AT A PARTICULAR VALUE OF HEADWAY
FOL	OBSERVED FREQUENCY AT LAST VALUE OF HEADWAY
NDIST	IDENTIFICATION NUMBER OF EACH DISTRIBUTION: 1 = UNIFRM 2 = LOGNRM 3 = NEGEXP 4 = SNEGEX 5 = GAMMA 6 = ERLANG (ROUNDED DOWN) 7 = ERLANG (ROUNDED UP)

SUBROUTINE CHIVAL

FINDS THE VALUE OF ALPHA FOR EACH DISTRIBUTION
(CALLED FROM ERLANG GAMMA LOGNRM NEGEXP SNEGEX UNIFRM)
(CALLS GAMMAF)

CONST	CONSTANT BASED ON NUMBER OF DEGREES OF FREEDOM
DFD2	NUMBER OF DEGREES OF FREEDOM DIVIDED BY TWO
F	HEIGHT OF FREQUENCY CURVE ABOVE X-AXIS FOR INTEGRATION
FLAST	VALUE OF F AT LAST X
NDIST	IDENTIFICATION NUMBER OF EACH DISTRIBUTION: 1 = UNIFRM 2 = LOGNRM 3 = NEGEXP 4 = SNEGEX 5 = GAMMA 6 = ERLANG (ROUNDED DOWN) 7 = ERLANG (ROUNDED UP)
NSTEP	MAXIMUM VALUE FOR DO-LOOP
POWER	DFD2 - 1
SUM	CUMULATIVE SUM OF ALPHA
XSTEP	CHI SQUARE FOR EACH DISTRIBUTION DIVIDED BY NSTEP
X	CUMULATIVE SUM OF XSTEP

PROGRAM DISFIT

FINDS THE BEST FITTING DISTRIBUTION FOR OBSERVED HEADWAYS
(CALLS UNIFRM LOGNRM NEGEXP SNEGEX GAMMA ERLANG PAGPLT)

DHEAD	DOUBLE PRECISION VALUE OF HEAD
HEAD	HEADWAY (READ IN)
HMAX	MAXIMUM HEADWAY READ
HMIN	MINIMUM HEADWAY READ
IDWN	(ROUNDED DOWN) MESSAGE FOR ERLANG DISTRIBUTION
IGROUP	GROUP NUMBER INTO WHICH EACH HEADWAY IS ADDED
IUPP	(ROUNDED UP) MESSAGE FOR ERLANG DISTRIBUTION
K	INTEGER VALUE OF PARAMETER FOR ERLANG DISTRIBUTION
NRECAD	RECOVERY ADDRESS FOR FATAL EXECUTION ERROR (CDC ONLY)
RANGE	RANGE OF HEADWAYS
SFO	NUMBER OF OBSERVED VALUES FOR EACH INCREMENT OF HEADWAY (N*PO(I))
SPO	SINGLE PRECISION VALUE OF PO
SSD	SINGLE PRECISION VALUE OF THE STANDARD DEVIATION (SD)
SUMX	SUM OF HEADWAYS (FOR MEAN) (TIME IN SECONDS)
SUMXX	SUM OF SQUARE OF HEADWAYS (FOR VARIANCE)
SVAR	SINGLE PRECISION VALUE OF THE VARIANCE (VAR)
SXMEAN	SINGLE PRECISION VALUE OF THE MEAN (XMEAN)
TIME	TIME (IN HOURS)
VOLUME	VOLUME (IN VEH/HR)

SUBROUTINE ERLANG

COMPUTES THEORETICAL ERLANG DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS CHISUM CHIVAL)

ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 0.1)
ALPHAE	XMEAN/VAR
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
CONST	ALPHAE/IFACT
DT	INCREMENTAL VALUE OF TIME FOR INTEGRATION
F	VALUE OF THE CUMULATIVE THEORITICAL DISTRIBUTION AT T
FLAST	VALUE OF F AT T - DT
IFACT	FACTORIAL OF KM1
K	MEAN**2/VAR + XROUND
KM1	K - 1
NDIST	IDENTIFICATION NUMBER OF THIS DISTRIBUTION: 6 = ROUNDED DOWN 7 = ROUNDED UP
NUM	NUMBER OF ITERATIONS FOR EACH DT OF INTEGRATION
SXEXP	SINGLE PRECISION VALUE OF XEXP
T	TIME (ABCISSA)
XEXP	EXPONENT = -ALPHAE*T
XROUND	VALUE (0 OR 1) ADDED TO K, THEN TRUNCATED; YIELDING ROUNDING DOWN (XROUND=0), OR ROUNDING UP (XROUND=1)

SUBROUTINE GAMMA

COMPUTES THEORETICAL GAMMA DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS GAMMAF CHISUM CHIVAL)

A	XMEAN**2/VAR
ALPHAG	XMEAN/VAR
ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 0.1)
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
CONST	ALPHAG/(FACTORIAL OF A)
DT	INCREMENTAL VALUE OF TIME FOR INTEGRATION
F	VALUE OF THE CUMULATIVE THEORITICAL DISTRIBUTION AT T
FLAST	VALUE OF F AT T - DT
NUM	NUMBER OF ITERATIONS FOR EACH DT OF INTEGRATION
SXEXP	SINGLE PRECISION VALUE OF XEXP
T	TIME (ABCISSA)
XEXP	EXPONENT

FUNCTION GAMMAF

COMPUTES FACTORIALS OF REAL NUMBERS
(CALLED FROM CHIVAL GAMMA)

A00 A01 . . . A12 A13	VALUES USED TO COMPUTE THE FACTORIAL OF T
GAMMAF	FACTORIAL OF X
P	FACTORIAL OF A NUMBER BETWEEN 0.0 AND 1.0
SX	SINGLE PRECISION VALUE OF X
T	A NUMBER BETWEEN 0.0 AND 1.0
X	REAL NUMBER PASSED TO THIS FUNCTION
Z	WORKING VALUE, INITIALIZED TO X

SUBROUTINE LOGNRM COMPUTES THEORETICAL LOG NORMAL DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS CHISUM CHIVAL)

ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 0.1)
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
CONST	1 / SQRT(2*PI*YVAR)
DT	INCREMENTAL VALUE OF TIME FOR INTEGRATION
F	VALUE OF THE CUMULATIVE THEORITICAL DISTRIBUTION AT T
FLAST	VALUE OF F AT T - DT
NUM	NUMBER OF ITERATIONS FOR EACH DT OF INTEGRATION
SXEXP	SINGLE PRECISION VALUE OF XEXP
T	TIME (ABCISSA)
XEXP	EXPONENT
YMEAN	LOG(XMEAN) - 0.5*LOG((VAR/XMEAN**2) + 1)
YVAR	LOG((VAR/XMEAN**2) + 1)

SUBROUTINE NEGEXP COMPUTES THEORETICAL NEGATIVE EXPONENTIAL DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS CHISUM CHIVAL)

ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 0.1)
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
SXEXP	SINGLE PRECISION VALUE OF XEXP
T	TIME (ABCISSA)
XEXP	EXPONENT

SUBROUTINE PAGPLT LINE PLOTS OF EXPECTED FREQUENCY AND OBSERVED HEADWAYS ON PRINTER OUTPUT
(CALLED FROM DISFIT)

