TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/TX-90+1129-1		

4. Title and Subtitle	5. Report Date
COMPUTER PROGRAM FOR THE ANALYSIS OF BRIDGE BENT COLUMNS INCLUDING A GRAPHICAL INTERFACE	November 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
R. W. Stocks, C. P. Johnson, and J. M. Roesset	Research Report 1129-1
9. Performing Organization Name and Address	10. Work Unit No.
Center for Transportation Research The University of Texas at Austin Austin, Texas 78712-1075	11. Contract or Grant No. Research Study 3-5-86-1129
	13. Type of Report and Period Covered Interim
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P. O. Box 5051 Austin, Texas 78763-5051	14. Sponsoring Agency Code

15. Supplementary Notes

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

Research Study Title: "Bent Column Analysis and Design"

16. Abstract

The FORTRAN computer program BENTCOL was developed to provide an easy-to-use analysis tool for bridge bent columns which alleviates most of the cumbersome and tedious details involved with the current design process. An emphasis was placed on minimizing the required amount of input data and on producing clear, functional, and easy-to-read output. This computer program involved computerizing the Texas State Department of Highways and Public Transportation (TSDHPT) current office procedure as well as the development of an integral frame analysis. The output from both the approximate and frame analysis is in the exact form required as input data for the column design program PCA2, making the transition from analysis to design simple and direct. Using this new computer program as a design tool, the design engineer may analyze and design bent columns quickly and easily.

The PASCAL computer program BCINPUT was developed to investigate the use of a graphical interface for handling input arrangements on a microcomputer, specifically the IBM-AT personal computer. The microcomputer provides an inexpensive and innovative approach for handling input arrangements. The graphical interface emphasizes user-driven menus, user-friendliness, user-interaction, and graphical images to aid in the process of creating input data files. Using the graphical interface, the input data file required for the analysis program is created and edited easily, completing the analysis and design package.

17. Key Words	18. Distribution Statement
computer program, analysis, design, bridge bent columns, input data, integral frame analysis, approximate, graphical interface, microcomputer	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	168	

Form DOT F 1700.7 (8-69)

COMPUTER PROGRAM FOR THE ANALYSIS OF BRIDGE BENT COLUMNS INCLUDING A GRAPHICAL INTERFACE

by

R. W. Stocks
C. P. Johnson
J. M. Roesset

Research Report Number 1129-1

Research Project 3-5-86-1129

Bent Column Analysis and Design

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 TRANSPORTATION RESEARCH

Bureau of Engineering Research
THE UNIVERSITY OF TEXAS AT AUSTIN

November 1988

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 research was sponsored by the Texas State Department of Highways and Public Transportation. The research was conducted at The University of Texas at Austin. It was conducted over a period of two years.

The authors wish to thank Richard L. Wilkison of the TSDHPT Bridge Division for his involvement throughout the project. Thanks are also expressed to the CTR staff, in particular John Sutherland, who assisted us with the administrative aspects of the project. Special thanks go to Kay Lee for assisting us with the interim project reports as well as with the final project reports.

R. W. Stocks
C. P. Johnson
J. M. Roesset

Austin, Texas
November 1988

LIST OF REPORTS

Research Report No. 1129-1, "Computer Program for the Analysis of Bridge Bent Columns Including a Graphical Interface," by R. W. Stocks, C. P. Johnson, and J. M. Roesset, presents the development of a computer program to determine axial forces and moments in columns of bridge bents accounting for the AASHTO loading combinations of Load Groups I, II, and III and using either the simplified procedure of the Texas State Department of Highways and Public Transportation or an integral frame analysis. A graphical interface was developed for the IBM-AT microcomputer to input the needed data in a user-friendly, self-explanatory way. The computer program was adapted to the facilities of the Texas State Department of Highways and Public Transportation.

ABSTRACT

The FORTRAN computer program BENTCOL was developed to provide an easy-to-use analysis tool for bridge bent columns which alleviates most of the cumbersome and tedious details involved with the current design process. An emphasis was placed on minimizing the required amount of input data and on producing clear, functional, and easy-to-read output. This computer program involved computerizing the Texas State Department of Highways and Public Transportation (TSDHPT) current office procedure as well as the development of an integral frame analysis. The output from both the approximate and frame analysis is in the exact form required as input data for the column design program PCA2, making the transition from analysis to design simple and direct. Using this new computer program as a design tool, the design engineer may analyze and design bent columns quickly and easily.

The PASCAL computer program BCINPUT was developed to investigate the use of a graphical interface for handling input arrangements on a microcomputer, specifically the IBM-AT personal computer. The microcomputer provides an inexpensive and innovative approach for handling input arrangements. The graphical interface emphasizes user-driven menus, user-friendliness, user-interaction, and graphical images to aid in the process of creating data input files. Using the graphical interface, the input data file required for the analysis program is created and edited easily, completing the analysis and design package.

SUMMARY

The column analysis computer program BENTCOL allows the design engineer to quickly determine the forces needed for the design of TSDHPT bridge columns. BENTCOL has two analysis options. The first is a computerized version of the TSDHPT current office procedure using AASHTO Loading Groups I, II, and III for finding column axial loads and bending moments. The second is an integral frame analysis which was used primarily for investigating the accuracy of the current office procedure. The design engineer may choose either, or both, options using identical input data files.

The graphical interface BCINPUT developed on an IBM-AT microcomputer provides an innovative approach for bent column design. Using the event-driven graphical interface, the user can create with ease the input data file in a user-friendly environment. It has the intelligence to provide warnings and alerts in the event of unusual actions. It combines text, graphics, color and user events for a user-friendly and self-explanatory link between the computer and the user. Removing the difficulty of creating the input data allows for more time in reviewing and interpreting results.

IMPLEMENTATION STATEMENT

As a result of this research, a computer program called BENTCOL has been developed on an IBM-AT personal computer. It has been adapted to the computer facilities at the TSDHPT. The engineer may use BENTCOL to design the bent columns using the current office procedure or by considering integral behavior of the bent cap and column system. The first use allows for the design of bent columns without resorting to cumbersome and tedious hand calculations previously required. The second use requires fewer assumptions and may produce column design forces which better represent the actual behavior of the bridge columns.

TABLE OF CONTENTS

PREFACE iii

LIST OF REPORTS iii

ABSTRACT iii

SUMMARY iii

IMPLEMENTATION STATEMENT iv

CHAPTER 1. INTRODUCTION
- 1.1 Background 1
  - 1.1.1 Current Texas Highway Department Approach 1
- 1.2 Outline of Present Research 1
  - 1.2.1 Objectives and Purpose 1
  - 1.2.2 Analysis Procedures 4
- 1.3 Organization of Report 5

CHAPTER 2. FORTRAN PROGRAM BENTCOL
- 2.1 General Considerations 6
  - 2.1.1 Approximate Procedure 6
  - 2.1.2 Frame Analysis 7
- 2.2 Documentation of BENTCOL 9
  - 2.2.1 Input Guide and User Information 9
  - 2.2.2 Definition of Input Variables 12
  - 2.2.3 Description of Output 12
- 2.3 Example Problems 13
  - 2.3.1 Example Problem 1 13
  - 2.3.2 Example Problem 2 14
- 2.4 General Remarks and Comparisons 14

CHAPTER 3. PASCAL GRAPHICAL INTERFACE BCINPUT
- 3.1 General Considerations 16
- 3.2 Alternatives to a Graphical Interface 16
- 3.3 Graphical Interfaces 16
- 3.4 Description of BCINPUT 17
- 3.5 General Remarks 23

CHAPTER 4. CONCLUSIONS AND RECOMMENDATIONS
- 4.1 Conclusions 24
- 4.2 Recommendations for Future Research 24

REFERENCES 25

APPENDIX A. COMPUTER OUTPUT FOR EXAMPLE PROBLEM 1 27

APPENDIX B. COMPUTER OUTPUT FOR EXAMPLE PROBLEM 2 48

APPENDIX C. LISTING OF FORTRAN PROGRAM BENTCOL 70

APPENDIX D. LISTING OF PASCAL GRAPHICAL INTERFACE BCINPUT 122

CHAPTER 1. INTRODUCTION

1.1 BACKGROUND

The design of bridge bent columns requires the determination of axial forces and associated bending moments. Determining reasonable design loads creates several problems for the design engineer. First, a practical and usable analysis procedure must be developed. The assumptions and simplifications inherent to the method should be necessary and reasonable. Second, the analysis procedure must be tested, evaluated, and modified by experienced design engineers. Finally, the procedure is verified through successful application to actual bridge columns that perform as expected by the engineer. Figures 1 and 2 on the following page demonstrate typical Texas highway bridge bents.

The design loads necessary for analyzing bent columns are provided in the American Association of State Highway and Transportation Officials (AASHTO) specifications. The design loads include dead load, live load, live load impact, earth pressure, buoyancy, wind load on structure, wind load on live load, longitudinal force from live load, centrifugal force, rib shortening, shrinkage, temperature, earthquake, stream flow pressure, and ice pressure. The engineer must consider twelve loading combinations, or groups, when applying these design loads to bridge structures.

With a proven analysis procedure in hand, the design engineer is still faced with the task of applying the specified design loads to a particular structure in the evaluation of column axial forces and bending moments. Although a hand analysis is possible, a computerized approach is desirable. The hand analysis may be tedious and time-consuming and may include simplifying assumptions which make the approach possible. The computerized approach allows the design engineer to use a more sophisticated analysis procedure that has fewer assumptions than a similar procedure conceived with hand analysis in mind. The computerized approach also allows the user to easily modify and correct input data, to conduct "what if?" cases, and to review output presented in a clear and concise format.

In the case of bridge column analysis, several sets of axial loads and associated moments are produced. These values must be reviewed by the engineer for use in a column design program. Typically, the values output by the analysis program are transformed by hand into an input data file for use in a design program. However, the computerized analysis approach is very well suited for the task of providing output in a format that is understood by the design program. Linking analysis and design programs together creates a powerful tool for the engineer through computer-aided analysis and design.

1.1.1 Current Texas Highway Department Approach

The design of bent columns requires determining the critical axial loads and bending moments considering twelve loading combinations following AASHTO specifications. Of these twelve loading combinations or groups, three groups govern the design of columns for typical Texas highway bridges. Groups I, II, and III must be used by the design engineer to assign reasonable design loads to the bridge structure in determining column loads.

Currently, the design engineer follows a fill-in-the-blank approximate approach established by the Texas Highway Department. The approximate method takes the form of three generic design sheets which may be tailored to fit a particular bridge. The process is tedious and time-consuming, taking an experienced engineer three to four hours. Although this hand approach is functional, mistakes are easily made and hard to correct due to the numerous computations. After investing a considerable amount of time and effort, the engineer obtains twenty-two sets of column loads consisting of an axial load and two bending moments. These twenty-two values are used as input data for the column design program, **PCA2**. Currently, the computed axial loads and associated moments are keyed-in by hand to create the input data file for **PCA2**. Again, the task before the engineer is time-consuming and cumbersome. Once the input file is created by hand and reviewed, the design program may be utilized to determine the required steel reinforcement for the column under consideration. Finally, the engineer must review the answer and evaluate the practicality of the results. An unreasonable result will warrant review of the entire design process in an effort to detect errors. Unfortunately, if a mistake occurs in the early steps of the analysis, the engineer must sacrifice several additional hours to repeat the process.

1.2 OUTLINE OF PRESENT RESEARCH

The development of a bent column analysis program consisted of three phases. First, the current office procedure was computerized. Second, the computer program was expanded to include an integral frame analysis of the bent cap and column system. Finally, a graphical interface was developed to facilitate the creation of the input data file for the analysis program. The bent column analysis program was implemented on the computer system at the Texas Highway Department, later to be incorporated into a complete Computer-Aided Design and Drafting (CADD) package. The graphical interface is available for use on a microcomputer. A general flow chart of the column design process as approached by this research is presented in Fig 3.

1.2.1 Objectives and Purpose

One objective of this research was to develop an easy-to-use analysis tool for bent columns which alleviates most of the cumbersome and tedious details involved with the current process. An emphasis was placed on minimizing the required amount of input data and on producing clear,

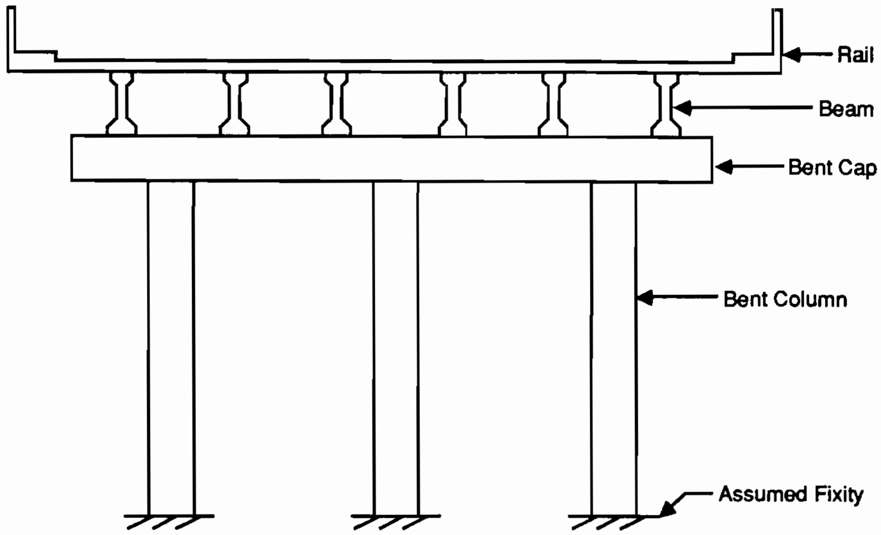


Fig 1. Bridge bent.

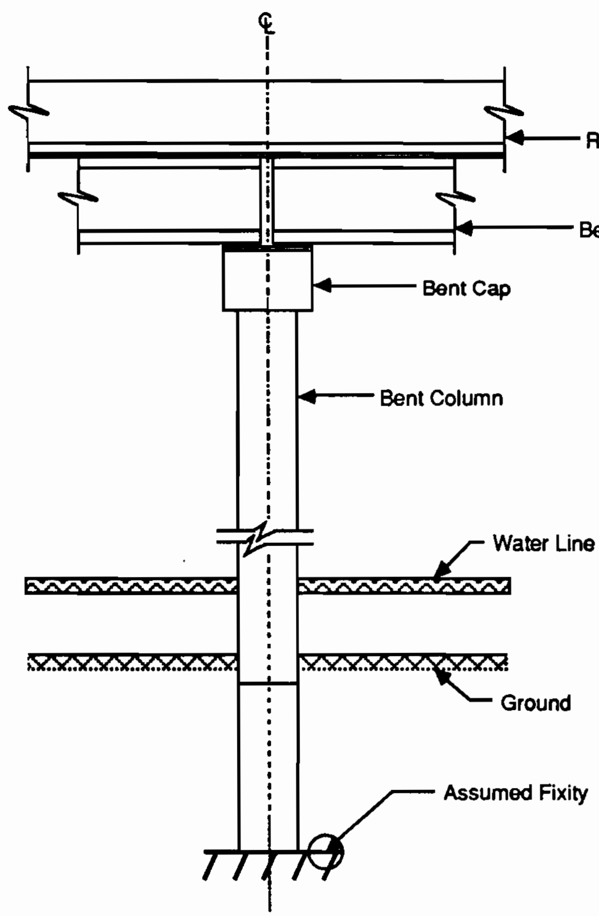


Fig 2. Bent column evaluation.

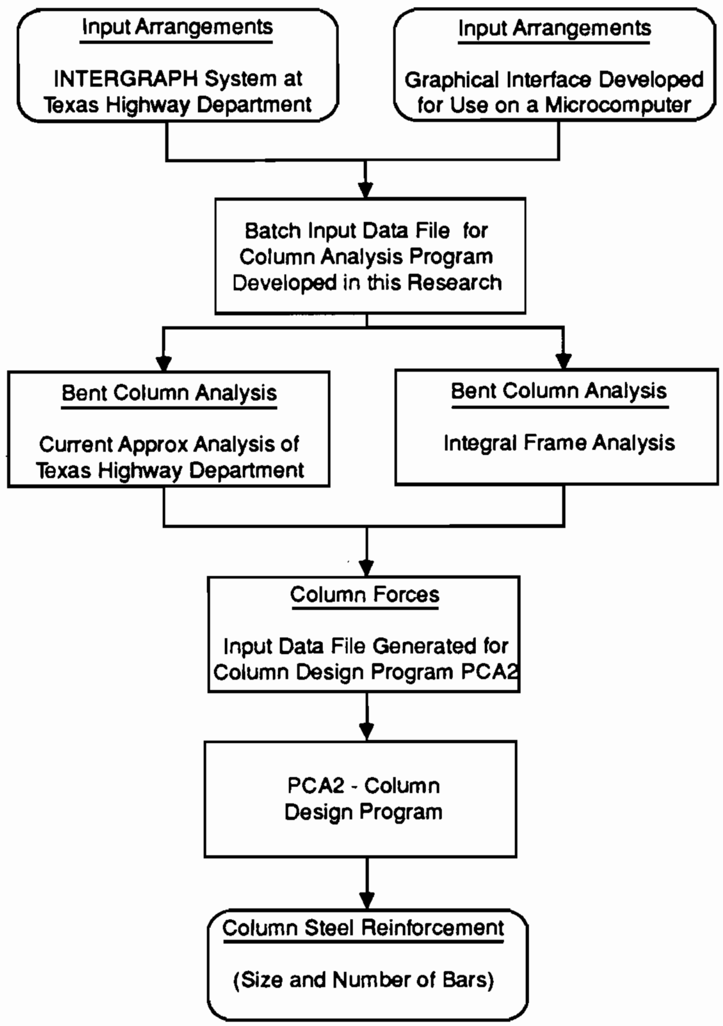


Fig 3. Column design process.

functional, and easy-to-read output. This area of the research involved computerizing the current office procedure as well as the development of an integral frame analysis to study the accuracy of the approximate analysis. The frame analysis requires exactly the same input data as does the approximate method since the member properties, nodal coordinates, and boundary conditions required in the frame analysis are automatically generated by the program. Internally generating necessary data for the frame analysis makes the more sophisticated frame analysis an attractive alternative for the design engineer. The output from both the approximate and frame analysis is in the exact form required as input data for the column design program **PCA2**, making the transition from analysis to design simple and direct. Using this new computer program design tool, the design engineer may analyze and design bent columns quickly and easily. The process not only requires less time but allows the engineer to devote efforts toward using good judgement rather than searching for computational mistakes. This computer program has been implemented on the VAX computer at the the Texas Highway Department. Eventually, the computer program will be incorporated into a Computer-Aided Drafting and Design (CADD) package currently being developed by the Texas Highway Department. The CADD package is being developed on the INTERGRAPH system. A special menu will be produced to create the required input data file for the computer program developed through this research. In the end, the complete CADD package will allow the analysis, design, and drafting of bridge bent columns to be more efficient and accurate than presently possible.

A final purpose of this research was to investigate the use of a graphical interface for handling input arrangements for the analysis program. As an alternative to using a graphically oriented mini-computer such as the INTERGRAPH system, the graphical interface was developed on an IBM personal computer. The microcomputer provides an inexpensive and innovative approach for handling input arrangements. The graphical interface emphasizes user-driven menus, user-friendliness, user-interaction, and graphical images to aid in the process of creating input data files. Using the graphical interface, the input data file required for the analysis program is created and edited easily, completing the analysis and design package.

1.2.2 Analysis Procedures

The primary objective of this research was to develop a computerized approach for the analysis of bridge bent columns which directly produces input for an existing column design program. Emphasis was placed on simplified and minimal input data. The current office procedure involves utilizing an approximate method derived for hand analysis. Since the bent cap and column system is an indeterminate frame, simplifying assumptions must be made in order to make a hand analysis possible. To this end, inflection points are assumed to be located at the mid-height of all columns in the rigid frame, allowing the engineer to easily compute approximate bending moments. This assumption may be reasonable for some cases due to the relative stiffness between the bent cap and the columns. A related drawback with the approximate method is that the increase in axial loads in the presence of horizontal forces is not taken into account. Neglecting the axial forces due to overturning forces results in final axial loads which are not conservative in all cases. A final major drawback with the approximate method is related to the distribution of the total bridge load to the columns. The simplifying assumption is that all columns receive an equal percentage of the total load on the bridge. However, this assumption is not always conservative since the interior columns of a bent typically receive a larger portion of the total load.

Although an integral frame analysis of the bent cap and column system would be very time-consuming by hand, the approach is readily adapted for a computer solution. The development of the more exact frame solution alleviated some of the uncertainties and inaccuracies associated with the approximate analysis. No assumptions must be made concerning the locations of inflection points, and the additional axial force due to overturning and actual distribution of loading are reflected in the final solution. The design of any structure requires many assumptions; thus, reducing the number of assumptions is very desirable to the design engineer. The frame analysis option of the program serves two major purposes. First, comparative case studies between the existing approximate method and the more exact approach are possible. Second, the frame analysis portion of program becomes an attractive alternative for the design engineer since the frame analysis does not require any additional input. The input data file contains exactly the same variables for both the approximate analysis and the frame analysis.

An additional benefit of the computerized solution of the analysis is the ability to produce input data files for the column design program. Both the approximate and frame option automatically generate output files in the form required by **PCA2**, allowing the engineer to transition quickly from analysis to design. Additional output files are also available which present all intermediate computations. Reducing the time and difficulty for analysis and design leaves the design engineer with adequate time and energy to evaluate the final solutions.

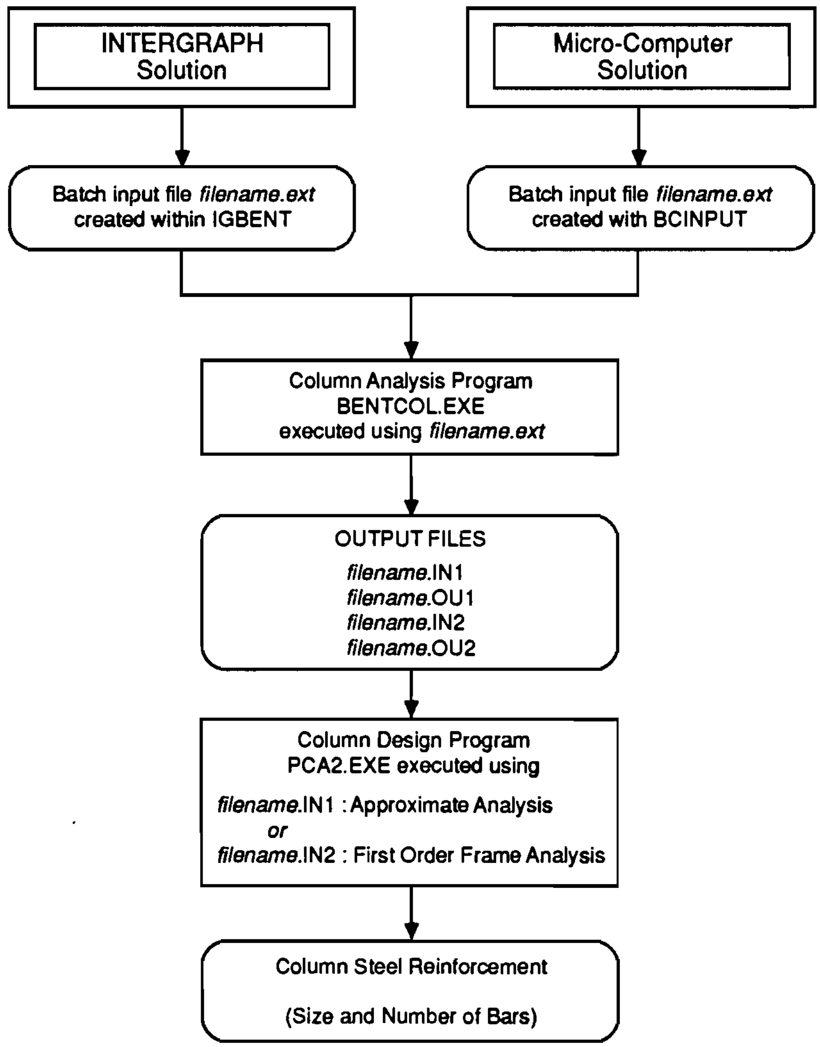
The bent column analysis program was written in FORTRAN IV and developed primarily on the Macintosh Plus microcomputers in the Civil Engineering Micro Lab at The University of Texas at Austin. The graphical interface was written in PASCAL and developed on the IBM-AT microcomputer in the Civil Engineering Micro Lab at The University of Texas at Austin.

1.3 ORGANIZATION OF REPORT

Chapter 2 describes the bent column analysis FORTRAN program, including program documentation and an input guide with descriptive figures. Example problems and comparisons between the approximate and frame analysis are also presented.

Chapter 3 describes the PASCAL graphical interface as well as general comments concerning the handling of input arrangements.

Chapter 4 presents conclusions and recommendations for additional research relating to the analysis of bent columns and useful computer programs.



Summary of column analysis and design.

CHAPTER 2. FORTRAN PROGRAM BENTCOL

2.1 GENERAL CONSIDERATIONS

The analysis of bridge bent columns as presented in this research follows two procedures. First, an analysis option is available which utilizes the current approximate approach as established by the Texas State Department of Highways and Public Transportation (TSDHPT). This approach models the fill-in-the-blank design charts currently being used. The second analysis option utilizes an integral frame analysis which treats the bent cap and column system as a rigid frame. Use of the frame analysis option requires no special knowledge of the matrix solution of the indeterminate structure. The frame analysis utilizes a six degree of freedom linear beam element. A basic understanding of matrix methods of structural analysis will allow the user to better understand the analysis procedure; however, the program automatically generates the member properties, nodal coordinates which define the frame geometry, and the required boundary conditions. Therefore, with a basic understanding of the current office procedures, the user may choose either analysis procedure. Both analysis procedures use the same input data files. The following sections include a description of the approximate and frame analyses, documentation of **BENTCOL**, example problems, and a comparison of the two methods.

2.1.1 Approximate Procedure

The TSDHPT has identified AASHTO Groups I, II, and III as the critical loading combinations which must be considered in the design of typical Texas highway bridge bent columns. In order to apply the design loadings to a particular structure, the bridge must be defined by several variables including bridge geometry, geographic location, and properties of the construction materials. The following paragraphs present the analysis procedure according to the approximate approach.

The first step in the analysis procedure is to determine the Euler buckling load of the column needed for computing the ACI moment magnifier. This computation requires the selection of an appropriate effective length factor as defined in AASHTO Article 8.16.5.2.3. The TSDHPT uses a k-factor of 1.25 for both x-axis and y-axis bending for typical bridge columns. Since the k-factor is an input variable to the program, this value may be modified by the design engineer as necessary.

Next, the axial force on each column is computed. The dead load (DL) per column computations include the weights of the bridge rails, slab, beams, bent cap, and the column itself. The live loads as specified by AASHTO include Lane Live Load (LLLANE) and Truck Live Load (LLTRUCK). As with the dead load, the live load is computed per column. The larger of the two live load values is used in computing Live Load plus Impact (LL+I). Dead Load and Live Load plus Impact are modified by appropriate load factors specified in each Group when computing the total axial load on each column. The load intensity is reduced according to AASHTO Article 3.12 for multiple lane loading in view of the improbability of coincident maximum loading. For one or two lanes, no reduction is allowed. For three lanes, the reduction factor is 0.90. For four lanes, the reduction factor is 0.75. Groups I, II, and III are described in detail in the following three paragraphs. Each of the groups forms a separate subroutine in the FORTRAN computer program. To incorporate other load groups of interest to other states it would be sufficient to add the appropriate subroutines for each group.

Following Load Factor Design, Group I loading combination consists of $1.3[(\beta D^*DL) + 5/3\,(LL+I) + CF + SF]$. The value 1.3 is the load factor. The variable ßD is the load combination coefficient for dead load. A value of 0.75 is assigned when checking a column for the minimum axial load and maximum moment or maximum eccentricity. A value of 1.0 is assigned when checking a column for maximum axial load and minimum moment. Both values are used in the analysis procedure to compute two sets of axial load and associated moments. The value of CF represents the centrifugal force associated with curved bridges. The value of SF represents the stream flow for columns subjected to design water pressure. The column bending moments (separated into components parallel and transverse to the center-line of the roadway) are computed by assuming inflection points at the mid-height of the column and by applying the ACI moment magnifier. In accordance with Group I specifications, the calculated moments are compared with the minimum eccentricity moments to ensure that the critical bending moments are used in design. Using the design loadings and the AASHTO-specified point of application of each design load, Group I produces two sets of axial loads and associated moments.

Group II loading consists of $1.3(\beta D^*DL + SF + W)$. The values of W represent design wind pressures on the bridge superstructure, specified in AASHTO Article 3.15.2.1.1, as well as design wind forces applied directly to the bridge substructure, specified in AASHTO Article 3.15.2.2. The superstructure consists of the rail, slab, haunch, beam, pad, and pedestal. The substructure consists of the cap and column. As in Group I, the column bending moments are resolved into components parallel and transverse to the center-line of the roadway. The design wind forces provided by AASHTO are derived on a base wind velocity of 100 miles per hour. According to AASHTO Article 3.15, the design forces must be reduced or increased by the ratio of the square of the design wind velocity to the square of the base wind velocity. Stream flow contribution must also be considered in Group II. Since both stream flow and wind pressure may act simultaneously, special consideration must be given to the moments computed for these forces acting directly on the column. To ensure that a maximum value is always obtained, the perpendicular (in-plane) moment is first computed assuming wind force over the total height of the column. Second, the stream flow moment perpendicular is computed using the design water depth. If the stream flow moment is larger than the "wind on column" moment, the difference between the two values is added to the total design moment. Otherwise, the presence of stream flow is in effect ignored since it does not govern. Conservatively, the parallel (out-of-plane) moment due to wind on the column is computed using the total height of the column. As with Group I, the final summations of design moments are modified by the ACI moment magnifier, producing ten sets of axial loads and associated moments.

Group III loading consists of 1.3[ßD*DL + (LL+I) + .3W + WL + LF + SF + CF]. The values of WL represent lateral forces per linear foot of wind load on moving live load, specified in AASHTO Article 3.15.2.1.2. The values of LF represent the longitudinal force as a percentage of the live load, specified in AASHTO Article 3.9. Note that according to AASHTO Article 3.15 the design pressures for wind on live load are not reduced or increased using the "squared" ratio described in the preceding paragraph. Also, note that the stream flow moment must be handled similarly for Group III. For Group III, the additional moment is computed by subtracting 0.3 times the "wind on column" moment from the stream flow moment. The summations of design moments are modified by the ACI moment magnifier, producing ten sets of axial load and associated moments.

The approximate analysis produces a total of twenty-two sets of axial load and associated moments. Two output data files produced by the computer program are associated with the approximate analysis. One file, named "APROX.OUT", contains a detailed echo of most of the computations performed throughout the analysis. This detailed version of the output follows a format similar to that of the fill-in-the-blank design sheets, providing the design engineer with the opportunity to review the intermediate steps of the analysis. A second file, named "INPUT.PCA", contains the input data necessary for running the column design program, **PCA2**. This data file is used as a direct link to the **PCA2** program, allowing the design engineer to quickly transition from analysis to design.

2.1.2 Frame Analysis

The frame analysis treats the bent cap and column system as a rigid frame. As with the approximate analysis, AASHTO Groups I, II, and III are utilized in computing the design axial forces and moments. A primary difference between the two analysis procedures is that no assumptions are made concerning locations of inflection points with the integral frame solution. Another major advantage of the frame analysis is that the distribution of axial load is accurately proportioned to each column. Recall that the approximate method equally distributed the total bridge load between all columns. The frame solution more practically distributes the load evenly between all the beams. Also, with the frame analysis each column has a unique axial load and associated moments for each load combination and wind direction. For example, a three column bent produces three times the number of sets of design loads with the frame analysis. Instead of the standard twenty-two always obtained with the approximate analysis, sixty-six (22 times the number of columns) sets of axial loads and associated moments are produced. However, this increase represents no additional work for the design engineer since the input data file for **PCA2** is automatically generated by the analysis program.