I	IDENTIFICATION NUMBER OF THIS DISTRIBUTION
ICOL	POINT THROUGH WHICH PLUS CHARACTERS ARE DRAWN
J	UPPER CLASS BOUNDARY (HEADWAY TIME)
JM1	LOWER CLASS BOUNDARY (HEADWAY TIME)
K	POINT WHERE BLANK CHARACTERS START ON THE END OF THE CURRENT LINE TO BE DRAWN
L	POINTS WHERE BLANKS ARE RETAINED
LINE	A LINE OF 120 CHARACTERS
NB	POINT TO WHICH PLUS CHARACTERS ARE TO BE DRAWN
NBLANK	A BLANK CHARACTER ( )
NCHAR	CHARACTER AT TERMINAL POINT OF EACH LINE
NCLOSE	A CLOSE PARENTHESIS CHARACTER TO INDICATE WHEN OBSERVED AND/OR EXPECTED FREQUENCY EXCEEDS 30 PERCENT
NEQUAL	AN EQUAL CHARACTER (=) OCCURS WHEN OBSERVED FREQUENCY = EXPECTED FREQUENCY
NPLUS	A PLUS CHARACTER (+) REPRESENTING OBSERVED FREQUENCY
NS	POINT AT WHICH STAR CHARACTER IS TO BE DRAWN
NSTAR	A STAR CHARACTER (*) REPRESENTING EXPECTED FREQUENCY

SUBROUTINE SNEGEX COMPUTES THEORETICAL SHIFTED NEGATIVE EXPONENTIAL DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS CHISUM CHIVAL)

ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 1.0)
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
CONST	1 / (XMEAN - TAU)
SXEXP	SINGLE PRECISION VALUE OF XEXP
SXMEAN	SINGLE PRECISION VALUE OF XMEAN
T	TIME (ABCISSA)
TAU	SHIFT (XMEAN - SD)
XEXP	EXPONENT

SUBROUTINE UNIFRM COMPUTES THEORETICAL UNIFORM DISTRIBUTION
(CALLED FROM DISFIT)
(CALLS CHISUM CHIVAL)

A	LOWER BOUNDARY OF AREA OF INTEREST
ALAST	LAST AREA (BEFORE T WAS INCREMENTED BY 1.0)
AREA	AREA UNDER THE THEORITICAL DISTRIBUTION TO THE LEFT OF T (ABCISSA VALUE)
B	UPPER BOUNDARY OF AREA OF INTEREST
CONST	1 / (B - A)
SA	SINGLE PRECISION VALUE OF A
SQRT3	SQUARE ROOT OF THREE (1.732050808)
T	TIME (ABCISSA)

5. ALPHABETICAL LISTING OF ALL ROUTINES AND THE ROUTINES WHICH CAN CALL THEM

CHISUM - ERLANG GAMMA LOGNRM NEGEXP SNEGEX UNIFRM
CHIVAL - ERLANG GAMMA LOGNRM NEGEXP SNEGEX UNIFRM
ERLANG - DISFIT
GAMMA - DISFIT
GAMMAF - CHIVAL GAMMA
LOGNRM - DISFIT
NEGEXP - DISFIT
PAGPLT - DISFIT
SNEGEX - DISFIT
UNIFRM - DISFIT

6. GENERALIZED CALLING SEQUENCE DIAGRAM

```
                              DISFIT
                                ↓
                                ↓
    ---------------------------------------------------------------------
    ↓         ↓          ↓          ↓          ↓          ↓         ↓
    ↓         ↓          ↓          ↓          ↓          ↓         ↓
  UNIFRM    LOGNRM     NEGEXP     SNEGEX     GAMMA      ERLANG     PAGPLT
  →CHISUM   →CHISUM    →CHISUM    →CHISUM    →GAMMAF    →CHISUM
  →CHIVAL   →CHIVAL    →CHIVAL    →CHIVAL    →CHISUM    →CHIVAL
   >GAMMAF   >GAMMAF    >GAMMAF    >GAMMAF   →CHIVAL     >GAMMAF
                                              >GAMMAF
```