As mentioned in the discussion of the approximate analysis procedure, AASHTO specifies the locations on the bridge structure at which design forces must be applied. By assuming mid-height inflection points, the approximate method conservatively computes moments based on an increased column height by applying the design forces at the fictitious locations. For the frame solution, forces must be applied on the frame itself. Thus, to approximate the AASHTO requirements, the design forces were increased by the ratio of the height of the AASHTO-specified location to the distance between the cap center-line and assumed fixity (F*D = f*d; i.e., f = F*D/d). The larger force "f" is applied to compute moments that result from forces acting above the structure, as shown in Fig 4.

The direct stiffness method with three degrees of freedom at each joint in the frame is utilized in the frame analysis. The frame is idealized using linear line segments to represent each member. The standard assumptions of small deformations, plane sections remaining plane, and true rigid connections between members are made. Also, it is assumed that members are originally straight and that a linear elastic material following Hooke's Law is being used. Finally, the cap and the columns must be prismatic. The solution provided by the frame analysis yields the axial load and in-plane bending moment for each column. These member forces evolve from an elastic analysis, and must therefore be modified by the ACI moment magnifier to approximate second order effects.

The frame analysis utilizes the same input data file used in the approximate analysis to generate the additional information necessary for the matrix solution of the rigid frame. The generator assumes uniform beam and uniform column spacing as well as a rigid base at the assumed fixity. First, a subroutine is called which generates and numbers frame members and nodal coordinates. Also, boundary conditions and point load locations are assigned. With this information, the frame may be analyzed quickly utilizing a banded matrix solver. The frame is analyzed by placing concentrated unit loads or uniformly distributed unit loading at all appropriate locations on the frame as shown in Fig 5, UNIT LOAD ANALYSES. Specifically, concentrated unit loads

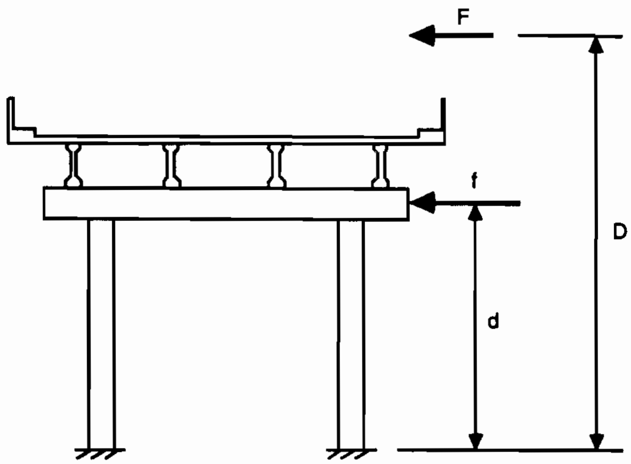


Fig 4. Force on bridge superstructure.

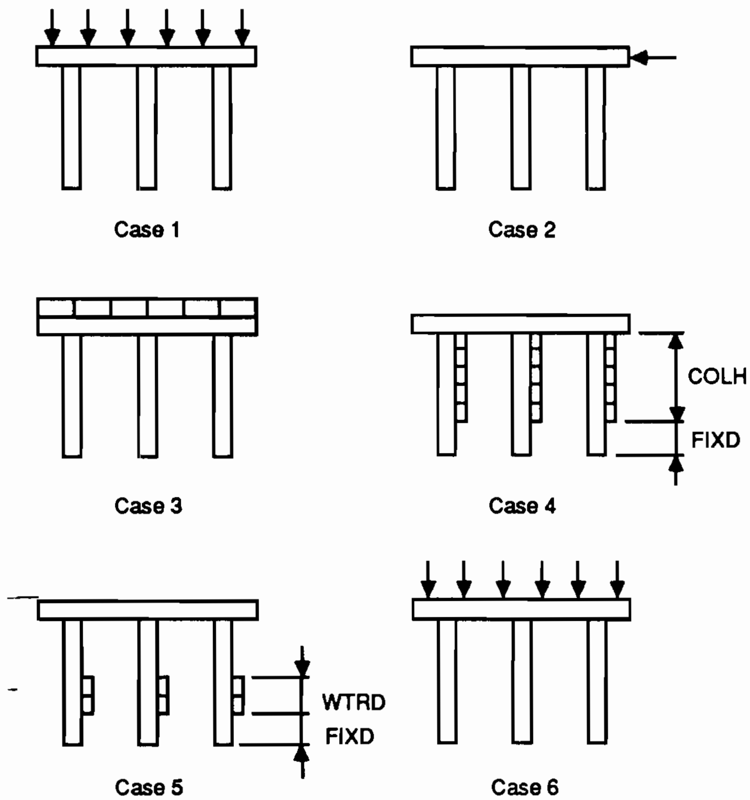


Fig 5. Unit load analyses.

are placed at the forward beam locations (CASE 1), at the back beam locations (CASE 6), and at the cap center-line (CASE 2). Uniformly distributed unit loading is placed along the length of the cap (CASE 3), along the entire exposed length of the columns (CASE 4), and along the water depth on the column (CASE 5). Since the solution of the frame is elastic, the column forces may be solved independently for each of the six previously mentioned load types. Each of the six solutions of the frame provides coefficients which are used in computing and summing the final axial force and moments for all columns. Three subroutines formatted similar to the three associated with the approximate analysis utilize the coefficients computed with the six unit frame analyses. As with the approximate analysis, the ACI moment magnifier is applied to the computed bending moments to obtain design moments. Conservatively, the largest axial load for a given frame is used in computing the moment magnifier for all columns.

The final result of the frame analysis is a set of column forces which better represents the actual behavior of the bent cap and column system. Although the matrix solution would be very tedious and time-consuming by hand, the automated computer approach presented in this research allows the design engineer to perform a sophisticated frame analysis in a very short period of time with no additional effort.

2.2 DOCUMENTATION OF BENTCOL

This section provides an input guide and user information, definition of input variables, and a description of the output files generated by **BENTCOL**. The program offers the choice of two analysis procedures which utilize the same input data file. The format of the input data file is also presented.

2.2.1 Input Guide and User Information

The program **BENTCOL** computes the axial load and associated moments of bridge piers following the TSDHPT procedure and AASHTO specifications for bridge bents with two or more columns. As established by current office practice, AASHTO Groups I, II, and III are utilized in computing the critical column loads. The user has two choices concerning the analysis procedure when using **BENTCOL**. The approximate analysis, selected by setting ICODE = 1, follows the current TSDHPT approach. A frame analysis, selected by setting ICODE = 2, treats the bent cap and column system as an integral frame. The arrays in the frame analysis option are dimensioned to allow ten-column bridge bents. The program may be easily modified to allow for more columns if necessary. The limit of ten columns represents a practical number of columns for the types of Texas highway bridge bents typically treated by this approach. No limit exists with the approximate analysis. Note that when the second option is selected, both the approximate analysis and the frame analysis are performed. The output files produced by each of the analyses are explained in section 2.2.3, **Description of Output**.

The IAXIS input variable determines the type of bending that is allowed in a particular frame. IAXIS = 3 represents the typical case of biaxial column bending. However, certain frames may be treated by the engineer as either braced in-plane or out-of-plane. If IAXIS = 1, only in-plane moments are computed (i.e., the system is braced against out-of-plane bending). If IAXIS = 2, only out-of-plane moments are computed (i.e., the system is braced against in-plane bending). A typical example for IAXIS = 2 is the case of a system that is located in a stream and is in-filled between all columns to prevent the accumulation of debris.

The DEFL2 input variable allows the engineer to assign an out-of-plane column deflection limit. The valued is used to compute a base moment associated with the specified deflection limit. The computed deflection limit moment is compared with the moments obtained following AASHTO. If the deflection limit moment is not larger than the moment computed following AASHTO, the value output is the AASHTO moment. Set DEFL2 = 0 when no deflection moment computation is desired. This input variable is included at the request of the TSDHPT in order to provide the design engineer with the option of approximating temperature-induced moments or the option of establishing practical deflection limits for a particular structure.

Certain input variables which may seem redundant are used as control variables in the program. The program will compute loads for circular or rectangular columns. If the column is circular, set the column dimensions D1 and D2 equal to zero. Likewise, if the column is rectangular, set the column radius R equal to zero. The degree of curve CRV is set equal to zero for straight bridges. Similarly, the stream velocity V is set equal to zero in the absence of design stream flow.

The format for the input data file is FREE FIELD. The input variables on each line must be separated by a comma or space. Specific column locations for each input variable are NOT necessary. The type (Real, Integer, or Character) of each input variable and the general form of the input data file are shown below. Figures 6, 7, and 8 graphically illustrate most of the input variables which are used to describe the geometry of the bridge bent.

INPUT VARIABLE TYPES:

• Integer: ICODE, IAXIS, NOLAN, NORAL, NOBMF, NOBMB, NOCOL, INUMB, ISIZE, JNUMB, JSIZE

• Character: CARD1, CARD2, CARD3

• Real: All Other Input Variables

FORMAT OF INPUT DATA FILE:

ICODE							
CARD1							
CARD2							
CARD3							
IAXIS	**NOLAN**						
PHI	**NORAL**	**RALHT**					
WRAIL	**SLBW**						
SLBD	**CAPW**	**CAPL**					
CAPD	**WBMF**	**NOBMF**	**BMDF**	**ECCF**	**BMSPF**		
SPANF	**WBMB**	**NOBMB**	**BMDB**	**ECCB**	**BMSPB**		
SPANB	**PAD**	**HNCH**					
PDSTIFC	**NOCOL**	**COLH**	**COLSP**	**COLK**	**CM**	**FIXD**	**DEFL2**
WCOL	**D1**	**D2**					
R	**DSMPH**						
CRV	**WTRD**						
V	**HDECK**	**CMULT**					
DSWIND	**COVER**	**INUMB**	**ISIZE**	**JNUMB**	**JSIZE**		
FY							

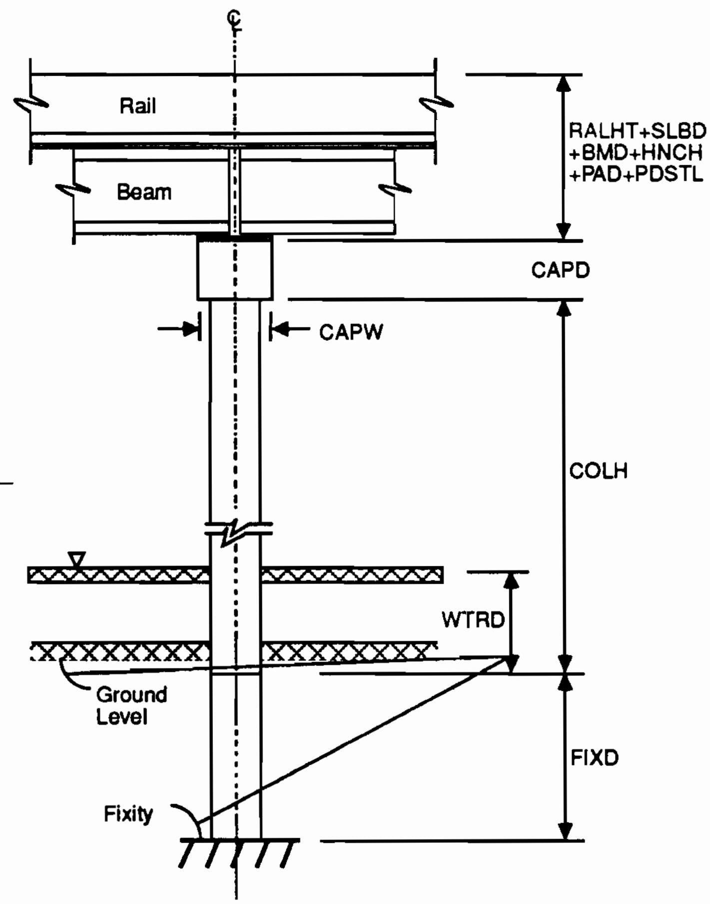


Fig 6. Input variables (elevation).

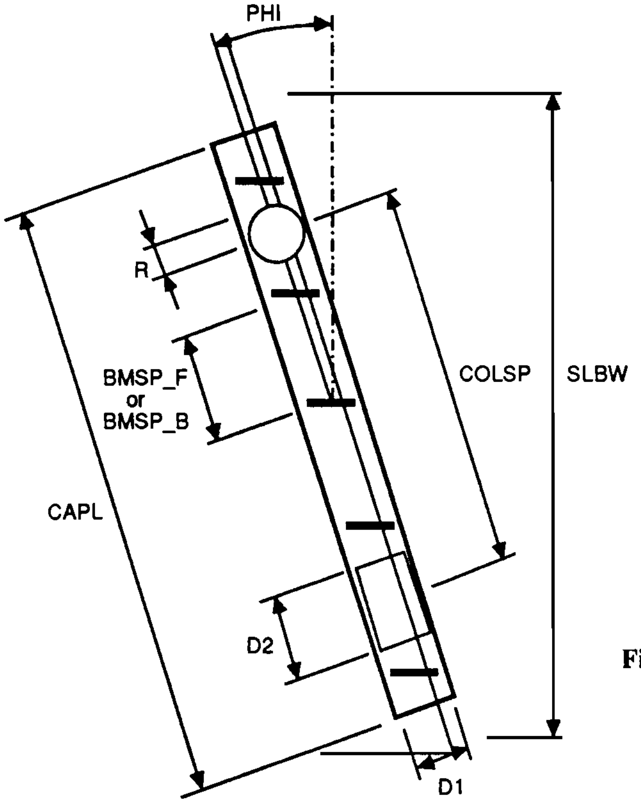


Fig 7. Input variables (plan).

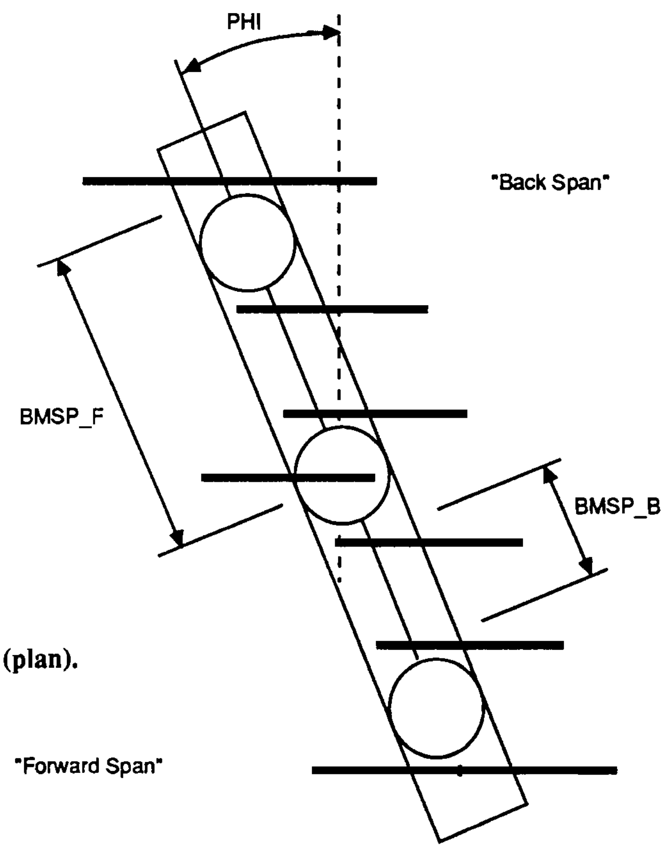


Fig 8. Input variables (plan).

2.2.2 *Definition of Input Variables*

The input variables are defined below. The input variable FORTRAN symbol is followed by a descriptive definition of the variable as well as the required units. The AASHTO Article number associated with certain input items is also listed where appropriate.

BMDB =depth of longitudinal beams, back span (feet)
BMDF =depth of longitudinal beams, forward span (feet)
BMSPB =average beam spacing measured along cap, back span (feet)
BMSPF =avg. beam spacing measured along cap, forward span (feet)
CAPD =depth of cap (feet)
CAPL =length of cap measured transverse to roadway (feet)
CAPW =width of cap measured parallel to roadway (feet)
CARD1 =text description of problem
CARD2 =text description of problem
CARD3 =text description of problem
COLH =height of column (feet)
COLK =effective length factor for compression members (AASHTO Article 8.16.5.2.3)
COLSP =column spacing measured along cap (feet)
COVER =uniform cover over steel reinforcing bars (inches)
CM =1.0, a factor relating actual moment diagram to an equivalent uniform moment diagram (AASHTO Article 8.16.5.2.7)
CMULT =continuity multiplier for wind on superstructure
CRV =degree of curve (AASHTO Article 3.10.1) (degrees)
D1 =depth of rectangular column measured along cap width (feet)
D2 =depth of rectangular column measured along cap length (feet)
DEFL2 =out-of-plane deflection limit for computing deflection moment (inches)
DSMPH =vehicle design speed (AASHTO Article 3.10.1) (mph)
DSWIND =design wind velocity for wind reduction (AASHTO Article 3.15) (mph)
ECCB =bearing pad eccentricity, back span (inches)
ECCF =bearing pad eccentricity, forward span (inches)
FC =specified compressive strength of concrete (psi)
FIXD =depth to fixity for column (feet)
FY =yield stress of steel reinforcing bars (ksi)
HDECK ="height above deck" (AASHTO Article 3.10.3) (feet)
HNCH =depth of haunch (feet)
IAXIS =code for bending axis option in PCA2 program
ICODE =code for approximate or frame analysis procedure
INUMB =minimum number of steel reinforcing bars
ISIZE =minimum size of steel reinforcing bars
JNUMB =maximum number of steel reinforcing bars
JSIZE =maximum size of steel reinforcing bars
NOBMB =number of longitudinal beams, back span
NOBMF =number of longitudinal beams, forward span
NOCOL =number of columns
NOLAN =number of traffic lanes
NORAL =number of rails
PAD =depth of beam bearing pad (feet)
PDSTL =depth of pedestal (feet)
PHI =skew angle of bent (degrees)
R =radius of circular column (feet)
RALHT =height of rail (feet)
SLBD =depth of roadway slab (feet)
SLBW =width of roadway slab(feet)
SPANB =length of back span (feet)
SPANF =length of forward span (feet)
V =velocity of water (AASHTO Article 3.18.1) (fps)
WBMB =weight per foot of beam, back span (k/ft)
WBMF =weight per foot of beam, forward span (k/ft)
WCOL =weight per foot of column (k/ft)
WRAIL =weight per foot of rail (k/ft)
WTRD =design depth of stream flow (feet)

2.2.3 *Description of Output*

The program produces up to four output files. When the approximate analysis is selected, only two files are produced. When the frame analysis is selected, two additional files are produced. The "*filename*.OU1" contains the detailed step-by-step echo of the intermediate computations throughout the approximate solution. The "*filename*.OU2" contains a summary of the results from the frame analysis. These files are intended to be used for review of the analysis procedure by the design engineer. The output files "*filename*.IN1" and "*filename*.IN2" are used directly as the input data files for the column design program **PCA2**. These files are in the format needed by the current **PCA2** column design program. A summary of the output files is presented at the end of this section. If **PCA2** is modified or updated, the program **BENTCOL** may need to be modified to reflect appropriate changes in the input format required by **PCA2**. The portion of **BENTCOL** which generates the input data file for **PCA2** is located in the main program. Separate, but very similar procedures are clearly labeled for the approximate analysis and the frame analysis. The format statements are grouped with their respective write statements in each of the two sections. The amount of input data other than the column axial loads and associated moments is very limited. Thus, any changes that may be necessary should be minor.

The units of column axial load are kips. Bending moments are output in units of kip-feet. These units are expected by the **PCA2** column design program.

- **Approximate Analysis (ICODE = 1)**
  - *filename*.OU -Long form showing intermediate computations.
  - *filename*.IN1 -Short form used as approximate input for **PCA2**.
- **Frame Analysis (ICODE = 2)**
  - *filename*.OU1 -See above.
  - *filename*.IN1 -See above.
  - *filename*.OU2 -Summary of frame analysis computations.
  - *filename*.IN2 -Short form used as frame input for **PCA2**.

2.3 EXAMPLE PROBLEMS

The following two sections demonstrate typical problem solutions using two typical TSDHPT bridges. The input data file for **BENTCOL** is included for each problem. The four output files from **BENTCOL** associated with Example Problem 1 and Example Problem 2 are listed in APPENDIX A and in APPENDIX B, respectively. Also, the output files from the column design program **PCA2** are included for both problems. These two problems serve as the basis for the general remarks and comparisons presented in Section 2.4 at the end of this chapter. The two example bridge structures are very similar. Example Problem 1 presents a two lane structure and Example Problem 2 presents a three lane structure. Most of the bridge data between the two example problems overlaps in order to simplify the following demonstration.

2.3.1 Example Problem 1

The design engineer begins the design process of a typical bridge column with the information presented below. See Fig 9 for a sketch of the bridge bent. These values formulate the majority of the input variables required for analyzing the bent cap and column system. The design engineer is responsible for assigning the additional input required, relying on established TSDHPT values and exercising good engineering judgement.

<u>GIVEN</u>: 25° Right forward skew on 2° Curve
Forward Span ≈ 60'; Back Span ≈ 69' — — 64.5' avg.
DL Rail - (T5) ≈ 0.330 k/ft
Slab Width ≈ 25' (2 lanes)
Bm. Spacing (forward & back) ≈ 3 spaces @ 6.800/cos25 =22.5'
Bm. Depth (forward & back) ≈ 3.333'
Slab Thickness = 7.500" = 0.625'
DL Bm. (forward & back) ≈ 0.516 k/ft
Cap Length ≈ [3(6.800) + 1.75(2)]÷cos25 = 26.37' ≈ 26.5'
Cap Dimensions = 2.75' square
Max. Wind Velocity = 80 mph
Design Stream Velocity ≈ 6.0 fps
Design Stream Flow Depth ≈ 22'
Column Height = 22'
Column Spacing = 16'
Use 2 - 30" dia. columns
Depth to fixity of column = 4'

The format of the input data file is shown in Section 2.2.1. For the purposes of this demonstration, both the approximate and the frame analysis are performed by setting ICODE = 2. Also, biaxial bending is selected by setting IAXIS = 3. The data file used for the actual analysis is presented under <u>INPUT DATA FILE FOR EXAMPLE PROBLEM 1</u>.

<u>INPUT DATA FILE FOR EXAMPLE PROBLEM 1:</u>

```
2
JOB NUMBER:   2.3.1
DESCRIPTION:  2-column Bent, 4 Beams, 2 Lanes
LOCATION:     Outahere,Tx                DATE:1 December 1987
3
25.  2
.330 2    2.67
.625 25.
2.75 2.75 26.5
60.  .516 4      3.333    0.     7.5
69.  .516 4      3.333    0.     7.5
.125 .083 .083
.736 3600. 2  22     16    1.25   1.0     4.  0.
1.25 0.   0.
2.   55.
6.   22.
80.  6.   1.
40.  2.   8   9      14    11
```

The column design program **PCA2** was executed using INPUT.PCA and FRINPUT.PCA. The output from the two runs, OUTPUT.PCA, is presented in APPENDIX A, sections A.1.3 and A.2.3, respectively. The output from **PCA2** includes an echo print of the design forces as well as the ratio of column capacity to column design load. These output files will be utilized in Section 2.4 in the discussion comparing the approximate analysis and the frame analysis. The size and number of steel reinforcing bars computed for each analysis procedure are presented on the following page.

PCA2 OUTPUT (Approximate Analysis:)

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	14	TOTAL AREA (SQ IN.)	17.78
BAR SIZE NO.	10	REINF RATIO (DECIMAL)	0.02515

PCA2 OUTPUT (Frame Analysis):

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	11	TOTAL AREA (SQ IN.)	17.16
BAR SIZE NO.	11	REINF RATIO (DECIMAL)	0.02428

2.3.2 Example Problem 2

The three column bent presented in Example Problem 2 is shown in Fig 10. As previously mentioned, most of the input for this example is the same as that for Example Problem 1 for purposes of simplifying the discussion. The unique input variables are listed below, followed by the actual input data file used in the analysis.

GIVEN: Slab Width ≈ 40' (3 lanes)
Bm. Spacing ≈ 5 spaces @ 6.800/cos25 = 37.5'
Cap Length ≈ [5(6.800) + 1.75(2)]÷cos25 = 41.37' ≈ 41.5'

INPUT DATA FILE FOR EXAMPLE 2:

```
2
JOB NUMBER: 2.3.1
DESCRIPTION: 3-column Bent, 6 Beams, 3 Lanes
LOCATION:    Outahere,Tx          DATE:1 December 1987
3
25.   2
.330  2     2.67
.625  40.
2.75  2.75  41.5
60.   .516  6        3.333    0.     7.5
69.   .516  6        3.333    0.     7.5
.125  .083  .083
.736  3600. 3    22     16    1.25   1.0    4.   0.
1.25  0.    0.
2.    55.
6.    22.
80.   6.    1.
40.   2.    8 -  9      14    11
```

The column design program **PCA2** was executed using INPUT.PCA and FRINPUT.PCA. The output from the two runs, OUTPUT.PCA, is presented in APPENDIX B, sections B.1.3 and B.2.3, respectively. The output from **PCA2** includes an echo print of the design forces as well as the ratio of column capacity to column design load. These output files will be utilized in Section 2.4 in the discussion comparing the approximate analysis and the frame analysis. The size and number of steel reinforcing bars computed for each analysis procedure are presented on page 15.

2.4 GENERAL REMARKS AND COMPARISONS

The output from the PCA2 column design program indicates that the approximate analysis procedure is slightly conservative for the two example problems. The bent columns in Example 1 according to the approximate analysis require 17.78 square inches of steel compared to the 17.16 square inches with the frame analysis. The bent columns in Example 2 according to the approximate analysis require 12.0 square inches of steel compared to the 11.43 square inches with the frame analysis. Examples 1 & 2 represent typical Texas highway bent geometries for which the approximate method yields reasonable final results. However, recall that this method is using an axial force which is underestimated and an in-plane bending moment which is overestimated. Although the similarity between the final results is apparent, it must be noted that frame analysis presents a more consistent and rational approach to analyzing bent columns. Using the frame analysis, the design engineer can confidently select the reinforcing steel based on actual design axial forces and bending moments.

As the bent geometry deviates from "typical" geometries, the uncertainty relating to the approximate method increases. The amount that the in-plane bending moment is overestimated by the approximate method is very unpredictable. For typical bents, the higher bending moment counteracts the underestimated axial force. The rationale is lacking, but the final results are apparently satisfactory. However, bridge bents which do not conform to typical situation do not behave according the the approximate method. For example, some bridge bents do not have the cantilever overhang demonstrated in Examples 1 & 2. Without an overhang, a higher percentage of the axial load is distributed to the interior columns. For comparative purposes, two bridge bents similar to Example 2 (6-beams, 3-columns @ 16' spacing) were analyzed. The first bent, BENT-A, had a cap length of 32 feet, i.e. no overhang. The second bent, BENT-B, had a cap length of 36 feet, i.e. 2' overhang, conforming to the typical geometry. The ratio of

PCA2 OUTPUT (Approximate Analysis):

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	12	TOTAL AREA (SQ) IN.)	12.00
BAR SIZE NO.	9	REINF RATIO (DECIMAL)	0.01698

PCA2 OUTPUT (Frame Analysis):

THE REINFORCEMENT PATTERN OF THE DESIGN IS

NO. OF BARS	9	TOTAL AREA (SQ IN.)	11.43
BAR SIZE NO.	10	REINF RATIO (DECIMAL)	0.01617

the frame analysis axial load to the approximate analysis axial load was computed. A ratio of approximately 1.40 was computed for BENT-A. A ratio of approximately 1.20 was computed for BENT-B. The magical balance of higher bending moment and lower axial force is lost as the geometry varies from the typical. The design engineer may now remove the uncertainty associated with classifying a bridge bent as typical or atypical by selecting a frame analysis which accurately produces design axial forces and bending moments.

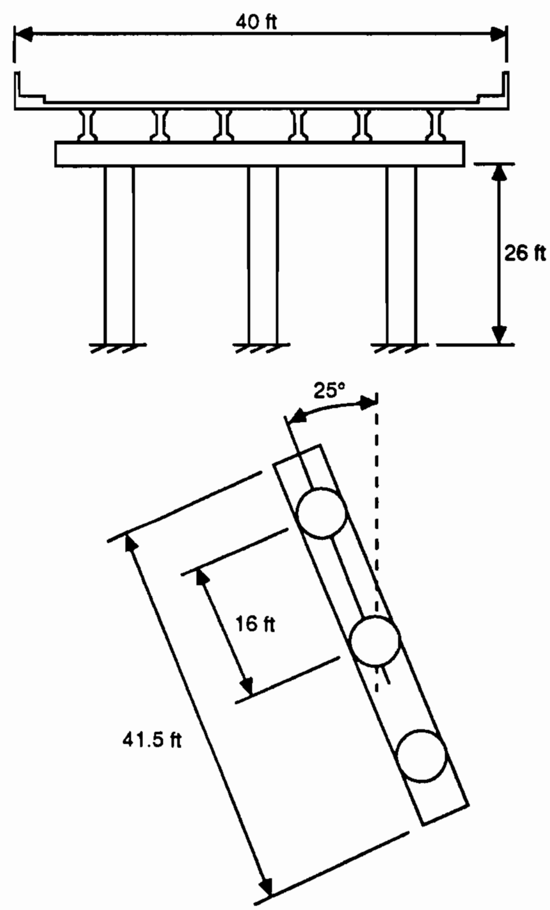


Fig 10. Example problem 2.

CHAPTER 3. PASCAL GRAPHICAL INTERFACE BCINPUT

3.1 GENERAL CONSIDERATIONS

The handling of input arrangements is of primary importance with any computer-aided analysis and design package. Even the most sophisticated analysis and design application computer programs are heavily dependent on the user's ability to understand and create the necessary input data files. A well-written application program may quickly become an unattractive design tool for the engineer if the input arrangements are confusing and difficult to understand. To complete a useful design package, a user-friendly, self-explanatory link between the computer and the user is desirable. In view of the importance of getting the proper information to an application program, the use and the development of a graphical interface was investigated in this research. The following sections discuss some alternatives to a graphical interface and the actual graphical interface developed with this research.

3.2 ALTERNATIVES TO A GRAPHICAL INTERFACE

Alternatives to a graphical interface include the use of batch input, the use of an interactive prompt mode, and the use of a spreadsheet arrangement such as LOTUS 1-2-3. Historically, batch input format has dominated input arrangements in most engineering application programs. Using batch input creates what is perhaps the least desirable environment for user-interaction with an application program, especially for a first-time user of the program. Typically, the user must create the batch input data through the use of an editor of some sort. The result is a collection of numbers and text oriented in a fashion understood only by the application program. To the user, the collection is sometimes nothing more than a confusing list of input items. The very nature of the batch input mode implies an unfriendly user environment in which modifications and readability are difficult.