7. ALPHABETICAL LISTING OF ALL THE VARIABLES, THEIR STORAGE TYPE, AND THE ROUTINES IN WHICH THEY ARE USED

```
A      =        = GAMMA  UNIFRM
ALAST  =        = ERLANG GAMMA  LOGNRM NEGEXP SNEGEX UNIFRM
ALPHA  / DISVAL / CHIVAL DISFIT
ALPHAE =        = ERLANG
ALPHAG =        = GAMMA
AREA   =        = ERLANG GAMMA  LOGNRM NEGEXP SNEGEX UNIFRM
A00
A01
 .
 .     =        = GAMMAF
 .
A12
A13
B      =        = UNIFRM
CHICHI / DISVAL / CHISUM CHIVAL DISFIT
CHISQD =        = CHISUM
CONFL  / DISVAL / CHIVAL DISFIT
CSUMFE =        = CHISUM
CSUMFO =        = CHISUM
DFD2   =        = CHIVAL
DHEAD  =        = DISFIT
DIFMAX =        = CHISUM
DT     =        = ERLANG GAMMA  LOGNRM
D0P0   / DISVAL / CHISUM CHIVAL DISFIT ERLANG GAMMA  LOGNRM NEGEXP SNEGEX
                  UNIFRM
F      =        = CHIVAL ERLANG GAMMA  LOGNRM
FE     =        = CHISUM
FEL    =        = CHISUM
FLAST  =        = CHIVAL ERLANG GAMMA  LOGNRM
FO     =        = CHISUM
FOL    =        = CHISUM
HEAD   =        = DISFIT
HMAX   =        = DISFIT
HMIN   =        = DISFIT
I      =        = CHISUM CHIVAL DISFIT ERLANG GAMMA  LOGNRM NEGEXP PAGPLT
                  SNEGEX UNIFRM
ICOL   =        = PAGPLT
IDWN   =        = DISFIT
IFACT  =        = ERLANG
IGROUP =        = DISFIT
ITITLE / DISVAL / DISFIT
IUPP   =        = DISFIT
IZ     =        = DISFIT
J      =        = DISFIT ERLANG GAMMA  LOGNRM PAGPLT
JM1    =        = PAGPLT
K      =        = DISFIT ERLANG PAGPLT
KM1    =        = ERLANG
L      =        = PAGPLT
LINE   =        = PAGPLT
MP3SD  / DISVAL / DISFIT ERLANG GAMMA  LOGNRM
N      / DISVAL / CHISUM DISFIT
NB     =        = PAGPLT
NBLANK =        = PAGPLT
NCHAR  =        = PAGPLT
NCLOSE =        = PAGPLT
NDF    / DISVAL / CHISUM CHIVAL DISFIT ERLANG GAMMA  LOGNRM NEGEXP SNEGEX
                  UNIFRM
NDIST  =        = CHISUM CHIVAL ERLANG
NEQUAL =        = PAGPLT
NPLUS  =        = PAGPLT
NRECAD =        = DISFIT
NS     =        = PAGPLT
NSTAR  =        = PAGPLT
NSTEP  =        = CHIVAL
NUM    =        = ERLANG GAMMA  LOGNRM
P      =        = GAMMAF
PARAM  / DISVAL / DISFIT ERLANG GAMMA  LOGNRM NEGEXP SNEGEX UNIFRM
PE     / DISVAL / CHISUM ERLANG GAMMA  LOGNRM NEGEXP PAGPLT SNEGEX UNIFRM
PI     / DISVAL / DISFIT LOGNRM
PO     / DISVAL / CHISUM DISFIT PAGPLT
POWER  =        = CHIVAL
RANGE  =        = DISFIT
SA     =        = UNIFRM
SD     / DISVAL / DISFIT LOGNRM SNEGEX UNIFRM
SFO    =        = DISFIT
SPO    =        = DISFIT
SQRT3  =        = UNIFRM
SSD    =        = DISFIT
SUM    =        = CHIVAL
SUMX   =        = DISFIT
SUMXX  =        = DISFIT
SVAR   =        = DISFIT
SX     =        = GAMMAF
SXEXP  =        = ERLANG GAMMA  LOGNRM NEGEXP SNEGEX
SXMEAN =        = DISFIT SNEGEX
T      =        = ERLANG GAMMA  GAMMAF LOGNRM NEGEXP SNEGEX UNIFRM
TAU    =        = SNEGEX
TIME   =        = DISFIT
VAR    / DISVAL / DISFIT ERLANG GAMMA  LOGNRM
VOLUME =        = DISFIT
X      =        = CHIVAL GAMMAF
XEXP   =        = ERLANG GAMMA  LOGNRM NEGEXP SNEGEX
XKSMCD / DISVAL / CHISUM DISFIT
XMEAN  / DISVAL / DISFIT ERLANG GAMMA  LOGNRM NEGEXP SNEGEX UNIFRM
XROUND =        = ERLANG
XSTEP  =        = CHIVAL
YMEAN  =        = LOGNRM
YVAR   =        = LOGNRM
Z      =        = GAMMAF
```