A slightly friendlier approach is found through the use of a user-interactive prompt mode. By prompting the user for input information, the application program becomes somewhat less mysterious as the user actually interacts with the computer. Typically, the user is prompted by a message sent to the screen to input a particular variable and enter return. Following a series a messages, the user completes the necessary input data required by the application program. Although the prompt mode may be considered an improvement to the batch environment described earlier, several drawbacks and criticisms can be cited. For example, the user is typically given very little freedom of choice in the progression of events. The prompting usually follows a regimented progression which is not controlled by the user. Another problem the user faces concerns correcting and modifying the input data. A program may prompt for several input items and a mistake may occur near the end of process. Often, to correct this mistake the user must repeat the entire process to eventually come to the location of the incorrect value. The difficulty in modifying input data is considered a major problem, often lessening the appeal of the prompt mode. A final consideration is the difficulty in reviewing the final collection of input data. In the prompt mode, once a value is input it is not seen again until the input data is echoed upon running the application program. Finding an error at this stage of the design process is very inconvenient and frustrating to the user. An option to review the input items in a collective fashion associated with a short description of the variable is desirable.

The use of a spreadsheet arrangement such as LOTUS 1-2-3 alleviates many of the problems associated with the batch mode and the prompt mode. With a spreadsheet arrangement, the user may move freely between all input variables. Also, the input fields accepting input values may be located with text descriptions. Associating a descriptive name in conjunction with the input field for the value greatly simplifies the task of the user, providing a logical relationship between each input variable and allowing for quick modifications. The use of a spreadsheet arrangement was given substantial consideration for use in this research. A primary reason against the selection of spreadsheet arrangement relates to the dependency of the input interface on the spreadsheet software. In addition to the cost of the software, another detriment is the need for the user to be an experienced user of the particular spreadsheet software on which the interface may be developed. A more attractive alternative might be an interface which is independent, that is, one which requires no additional software. Finally, using a package such as LOTUS 1-2-3 restricts the ability to create input screens which use input fields in association with a graphical representation of the problem. Relating an input variable with a graphical representation of the problem would greatly simplify use of an application program.

3.3 GRAPHICAL INTERFACES

A graphical interface combines text, graphics, and user events, creating a user-friendly and self-explanatory link between the computer and the user. The interface application is characterized as being "Event Driven", meaning that the user is in control of the actions that the computer takes at any given time. In this environment, the user's actions (events) are monitored by a main infinite event loop which responds to keystrokes by taking an appropriate action. This environment is based on user-driven menus which allow the user to intuitively take appropriate actions at any time. Upon completion of a selected task, control is returned to the main event loop as the computer waits for additional actions by the user.

With the interface, selection of particular menu items is easily accomplished by entering the key letter of the drop-down menu items. No command sequences, which typically represent a burden to the user, are necessary. In a menu-driven environment, the user, not the computer, is in complete control. An added feature of an interface is the ability to anticipate "bad" actions by the user. For example, trying to save a file in which the input is incomplete should be accompanied by an alert to the user. Warning the user of an unusual event is much more appealing than the error message typically associated with improper actions. However, another primary goal of a good interface is to always provide the user with alternative choices, avoiding dead-ends and traps which restrict the user's control. For example, the user may be fully aware that the input data is incomplete and still wish to save the data file. Certainly the choice to save a portion of the file should be an option available to the user; the interface should warn but not prevent the user from taking such an action. Additional features of the graphical interface developed in this research are demonstrated through figures in detail in Section 3.4.

A graphical interface can effectively combine graphics screens with appropriate input data fields to aid the user in defining and understanding a particular problem. Although it is helpful to have a text description associated with an input field, a more desirable situation is to place the input field on a graphical image which clearly represents and explains the input variable. Incorporating graphical input screens into the user interface creates an attractive and easy-to-use tool for handling input arrangements for an application program. Supplying the design engineer with an effective way to create the input data for an application program completes the analysis and design process nicely, allowing the engineer to quickly understand and modify input data.

3.4 DESCRIPTION OF BCINPUT

The graphical interface **BCINPUT** was developed in order to complete the analysis and design microcomputer package for Texas highway bridge bent columns. The interface provides the design engineer with the attractive alternative of using a stand-alone microcomputer approach to the problem. Using **BCINPUT**, the designer may quickly transition from the creation of the input data file to the analysis and design of bent columns. Removing the difficulty typically associated with creating input data files for engineering application programs allows the engineer to devote more time to reviewing and interpreting results. Furthermore, modifications to the input data file upon detection of mistakes are very easily made, allowing the entire design process to be repeated in a short period of time.

The graphical interface was developed using Bordland International's TURBO PASCAL and GRAPHIX TOOLBOX. The machine selected was the IBM-AT personal computer. The Pascal Programming Language was chosen over FORTRAN due to the necessity of developing an event-driven interface which supports screen formatting and high resolution graphics. The EGA monitor with the IBM-AT supports color and high resolution graphics (640 x 350 pixels). Using TURBO and the IBM-AT allowed the development of a graphical interface which effectively uses color to create an attractive and friendly user interface.

Figure 11 on the following page presents the "flow" of **BCINPUT**. The diagram represents the overall structure of the interface, indicating the options available to the user upon selection of each menu item. Note that the diagram is provided to demonstrate the overall structure of the interface. The user need not be concerned with this structure in order to use the interface. After the procedure corresponding to a particular action of the user is completed, control is returned to the main infinite loop as the computer awaits additional events from the user. As mentioned previously, the user is in complete control of the machine.

The drop-down selections associated with each menu item are demonstrated in Fig 12. The user may freely move from field to field by using the left and right arrow keys. The "active" menu changes color and the drop-down selections appear as the arrow keys are pressed. The items listed in each menu are selected by pressing the first letter of the key word which is shown in the color red. The remaining letters in the word are shown in white, and the background color is blue, demonstrated in the figures by the lines surrounding the text.

All file handling is available under the menu item **"FILES"**, including the **Q**uit selection which results in the return to the system prompt. Figure 13 demonstrates the screens associated with "FILES". "New" is chosen when the user desires to open or create a new file. If the file already exists, the user is alerted accordingly to avoid accidental re-initialization of the input variables. The possible options available upon this occurrence are shown in the figure. "Open" is chosen when the user desires to open an existing file. If the file is not found, the user is greeted by an alert message and options rather than a system crash. Finally, "Save" writes the input data to a file in the format accepted by the FORTRAN column analysis program **BENTCOL**. Warning messages associated with "Save" are also shown in the figure. Once the data is saved, the user may exit **BCINPUT**, modify the input data, or create a new data file.

The screens associated with **"OPTIONS/DESCRIPTION"** are shown in Fig 14. The screens associated with **"INPUT SCREENS"** are shown in Figs 15 and 16. Finally, the screen associated with **"HELP"** is shown in Fig 17. As indicated on the HELP screen, cursor movements are controlled by the arrow, tab, shift-tab, backspace, and enter keys. All windows are closed by pressing the **"F1"** function key. Note that each screen must be closed before entering another window. Closing each window returns control to the main event loop. Finally, the naming of files must follow standard DOS filename procedures, beginning the first space of the input field.

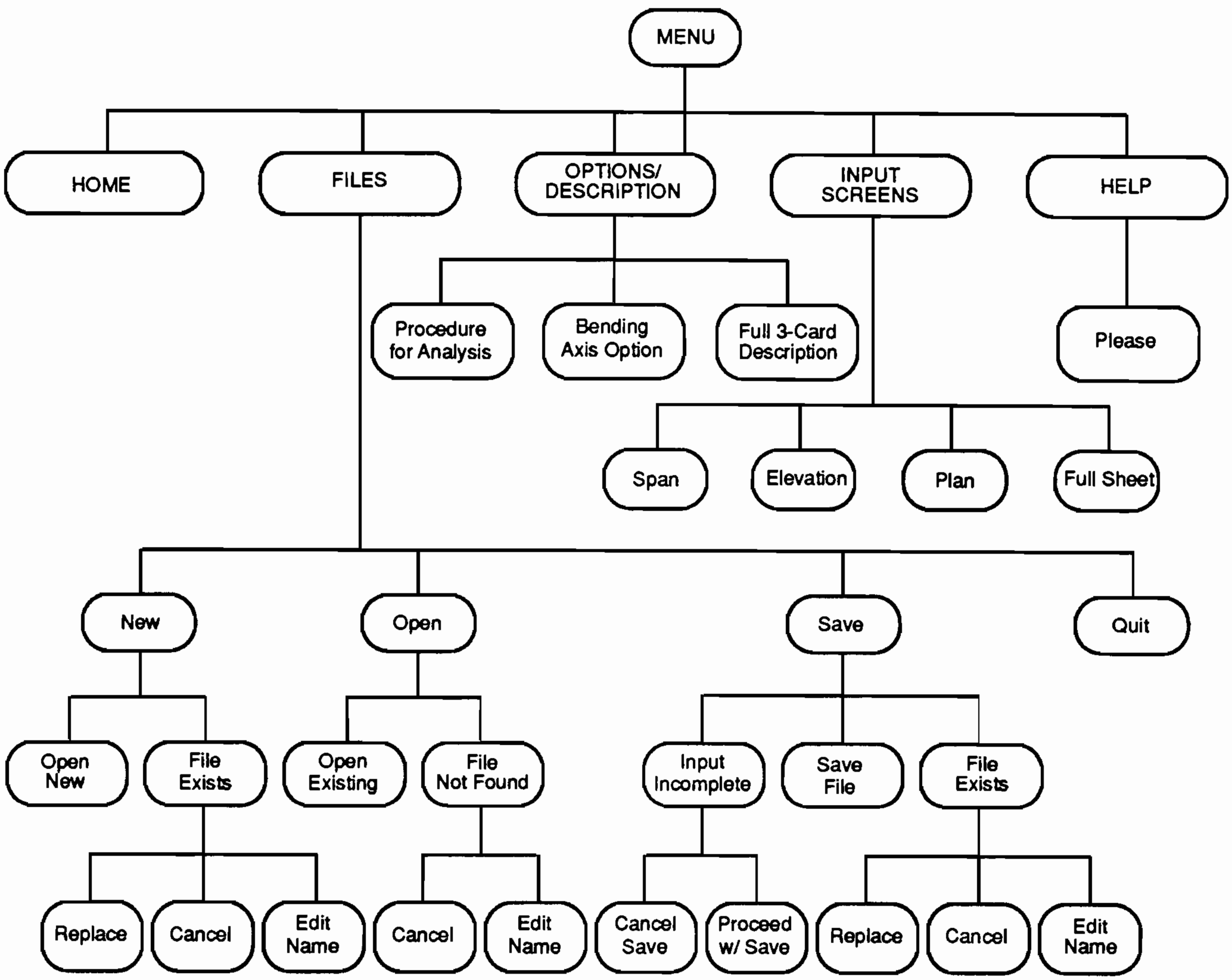


Fig 11. Flow of BCINPUT.

Fig 12. Menu screens.

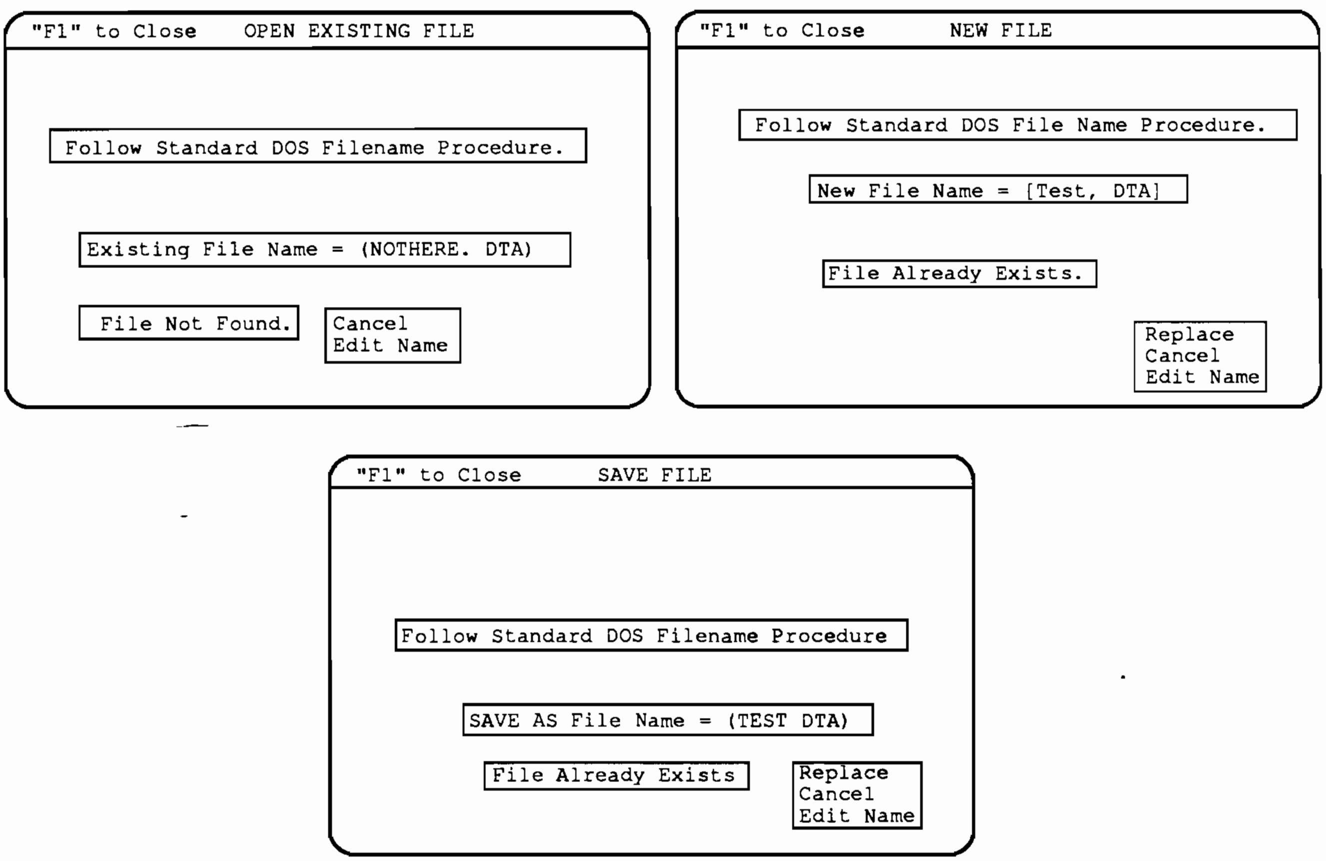


Fig 13. "Files" screens.

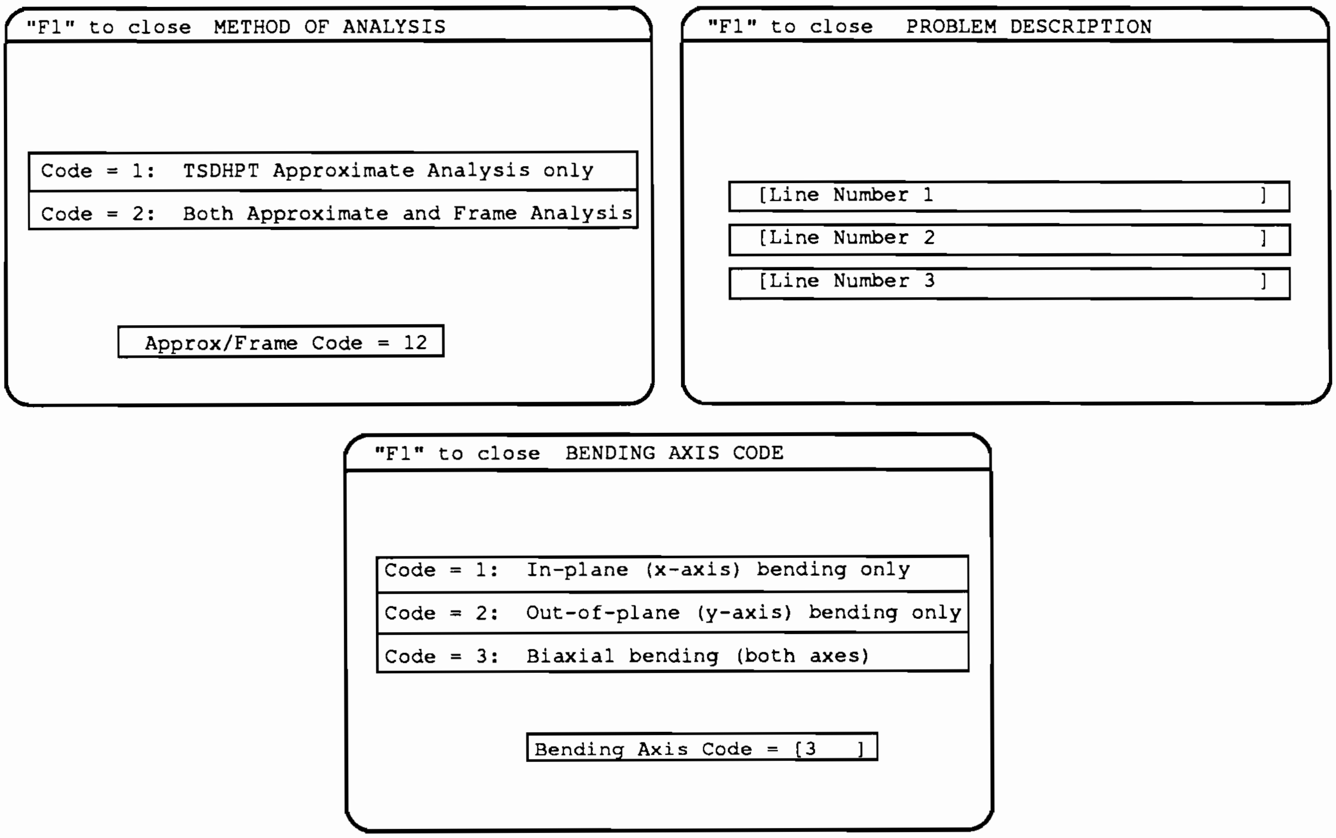


Fig 14. "Options/Description" screens.

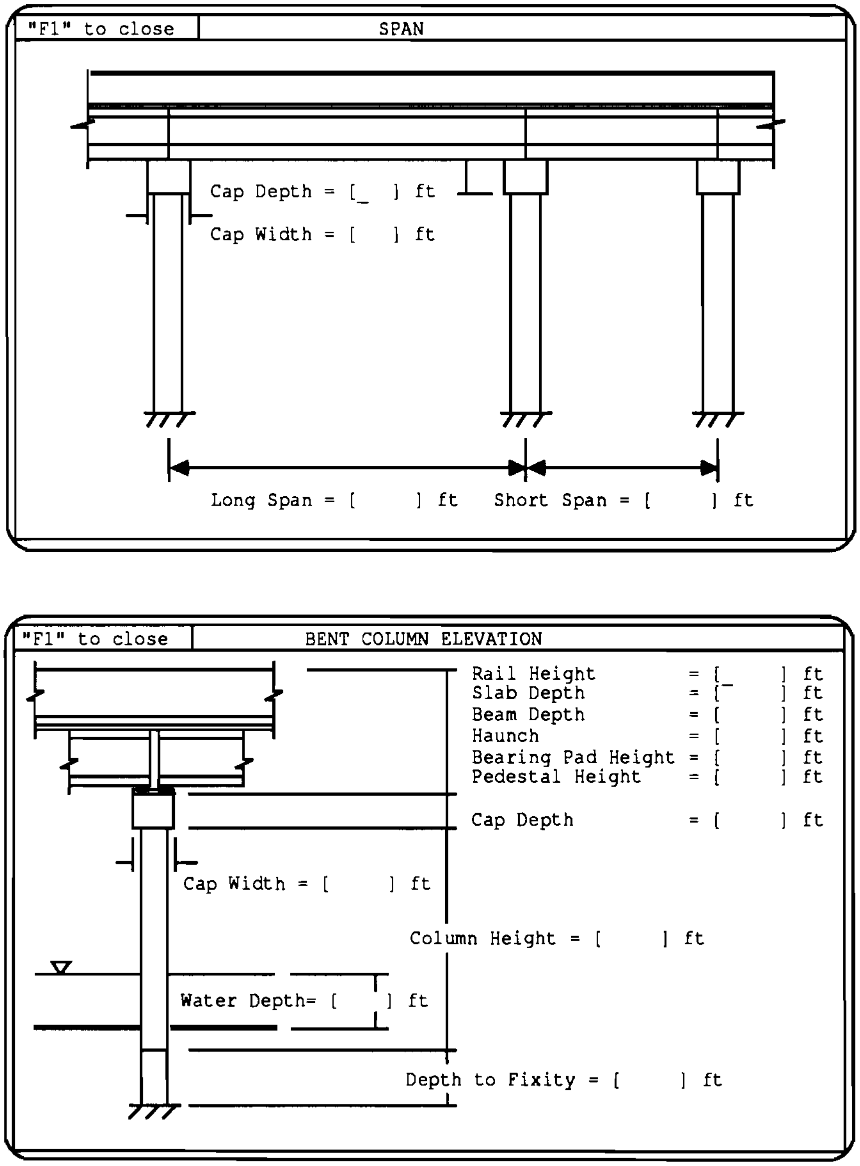
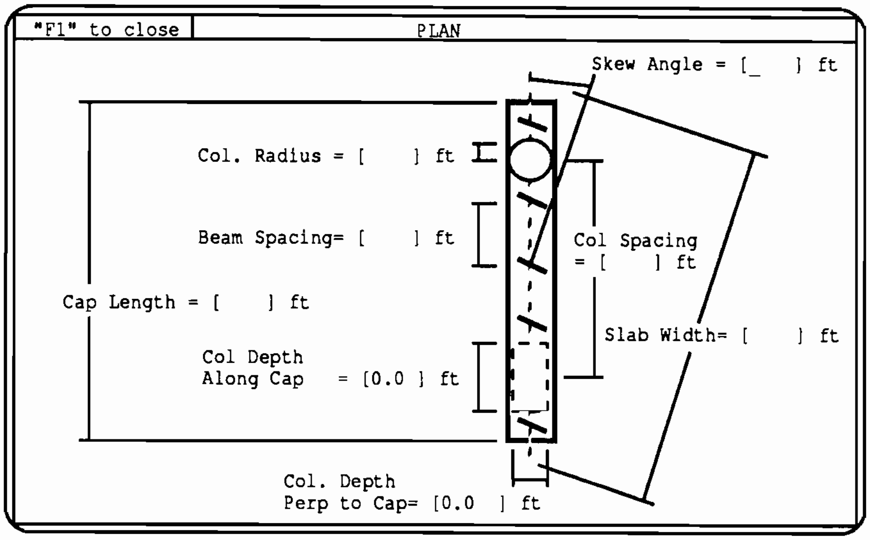


Fig 15. Span and elevation input screens.



Number of Lanes	=[ ]	
Span Length, Long	=[ ]	ft
Span Length, Short	=[ ]	ft
Skew Angle	=[ ]	deg
Number of Rails	=[2]	
Wt/ft of Rail	=[ ]	k/ft
Rail Height	=[ ]	ft
Number of Beams	=[ ]	
Wt/ft of Beam	=[ ]	k/ft
Beam Depth	=[ ]	ft
Beam Spacing	=[ ]	ft
Number of Columns	=[ ]	
Wt/ft of Column	=[ ]	k/ft
Column Height	=[ ]	ft
Depth to Fixity	=[ ]	ft
Column Spacing	=[ ]	ft
Column Radius	=[ ]	ft
Col. D Perp to Cap	=[0.0]	ft
Col. D Along Cap	=[0.0]	ft
Fc of Concrete	=[3600.]	ft
Column K-factor	=[1.25]	
Cm	=[1.0]	

Slab Depth	= [ ]	ft
Slab Width	= [ ]	ft
Cap Depth	= [ ]	ft
Cap Width	= [ ]	ft
Pedestial Height	= [0.125]	ft
Bearing Pad Height	= [0.083]	ft
Haunch	= [0.083]	ft
Velocity of Stream	= [80.0]	fps
Water Depth	= [ ]	ft
Degree of Curve	= [ ]	deg
Design Speed	= [55.0]	mph
Texas Design Wind	= [80.0]	mph
AASHTO Ht Above Dk	= [6.0]	ft
Bent Cont. Mult.	= [1.0]	ft
Out-of plane Defl.	= [0.0]	in
Approx/Frame Code	= [2]	
Bending Axis Code	= [3]	
Name of Input File	= [TEST.DTA]	

"F1" to Close

Fig 16. Plan and full sheet input screens.

The graphical interface presented in this section represents one approach to the handling of input data files. Figure 1 of Chapter 1 demonstrates how the graphical interface **BCINPUT** fits into the overall column design process. A more detailed discussion is presented here to explain the use of a PASCAL user interface in creating the input data file for a FORTRAN application program. From the main infinite event loop, the interface writes the input data field such as "LONG SPAN = [_] ft" to the screen. The computer detects any keystrokes by the user and places the ASCII characters in the input field. At this stage, it is interesting to note that the characters input are not yet "real" numbers. All the data input by the user is read by the computer as text, including letters, numbers, and other symbols such as "+", ".", and "-". Once a particular input field is complete, the user proceeds to the other input fields until all required input is completed. Upon the completion of an entire screen, the user selects "F1 to close", allowing the computer to store the ASCII characters for each input field in a single array containing all the input data. Once all the required input is completed, the user must save the data file. Saving the file simply writes to disk all the text entered and stored in the total input array. The data is written to the filename specified by the user in the format required for the column analysis program **BENTCOL**. Stated simply, the interface functions primarily as a tailored editor for creating the input data file for the application program. A similar batch text input data file could be created using any word processing software. Upon exiting **BCINPUT** by selecting **"Quit"**, the user may choose to execute the column analysis program **BENTCOL**. The user is prompted for the name of the input data file created using **BCINPUT**. The FORTRAN program opens the specified file and reads the data to be used in the analysis. Upon reading the input data and assigning the values to variables in the program the ASCII data is finally interpreted as numbers by the computer and the column analysis may proceed.

3.5 GENERAL REMARKS

The listing of the PASCAL program **BCINPUT** is presented completely in APPENDIX D. Note that the version discussed herein represents the initial graphical interface developed in this research. The final version presented to the TSDHPT was adapted and improved, but not included in this thesis due to time constraints. Modifications include some string checking and additional checks relating to the handling of files. Also, default input variables which are constant for most problems were removed from the **"Full Sheet"** input screen and placed on a separate screen for improved readability.

It is important to note the broad application of graphical interfaces to several engineering problems. The creation of input data files is often the biggest obstacle preventing the design engineer from effectively using an application program. The graphical interface developed in this research is intended to demonstrate the potential and need for improved methods of handling input data. Also, the intent is to suggest that it is possible to develop a user-friendly environment on IBM-type personal computers through the use of an event-driven graphical interface.

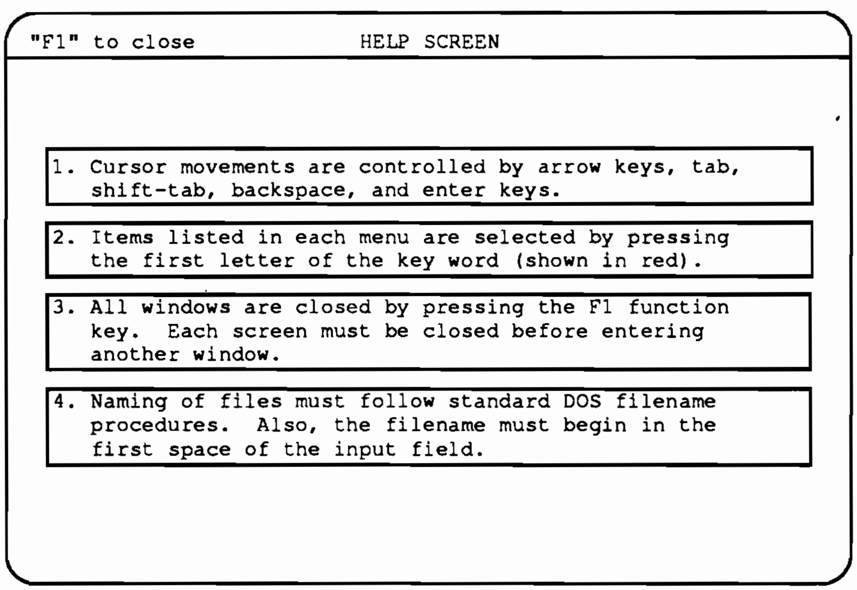


Fig 17. Help screen.

CHAPTER 4. CONCLUSIONS AND RECOMENDATIONS

4.1 CONCLUSIONS

The column analysis computer program **BENTCOL** developed at The University of Texas at Austin allows the design engineer to quickly determine design forces needed for the design of TSDHPT bridge columns. The program includes two analysis options, including an approximate analysis and an integral frame analysis of the bent cap and column system. The approximate method is primarily a computerized version of the TSDHPT current office procedure which uses AASHTO Groups I, II, and III in the determination of column axial loads and bending moments. The integral frame analysis developed in this research was used to investigate the accuracy of the current office procedure. The frame analysis option is also available to the design engineer, allowing a better analysis of the bridge columns. This option requires fewer assumptions than the approximate method and produces column design forces which better represent the actual behavior of the bridge columns. An attractive feature of the frame analysis is the automatic generation of additional information necessary for the matrix solution to the indeterminate structure. The design engineer may choose either, or both, analysis procedures using identical input data files.

Along with minimized input, the analysis program also produces input data files in the exact format necessary for the column design program **PCA2**. Generating directly the input data files for the design program greatly simplifies the tasks before the design engineer, providing a direct link between analysis and design. Linking the analysis and design process provides the design engineer with a powerful and effective tool for designing bridge columns.

The graphical interface **BCINPUT** developed on an IBM-AT microcomputer provides an innovative approach for creating the input data required by the column analysis program. An effective approach for handling the input data for the analysis program completes the total design package nicely. Using the event-driven graphical interface, the user may create with ease the input data file for the analysis program in a user-friendly environment. Creating and modifying the input data files becomes a simple and routine procedure, providing the user with warnings and alerts in the event of unusual actions. The interface combines text, graphics, color, and user events, providing a user-friendly and self-explanatory link between the computer and the user. Removing the difficulty typically associated with creating input data files for engineering application programs allows the engineer to devote more time to reviewing and interpreting results.

4.2 RECOMMENDATIONS FOR FUTURE RESEARCH

Future research in the area of bridge analysis could involve modifying the column analysis program **BENTCOL** or developing a more powerful analysis tool. One area of investigation is the soil interaction of the bridge column. The current approach requires an assumption of "fixity" based on the soil properties. Another area of possible research is the investigation of second order effects of the columns. The funding for this research is currently being used to conduct a study of a non-linear frame subjected to biaxial bending. This study should provide useful information relating to the selection of column k-factors in approximating second order effects.

Future research in the area of graphical interfaces could involve improving the PASCAL program **BCINPUT**, or the use of **BCINPUT** as a model for other improved interfaces. As stated previously, the graphical interface developed in this research was intended to demonstrate the potential and need for improved methods of handling input data. The addition of a user-friendly interface greatly enhances the usefulness of any application program. The interface presented may be improved greatly by the inclusion of string checking and improved error checking relating to the handling of files. The graphical interface developed in this research should effectively demonstrate the potential for developing a user-friendly, user-driven environment on an IBM personal computer.

REFERENCES

(1) *Column Design Sheets*, Texas State Department of Highways and Public Transportation, Bridge Division, provided by Richard L. Wilkison.

(2) *Owner's Handbook*, Turbo Graphix Toolbox version 1, BORDLAND INTERNATIONAL Inc., 1985.

(3) *Reference Manual*, Turbo Pascal version 3.0, BORDLAND INTERNATIONAL Inc., 1986.

(4) *Standard Specifications for Highway Bridges*, 12th Edition, American Association of State Highway Bridges, Washington, D.C., 1977.