PARTIAL LIST OF RESEARCH REPORTS PUBLISHED BY THE CENTER FOR HIGHWAY RESEARCH

(Continued from inside front cover)

121-5 "Construction and Load Tests of a Segmental Precast Box Girder Bridge Model," by S. Kashima and J. E. Breen, February 1975.

121-6F "Minimizing Construction Problems in Segmentally Precast Box Girder Bridges," by J. E. Breen, R. L. Cooper, and T. M. Gallaway, August 1975.

123-16 "Fatigue and Stress Analysis Concepts for Modifying the Rigid Pavement Design System," by Piti Yimprasert and B. Frank McCullough, January 1973 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-18 "Probabilistic Design Concepts Applied to Flexible Pavement System Design," by Michael I. Darter and W. Ronald Hudson, May 1973 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-19 "Use of Condition Surveys, Profile Studies, and Maintenance Studies In Relating Pavement Distress to Pavement Performance," by Robert Phillip Smith and B. Frank McCullough, May 1974 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-21 "Rigid Pavement Design System Input Guide for Computer Program RPS2," by Robert F. Carmichael and B. Frank McCullough, May 1974 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-23 "Stochastic Study of Design Parameters and Lack-of-Fit of Performance Model in the Texas Flexible Pavement Design System," by Malvin Holsen and W. Ronald Hudson, April 1974 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-26 "Modification and Implementation of the Rigid Pavement Design System," by Robert F. Carmichael and B. F. McCullough, January 1975 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-27 "Influence of Cognitive Style in a Methodology for Data Base Design," by Ronald R. Bush, February 1975 (published jointly with the Texas Highway Department and the Texas Transportation Institute, Texas A&M University).

123-30F "Overview of Pavement Management Systems Developments in the State Department of Highways and Public Transportation," by W. Ronald Hudson, B. Frank McCullough, Jim Brown, Gerald Peck, and Robert L. Lytton, January 1976 (published jointly with the Texas State Department of Highways and Public Transportation and the Texas Transportation Institute, Texas A&M University).

124-1F "Functional Classification of Highway Systems," by Walter C. Vodrazka and John L. Staha, June 1972.

153-1F "Strength and Serviceability of Inverted T-Beam Bent Caps Subject to Combined Flexure, Shear, and Torsion," by Richard W. Furlong and Sher Ali Mirza, August 1974.

154-1 "The Behavior of Multiple Lap Splices in Wide Sections," by Mark A. Thompson, James O. Jirsa, John E. Breen, and Donald F. Meinheit, January 1975.

154-2 "The Performance of Lapped Splices Under Rapid Loading," by T. Rezansoff, M. P. Bufkin, J. O. Jirsa, and J. E. Breen, January 1975.

154-3F "The Strength of Anchored Bars: A Reevaluation of Test Data on Development Length and Splices," by C. O. Orangun, J. O. Jirsa, and J. E. Breen, January 1975.