APPENDIX A. COMPUTER OUTPUT FOR EXAMPLE PROBLEM 1

A.1.1 APROX.OUT

```
*****************************************************************************************

     TEXAS STATE DEPARTMENT OF HIGHWAYS & PUBLIC TRANSPORTATION:  BRIDGE DIVISION

     COLUMN ANALYSIS:  GROUPS I,II, & III              (AASHTO SPECIFICATIONS)

     AUGUST 1987                                                   VERSION 1.00

*****************************************************************************************


NAME OF INPUT FILE:  EXAMPLE PROBLEM #1


JOB NUMBER:    2.3.1
DESCRIPTION:   2-column bent, 4 beams, 2 lanes
LOCATION:      Outahere, TX                        DATE:  1 December 1987

     BENDING AXIS CODE  =    3

     NUMBER OF LANES    =    2               SLAB DEPTH           =     .625  FT
     SPAN LENGTH, AVG   =   64.500  FT       SLAB WIDTH           =   25.000  FT
     FWD SPAN LENGTH    =   60.000  FT       CAP DEPTH            =    2.750  FT
     BACK SPAN LENGTH   =   69.000  FT       CAP WIDTH            =    2.750  FT
     SKEW ANGLE         =   25.000  DEG      CAP LENGTH           =   26.500  FT

     NUMBER OF RAILS    =    2               PEDESTAL HEIGHT      =     .125  FT
     WT/FT OF RAIL      =     .330  K/FT     BEARING PAD HEIGHT   =     .083  FT
     RAIL HEIGHT        =    2.670  FT       HAUNCH               =     .083  FT

     NUMBER BEAMS, FWD  =    4               VELOCITY OF STREAM   =    6.000  FPS
     BEAM SPACING, FWD  =    7.500  FT       WATER DEPTH          =   22.000  FT
     WT/FT OF BEAM, FWD =     .516  K/FT     DEGREE OF CURVE      =    2.000  DEG
     BEAM DEPTH, FWD    =    3.333  FT       DESIGN SPEED         =   55.000  MPH
     BEARING ECC, FWD   =     .000  IN

     NUMBER BEAMS, BACK =    4               TEXAS DESIGN WIND    =   80.000  MPH
     BEAM SPACING, BACK =    7.500  FT       AASHTO HT ABOVE DK   =    6.000  FT
     WT/FT OF BEAM, BK  =     .516  K/FT     BENT CONT. MULT.     =    1.000
     BEAM DEPTH, BACK   =    3.333  FT       OUT OF PLANE DEFL    =     .000  IN
     BEARING ECC, BACK  =     .000  IN

     NUMBER OF COLUMNS  =    2               STEEL YIELD STRESS   =   40.000  KSI
     WT/FT OF COLUMN    =     .736  K/FT     UNIFORM SIZE COVER   =    2.000  IN
     COLUMN HEIGHT      =   22.000  FT       LOWER LIMIT NUMBER   =   8
     DEPTH TO FIXITY    =    4.000  FT       LOWER LIMIT SIZE     =   9
     COLUMN SPACING     =   16.000  FT       UPPER LIMIT NUMBER   =   14
     COLUMN RADIUS      =    1.250  FT       UPPER LIMIT SIZE     =   11
     COL. D PERP TO CAP =     .000  FT
     COL. D ALONG CAP   =     .000  FT
```

DEAD LOAD AND LIVE LOAD

REDUCTION FACTOR	=	1.000	
DEAD LOAD PER COL	=	197.601	KIPS
LL LANE	=	59.280	KIPS
LL TRUCK	=	63.641	KIPS
IMPACT FACTOR	=	1.264	
LL+I	=	80.432	KIPS

COLUMN CAPACITY, DEFL. LIMIT MOMENT, & BEARING ECC. MOMENTS

Fc OF COLUMN	=	3600.0	PSI
MODULUS OF ELAS	=	3420.0	KSI
MOMENT OF INERTIA	=	39760.8	IN**4
K-FACTOR	=	1.25	
Cm	=	1.00	
Pc	=	3529.5	KIPS
0.7*Pc	=	2470.6	KIPS
DEFL LIM MOM PAR	=	.0	FT-KIP
DL ECCEN. MOMENT	=	.0	FT-KIP
LL ECCEN. MOMENT	=	.0	FT-KIP

GROUP I LOADING

CENTRIFUGAL FORCE (GROUP I)

C	=	7.08	C.F.	=	4.505
H Perp	=	26.00	CF Mom Perp	=	106.147
H Par	=	39.00	CF Mom Par	=	74.247

STREAM FLOW (GROUP I)

P	=	.024	SF Mom Perp	=	14.520

***************** Bd = 1.00 ****************

Pu	=	431.420	Mom Mag	=	1.212

Sum Mom Perp	=	190.054
Sum Mom Par	=	116.941

```
MINIMUM ECCENTRICITY MOMENT

    Min Mom   =  130.673
    Resultant=  223.149

CONTROLLING MOMENTS (GROUP I)

    Mom Perp =  190.054
    Mom Par  =  116.941

****************** Bd =  .75 *****************

Pu    =  367.199     Mom Mag     =     1.175

Sum Mom Perp =  184.251
Sum Mom Par  =  113.370

MINIMUM ECCENTRICITY MOMENT

    Min Mom   =  107.825
    Resultant=  216.336

CONTROLLING MOMENTS (GROUP I)

    Mom Perp =  184.251
    Mom Par  =  113.370
```

GROUP II LOADING

WIND ON SUPERSTRUCTURE (GROUP II)

```
Wind Area Per Bent =  446.28

H Perp             =   19.21
H Par              =   32.21
```

Skew	Mom Perp	Mom Par
0.	194.238	151.871
15.	160.061	172.729
30.	137.537	202.700
45.	99.213	204.455
60.	31.623	175.398

WIND ON CAP (GROUP II)

Cap Area Perp = 7.56
Cap Area Par = 72.87

H Perp = 14.38
H Par = 27.37

Skew	Mom Perp	Mom Par
0.	1.971	16.862
15.	1.666	25.647
30.	1.247	32.683
45.	.744	37.493
60.	.189	39.747

WIND ON COLUMN (GROUP II)

Column Area Perp = 55.00
Column Area Par = 55.00

H Perp = 11.00
H Par = 15.00

Skew	Mom Perp	Mom Par
0.	21.933	13.946
15.	18.538	21.212
30.	13.881	27.032
45.	8.277	31.010
60.	2.109	32.874

WIND REDUCTION (GROUP II)

Wind Red. Factor = .640

	100 MPH Wind		Reduced Wind	
Skew	Mom Perp	Mom Par	Mom Perp	Mom Par
0.	218.141	182.680	139.611	116.915
15.	180.265	219.588	115.369	140.536
30.	152.665	262.415	97.706	167.946
45.	108.234	272.958	69.270	174.693
60.	33.921	248.020	21.710	158.733

ADDITIONAL STREAM FLOW MOMENT PERP (GROUP II)

Skew	Mom Perp	Mom Par
0.	.000	
15.	.000	
30.	.639	
45.	6.243	
60.	12.411	

***************** Bd = 1.00 ****************

Pu = 256.882 Mom Mag = 1.116

Skew	Mom Perp	Mom Par
0.	202.554	169.626
15.	167.384	203.897
30.	142.684	243.664
45.	109.558	253.454
60.	49.504	230.297

***************** Bd = .75 ****************

Pu = 192.661 Mom Mag = 1.085

Skew	Mom Perp	Mom Par
0.	196.844	164.844
15.	162.665	198.149
30.	138.661	236.795
45.	106.469	246.308
60.	48.108	223.805

GROUP III LOADING

0.3 WIND (GROUP III)

Skew	Mom Perp	Mom Par
0.	65.442	54.804
15.	54.079	65.876
30.	45.800	78.725
45.	32.470	81.887
60.	10.176	74.406

WIND ON LIVE LOAD (GROUP III)

H Perp = 26.00
H Par = 39.00

Skew	Mom Perp	Mom Par
0.	75.991	53.153
15.	62.620	60.454
30.	53.808	70.943
45.	38.815	71.557
60.	12.372	61.388

LONGITUDINAL FORCE (GROUP III)

H Perp = 26.00 LF Mom Perp = 32.567
H Par = 39.00 LF Mom Par = 104.763

ADDITIONAL STREAM FLOW MOMENT PERP (GROUP III)

Skew	Mom Perp	Mom Par
0.	7.940	
15.	8.959	
30.	10.356	
45.	12.037	
60.	13.887	

*************** Bd = 1.00 ***************

Pu = 361.444 Mom Mag = 1.171

Skew	Mom Perp	Mom Par
0.	438.693	436.986
15.	402.579	464.963
30.	378.680	500.501
45.	338.111	506.253
60.	266.713	479.374

***************** Bd = .75 ****************

Pu = 297.223 Mom Mag = 1.137

Skew	Mom Perp	Mom Par
0.	425.731	424.074
15.	390.684	451.225
30.	367.491	485.712
45.	328.121	491.294
60.	258.833	465.210

************* SUMMARY OF COLUMN FORCES ************

	Axial	Mom Perp	Mom Par
GROUP I			
	432.	191.	117.
	368.	185.	114.
GROUP II			
	257.	203.	170.
	257.	168.	204.
	257.	143.	244.
	257.	110.	254.
	257.	50.	231.
	193.	197.	165.
	193.	163.	199.
	193.	139.	237.
	193.	107.	247.
	193.	49.	224.
GROUP III			
	362.	439.	437.
	362.	403.	465.
	362.	379.	501.
	362.	339.	507.
	362.	267.	480.
	298.	426.	425.
	298.	391.	452.
	298.	368.	486.
	298.	329.	492.
	298.	259.	466.

A.1.2 INPUT.PCA

```
JOB NUMBER:    2.3.1
DESCRIPTION:   2-column bent, 4 beams, 2 lanes
LOCATION:      Outahere, TX                          DATE:  1 December 1987
   1    1               30.00
              3420.0
     40.0
   1            2.0    8    9         14   11
   3          22       3
 432. 191. 117. 368. 185. 114. 257. 203. 170. 257. 168. 204. 257. 143. 244.
 257. 110. 254. 257.  50. 231. 193. 197. 165. 193. 163. 199. 193. 139. 237.
 193. 107. 247. 193.  49. 224. 362. 439. 437. 362. 403. 465. 362. 379. 501.
 362. 339. 507. 362. 267. 480. 298. 426. 425. 298. 391. 452. 298. 368. 486.
 298. 329. 492. 298. 259. 466.
```

A.1.3 OUTPUT.PCA

```
         TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2        STRENGTH DESIGN OF REINFORCED COLUMNS - 224114    VER 3.0 OCT 82


JOB NUMBER:   2.3.1

DESCRIPTION:  2-column bent, 4 beams, 2 lanes

PROB
LOCATION:     O:tahere, TX                             DATE:  1 December 1987


TABLE 1.  CONTROL DATA

     PROBLEM TYPE (1=DESIGN, 2=INVESTIGATION)                   1
     REINFORCEMENT PATTERN (1=CIRCULAR, 2=RECTANGULAR)          1
     DIAMETER OF SECTION (INCHES)                           30.00


                                                  TABLE NUMBER
                                         2    3    4    5    6    7    8
     KEEP FROM PRECEDING PROB (1=YES)    0    0    0    0    0    0    0


TABLE 2.  CONSTANTS              * = DEFAULT VALUES

     CONCRETE (KSI)
         STRENGTH              3.60 *    STRESS BLOCK           3.06 *
         MODULUS OF ELAST     3420.      ULTIMATE STRAIN      0.0030 *

     CAPACITY REDUCTION FACTORS
         BENDING               0.90 *    COMPRESSION            0.70 *

     STEEL (KSI)
         YIELD STRENGTH         40.      MODULUS OF ELAST     29000. *

     REINFORCEMENT RATIOS (DECIMAL)
         MINIMUM              0.010 *    MAXIMUM               0.080 *

     MINIMUM CLEAR SPAC (IN.)  2.25 *


TABLE 3.  REINFORCEMENT DATA
                                  * = DEFAULT VALUES

     REINFORCEMENT MODE   1  -  BARS EQUALLY DISTRIBUTED

          CLEAR COVER (INCHES)         2.00

                            LOWER LIMIT     UPPER LIMIT
          NO. OF BARS            8              14
          BAR SIZE NO.           9              11
```

TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION

PCA2 STRENGTH DESIGN OF REINFORCED COLUMNS - 224114 VER 3.0 OCT 82

PROB (CONTD)

LOCATION: Outahere, TX DATE: 1 December 1987

TABLE 5. LOADING AND MOMENT CONTROL DATA

LOAD TYPE	3
NO. OF LOADS OR GROUPS	22
AXIS OPTION (1=X, 2=Y, 3=BOTH)	3

TABLE 8. AXIAL LOAD AND MOMENT GROUPS

APPLIED LOADS AND MOMENTS			COMPUTED LOADS AND MOMENTS			
LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	RATIO OF ULTIMATE TO APPLIED LOAD
432.	191.	117.	1090.	483.	296.	2.524
368.	185.	114.	1007.	508.	313.	2.738
257.	203.	170.	648.	513.	430.	2.526
257.	168.	204.	650.	426.	517.	2.532
257.	143.	244.	606.	338.	576.	2.360
257.	110.	254.	620.	265.	613.	2.413
257.	50.	231.	725.	141.	653.	2.823
193.	197.	165.	492.	502.	421.	2.550
193.	163.	199.	491.	415.	507.	2.546
193.	139.	237.	456.	328.	560.	2.362
193.	107.	247.	468.	259.	598.	2.422
193.	49.	224.	561.	142.	648.	2.898
362.	439.	437.	365.	444.	442.	1.010
362.	403.	465.	367.	410.	473.	1.017
362.	379.	501.	359.	376.	497.	0.993
362.	339.	507.	372.	349.	522.	1.028
362.	267.	480.	424.	312.	561.	1.170
298.	426.	425.	299.	428.	427.	1.005
298.	391.	452.	301.	396.	458.	1.012
298.	368.	486.	295.	364.	481.	0.989
298.	329.	492.	304.	337.	504.	1.023
298.	259.	466.	346.	301.	541.	1.161

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	14	TOTAL AREA (SQ IN.)	17.78
BAR SIZE NO.	10	REINF RATIO (DECIMAL)	0.02515

A.2.1 FRAME.OUT

```
******************************************************************************************

     TEXAS STATE DEPARTMENT OF HIGHWAYS & PUBLIC TRANSPORTATION:  BRIDGE DIVISION

     COLUMN ANALYSIS:  GROUPS I,II, & III                  (AASHTO SPECIFICATIONS)

     AUGUST 1987                                                       VERSION 1.00

******************************************************************************************

NAME OF INPUT FILE:  EXAMPLE PROBLEM #1


JOB NUMBER:    2.3.1
DESCRIPTION:   2-column bent, 4 beams, 2 lanes
LOCATION:      Outahere, TX                          DATE:  1 December 1987


      BENDING AXIS CODE   =    3

      NUMBER OF LANES     =    2               SLAB DEPTH          =     .625  FT
      SPAN LENGTH, AVG    =   64.500  FT       SLAB WIDTH          =   25.000  FT
      FWD SPAN LENGTH     =   60.000  FT       CAP DEPTH           =    2.750  FT
      BACK SPAN LENGTH    =   69.000  FT       CAP WIDTH           =    2.750  FT
      SKEW ANGLE          =   25.000  DEG      CAP LENGTH          =   26.500  FT

      NUMBER OF RAILS     =    2               PEDESTAL HEIGHT     =     .125  FT
      WT/FT OF RAIL       =     .330  K/FT     BEARING PAD HEIGHT  =     .083  FT
      RAIL HEIGHT         =    2.670  FT       HAUNCH              =     .083  FT

      NUMBER BEAMS, FWD   =    4               VELOCITY OF STREAM  =    6.000  FPS
      BEAM SPACING, FWD   =    7.500  FT       WATER DEPTH         =   22.000  FT
      WT/FT OF BEAM, FWD  =     .516  K/FT     DEGREE OF CURVE     =    2.000  DEG
      BEAM DEPTH, FWD     =    3.333  FT       DESIGN SPEED        =   55.000  MPH
      BEARING ECC, FWD    =     .000  IN

      NUMBER BEAMS, BACK  =    4               TEXAS DESIGN WIND   =   80.000  MPH
      BEAM SPACING, BACK  =    7.500  FT       AASHTO HT ABOVE DK  =    6.000  FT
      WT/FT OF BEAM, BK   =     .516  K/FT     BENT CONT. MULT.    =    1.000
      BEAM DEPTH, BACK    =    3.333  FT       OUT OF PLANE DEFL   =     .000  IN
      BEARING ECC, BACK   =     .000  IN

      NUMBER OF COLUMNS   =    2               STEEL YIELD STRESS  =   40.000  KSI
      WT/FT OF COLUMN     =     .736  K/FT     UNIFORM SIZE COVER  =    2.000  IN
      COLUMN HEIGHT       =   22.000  FT       LOWER LIMIT NUMBER  =   8
      DEPTH TO FIXITY     =    4.000  FT       LOWER LIMIT SIZE    =   9
      COLUMN SPACING      =   16.000  FT       UPPER LIMIT NUMBER  =   14
      COLUMN RADIUS       =    1.250  FT       UPPER LIMIT SIZE    =   11
      COL. D PERP TO CAP  =     .000  FT
      COL. D ALONG CAP    =     .000  FT
```