155-1F "Static and Buckling Analysis of Highway Bridges by Finite Element Procedures," by C. Philip Johnson, Thaksin Thepchatri, and Kenneth M. Will, August 1973.

156-1 "A Correlation Study of the Mays Road Meter with the Surface Dynamics Profilometer," by Roger S. Walker and W. Ronald Hudson, February 1973.

156-2 "The Use of Spectral Estimates for Pavement Characterization," by Roger S. Walker and W. Ronald Hudson, August 1973.

156-3 "Analysis of Characteristic Roughness Patterns in Pavements and the Relationship Between Roughness and Pavement Distress," by Hugh J. Williamson and W. Ronald Hudson, February 1974.

156-4 "The Characterization of Road Roughness on Bridge Decks and the Adjoining Pavement," by David B. Law, Hugh J. Williamson, and W. Ronald Hudson, April 1975.

156-5F "A Study of the Relationships Between Various Classes of Road-Surface Roughness and Human Ratings of Riding Quality," by Hugh J. Williamson, W. Ronald Hudson, and C. Dale Zinn, August 1975.

157-1F "Static, Fatigue, and Impact Strength of Electroslag Weldments," by J. S. Noel and A. A. Toprac, December 1972.

158-1F "The Effect of Diaphragms in Prestressed Concrete Girder and Slab Bridges," by S. Sengupta and J. E. Breen, October 1973.

160-1F "Dynamic Traffic Loading of Pavements," by Randy Machemehl and Clyde E. Lee, December 1974.

161-1 "A Survey of Earth Slope Failures and Remedial Measures in Texas," by Timothy G. Abrams and Stephen G. Wright, December 1972.

161-2F "A Survey and Evaluation of Remedial Measures for Earth Slope Stabilization," by Rudolph G. Schweizer and Stephen G. Wright, August 1974.

176-1 "The Behavior of Axially Loaded Drilled Shafts in Sand," by Fadlo T. Touma and Lymon C. Reese, December 1972.

176-2 "Behavior of an Axially Loaded Drilled Shaft Under Sustained Loading," by John A. Wooley and Lymon C. Reese, May 1974.

176-3 "Behavior of Three Instrumented Drilled Shafts Under Short Term Axial Loading," by Donald E. Engeling and Lymon C. Reese, May 1974.

177-1 "Drying Shrinkage and Temperature Drop Stresses in Jointed Reinforced Concrete Pavement," by Felipe Rivero-Vallejo and B. Frank McCullough, May 1976.

177-2 "A Sensitivity Analysis of Continuously Reinforced Concrete Pavement Model CRCP-1 for Highways," by Chypin Chiang, B. Frank McCullough, and W. Ronald Hudson, August 1975.

181-1F "Truck Weight Surveys by In-Motion Weighing," by Randy B. Machemehl, Clyde E. Lee, and C. Michael Walton, September 1975.

183-1 "Tensile and Elastic Characteristics of Pavement Materials," by Bryant P. Marshall and Thomas W. Kennedy, January 1974.

183-2 "Fatigue and Repeared-Load Elastic Characteristics of Inservice Asphalt-Treated Materials," by Domingo Navarro and Thomas W. Kennedy, January 1975.

183-3 "Cumulative Damage of Asphalt Materials Under Repeated-Load Indirect Tension," by Calvin E. Cowher and Thomas W. Kennedy, January 1975.

183-4 "Comparison of Fatigue Test Methods for Asphalt Materials," by Byron W. Porter and Thomas W. Kennedy, April 1975.

502-1F "Evaluation and Revision of Texas Highway Department Rigid Pavement Design Procedure," by B. Frank McCullough, Harvey J. Treybig, and Ramesh K. Kher, November 1972.

505-1F "Utilization of New Analytical Methods in Bridge Design," by John J. Panak, August 1974.

514-1F "Effects of Temperature Change on Plastic Crash Cushions," by Victor N. Toth and Clyde E. Lee, January 1976.