COLUMN CAPACITY, DEFL. LIMIT MOMENT, & BEARING ECC. MOMENTS

Fc OF COLUMN	=	3600.0	PSI
MODULUS OF ELAS	=	3420.0	KSI
MOMENT OF INERTIA	=	39760.8	IN**4
K-FACTOR	=	1.25	
Cm	=	1.00	
Pc	=	3529.5	KIPS
0.7*Pc	=	2470.6	KIPS
DEFL LIM MOM PAR	=	.0	FT-KIP
DL ECCEN. MOMENT	=	.0	FT-KIP
LL ECCEN. MOMENT	=	.0	FT-KIP

GROUP I LOADING

***************** Bd = 1.00 ***************

COL NO.	AXIAL	MOM PERP	MOM PAR
1	444.88	156.01	117.72
2	418.13	150.88	117.72

***************** Bd = .75 ***************

COL NO.	AXIAL	MOM PERP	MOM PAR
1	380.65	151.21	114.10
2	353.91	146.24	114.10

GROUP II LOADING

WIND ON SUPERSTRUCTURE (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	19.480	169.809
1	15.	16.053	139.930
1	30.	13.794	120.239
1	45.	9.950	86.735
1	60.	3.171	27.645
2	0.	-19.481	170.133
2	15.	-16.053	140.197

2	30.	-13.794	120.469
2	45.	-9.950	86.901
2	60.	-3.172	27.698

WIND ON CAP (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	.224	1.957
1	15.	.190	1.654
1	30.	.142	1.238
1	45.	.085	.738
1	60.	.022	.188
2	0.	-.224	1.960
2	15.	-.190	1.657
2	30.	-.142	1.241
2	45.	-.085	.740
2	60.	-.022	.189

WIND ON COLUMN (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	1.156	20.670
1	15.	.977	17.471
1	30.	.732	13.082
1	45.	.436	7.800
1	60.	.111	1.988
2	0.	-1.156	20.670
2	15.	-.977	17.471
2	30.	-.732	13.081
2	45.	-.436	7.800
2	60.	-.111	1.988

WIND REDUCTION (GROUP II)

		100 MPH Wind		Reduced Wind	
Col No.	Skew	Axial	Mom Perp	Axial	Mom Perp
1	0.	20.861	192.435	13.351	123.159
1	15.	17.219	159.055	11.020	101.795
1	30.	14.667	134.559	9.387	86.118
1	45.	10.471	95.274	6.702	60.975
1	60.	3.304	29.821	2.115	19.086

2	0.	-20.861	192.763	-13.351	123.368
2	15.	-17.220	159.325	-11.021	101.968
2	30.	-14.668	134.791	-9.387	86.266
2	45.	-10.471	95.441	-6.702	61.082
2	60.	-3.304	29.874	-2.115	19.120

ADDITIONAL STREAM FLOW FORCES (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	.000	.000
1	15.	.000	.000
1	30.	.034	.603
1	45.	.329	5.884
1	60.	.654	11.696
2	0.	.000	.000
2	15.	.000	.000
2	30.	-.034	.603
2	45.	-.329	5.884
2	60.	-.654	11.696

******************** Bd = 1.00 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	284.00	181.36	172.19
1	15.	279.27	149.98	206.88
1	30.	275.99	127.84	247.13
1	45.	270.92	98.67	257.04
1	60.	262.03	45.68	233.60
2	0.	229.76	180.75	172.19
2	15.	234.50	149.31	206.88
2	30.	237.77	127.13	247.13
2	45.	242.84	97.90	257.04
2	60.	251.74	44.80	233.60

******************** Bd = .75 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	219.78	176.19	167.29
1	15.	215.05	145.70	200.99
1	30.	211.77	124.19	240.09
1	45.	206.70	95.85	249.72
1	60.	197.81	44.37	226.95
2	0.	165.54	175.59	167.29
2	15.	170.28	145.05	200.99
2	30.	173.55	123.51	240.09
2	45.	178.62	95.11	249.72
2	60.	187.52	43.52	226.95

GROUP III LOADING

0.3 WIND (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	6.258	57.731
1	15.	5.166	47.716
1	30.	4.400	40.368
1	45.	3.141	28.582
1	60.	.991	8.946
2	0.	-6.258	57.829
2	15.	-5.166	47.797
2	30.	-4.400	40.437
2	45.	-3.141	28.632
2	60.	-.991	8.962

WIND ON LIVE LOAD (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	3.179	59.431
1	15.	3.616	48.974
1	30.	4.243	42.083
1	45.	4.280	30.356
1	60.	3.672	9.676

2	0.	-3.179	59.545
2	15.	-3.616	49.068
2	30.	-4.243	42.163
2	45.	-4.280	30.414
2	60.	-3.672	9.694

LONGITUDINAL FORCE (GROUP III)

Col No.	Axial	Mom Perp
1	2.922	25.470
2	-2.922	25.519

ADDITIONAL STREAM FLOW FORCES (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	.419	7.483
1	15.	.472	8.443
1	30.	.546	9.760
1	45.	.635	11.344
1	60.	.732	13.088
2	0.	-.419	7.483
2	15.	-.472	8.443
2	30.	-.546	9.760
2	45.	-.635	11.344
2	60.	-.732	13.088

********************* Bd = 1.00 *********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	390.44	362.45	444.37
1	15.	394.30	332.26	472.79
1	30.	394.40	312.27	508.88
1	45.	394.51	278.35	514.73
1	60.	394.64	218.67	487.42
2	0.	332.45	359.56	444.37
2	15.	328.59	329.32	472.79
2	30.	328.49	309.29	508.88
2	45.	328.37	275.30	514.73
2	60.	328.25	215.51	487.42

******************** Bd = .75 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	326.21	351.58	431.05
1	15.	330.08	322.29	458.62
1	30.	330.18	302.90	493.63
1	45.	330.29	269.99	499.30
1	60.	330.42	212.11	472.82
2	0.	268.23	348.78	431.05
2	15.	264.36	319.44	458.62
2	30.	264.27	300.01	493.63
2	45.	264.15	267.04	499.30
2	60.	264.03	209.04	472.82

A.2.2 FRINPUT.PCA

```
JOB NUMBER:     2.3.1
DESCRIPTION:    2-column bent, 4 beams, 2 lanes
LOCATION:       Outahere, TX                            DATE:  1 December 1987
   1    1                 30.00
                3420.0
     40.0
   1               2.0    8    9        14    11
   3          44          3
 445. 156. 118. 418. 151. 118. 381. 151. 114. 354. 146. 114. 284. 181. 172.
 279. 150. 207. 276. 128. 247. 271.  99. 257. 262.  46. 234. 230. 181. 172.
 234. 149. 207. 238. 127. 247. 243.  98. 257. 252.  45. 234. 220. 176. 167.
 215. 146. 201. 212. 124. 240. 207.  96. 250. 198.  44. 227. 166. 176. 167.
 170. 145. 201. 174. 124. 240. 179.  95. 250. 188.  44. 227. 390. 362. 444.
 394. 332. 473. 394. 312. 509. 395. 278. 515. 395. 219. 487. 332. 360. 444.
 329. 329. 473. 328. 309. 509. 328. 275. 515. 328. 216. 487. 326. 352. 431.
 330. 322. 459. 330. 303. 494. 330. 270. 499. 330. 212. 473. 268. 349. 431.
 264. 319. 459. 264. 300. 494. 264. 267. 499. 264. 209. 473.   0.   0.   0.
```

A.2.3 OUTPUT.PCA

```
           TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2          STRENGTH DESIGN OF REINFORCED COLUMNS - 224114     VER 3.0 OCT 82


JOB NUMBER:   2.3.1

DESCRIPTION:  2-column bent, 4 beams, 2 lanes

PROB
LOCATION:     Outahere, TX                                 DATE:  1 December 1987


TABLE 1.  CONTROL DATA

     PROBLEM TYPE (1=DESIGN, 2=INVESTIGATION)                        1
     REINFORCEMENT PATTERN (1=CIRCULAR, 2=RECTANGULAR)               1
     DIAMETER OF SECTION (INCHES)                                30.00


                                                     TABLE NUMBER
                                            2    3    4    5    6    7    8
     KEEP FROM PRECEDING PROB (1=YES)       0    0    0    0    0    0    0


TABLE 2.  CONSTANTS                 * = DEFAULT VALUES

     CONCRETE (KSI)
         STRENGTH                 3.60 *     STRESS BLOCK             3.06 *
         MODULUS OF ELAST        3420.       ULTIMATE STRAIN        0.0030 *

     CAPACITY REDUCTION FACTORS
         BENDING                  0.90 *     COMPRESSION              0.70 *

     STEEL (KSI)
         YIELD STRENGTH            40.       MODULUS OF ELAST       29000. *

     REINFORCEMENT RATIOS (DECIMAL)
         MINIMUM                 0.010 *     MAXIMUM                 0.080 *

     MINIMUM CLEAR SPAC (IN.)     2.25 *


TABLE 3.  REINFORCEMENT DATA
                                    * = DEFAULT VALUES

     REINFORCEMENT MODE   1  -  BARS EQUALLY DISTRIBUTED

          CLEAR COVER (INCHES)         2.00

                            LOWER LIMIT      UPPER LIMIT
          NO. OF BARS            8               14
          BAR SIZE NO.           9               11
```

TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2 STRENGTH DESIGN OF REINFORCED COLUMNS - 224114 VER 3.0 OCT 82

PROB (CONTD)
LOCATION: Outahere, TX DATE: 1 December 1987

TABLE 5. LOADING AND MOMENT CONTROL DATA

LOAD TYPE	3
NO. OF LOADS OR GROUPS	44
AXIS OPTION (1=X, 2=Y, 3=BOTH)	3

TABLE 8. AXIAL LOAD AND MOMENT GROUPS

APPLIED LOADS AND MOMENTS			COMPUTED LOADS AND MOMENTS			
LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	RATIO OF ULTIMATE TO APPLIED LOAD
445.	156.	118.	1181.	414.	313.	2.655
418.	151.	118.	1153.	418.	327.	2.760
381.	151.	114.	1103.	439.	331.	2.898
354.	146.	114.	1070.	443.	346.	3.026
284.	181.	172.	742.	473.	449.	2.613
279.	150.	207.	712.	385.	531.	2.558
276.	128.	247.	651.	302.	583.	2.358
271.	99.	257.	646.	236.	612.	2.383
262.	46.	234.	719.	126.	642.	2.743
230.	181.	172.	603.	474.	451.	2.620
234.	149.	207.	600.	362.	531.	2.564
238.	127.	247.	559.	298.	580.	2.349
243.	98.	257.	577.	233.	611.	2.376
252.	45.	234.	694.	124.	645.	2.753
220.	176.	167.	593.	474.	450.	2.695
215.	146.	201.	565.	384.	528.	2.626
212.	124.	240.	509.	298.	576.	2.400
207.	96.	250.	501.	232.	604.	2.418
198.	44.	227.	559.	124.	641.	2.824
166.	176.	167.	435.	461.	437.	2.618
170.	145.	201.	436.	372.	515.	2.565
174.	124.	240.	404.	288.	557.	2.320
179.	95.	250.	423.	224.	590.	2.360
188.	44.	227.	529.	124.	638.	2.813
390.	362.	444.	432.	401.	492.	1.108
394.	332.	473.	431.	364.	519.	1.096
394.	312.	509.	415.	329.	537.	1.055
395.	278.	515.	425.	301.	557.	1.080

395.	219.	487.	478.	264.	587.	1.207
332.	360.	444.	353.	384.	474.	1.066
329.	329.	473.	346.	347.	499.	1.053
328.	309.	509.	331.	312.	514.	1.010
328.	275.	515.	339.	285.	534.	1.036
328.	216.	487.	382.	251.	567.	1.164
326.	352.	431.	356.	387.	473.	1.097
330.	322.	459.	359.	351.	501.	1.090
330.	303.	494.	345.	317.	517.	1.047
330.	270.	499.	354.	290.	537.	1.075
330.	212.	473.	399.	256.	570.	1.207
268.	349.	431.	281.	366.	452.	1.049
264.	319.	459.	273.	330.	475.	1.035
264.	300.	494.	263.	298.	491.	0.995
264.	267.	499.	268.	272.	509.	1.019
264.	209.	473.	301.	239.	540.	1.142

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	11	TOTAL AREA (SQ IN.)	17.16
BAR SIZE NO.	11	REINF RATIO (DECIMAL)	0.02428

APPENDIX B. COMPUTER OUTPUT FOR EXAMPLE PROBLEM 2

B.1.1 APROX.OUT

```
*************************************************************************************

       TEXAS STATE DEPARTMENT OF HIGHWAYS & PUBLIC TRANSPORTATION:  BRIDGE DIVISION

       COLUMN ANALYSIS:  GROUPS I,II, & III                 (AASHTO SPECIFICATIONS)

       AUGUST 1987                                                     VERSION 1.00

*************************************************************************************

NAME OF INPUT FILE:  EXAMPLE PROBLEM #2


JOB NUMBER:    2.3.2
DESCRIPTION:   3-column bent, 6 beams, 3 lanes
LOCATION:      Outahere, TX                         DATE:  1 December 1987

      BENDING AXIS CODE  =    3

      NUMBER OF LANES    =    3                SLAB DEPTH          =      .625  FT
      SPAN LENGTH, AVG   =   64.500  FT        SLAB WIDTH          =    40.000  FT
      FWD SPAN LENGTH    =   60.000  FT        CAP DEPTH           =     2.750  FT
      BACK SPAN LENGTH   =   69.000  FT        CAP WIDTH           =     2.750  FT
      SKEW ANGLE         =   25.000  DEG       CAP LENGTH          =    41.500  FT

      NUMBER OF RAILS    =    2                PEDESTAL HEIGHT     =      .125  FT
      WT/FT OF RAIL      =     .330  K/FT      BEARING PAD HEIGHT  =      .083  FT
      RAIL HEIGHT        =    2.670  FT        HAUNCH              =      .083  FT

      NUMBER BEAMS, FWD  =    6                VELOCITY OF STREAM  =     6.000  FPS
      BEAM SPACING, FWD  =    7.500  FT        WATER DEPTH         =    22.000  FT
      WT/FT OF BEAM, FWD =     .516  K/FT      DEGREE OF CURVE     =     2.000  DEG
      BEAM DEPTH, FWD    =    3.333  FT        DESIGN SPEED        =    55.000  MPH
      BEARING ECC, FWD   =     .000  IN

      NUMBER BEAMS, BACK =    6                TEXAS DESIGN WIND   =    80.000  MPH
      BEAM SPACING, BACK =    7.500  FT        AASHTO HT ABOVE DK  =     6.000  FT
      WT/FT OF BEAM, BK  =     .516  K/FT      BENT CONT. MULT.    =     1.000
      BEAM DEPTH, BACK   =    3.333  FT        OUT OF PLANE DEFL   =      .000  IN
      BEARING ECC, BACK  =     .000  IN

      NUMBER OF COLUMNS  =    3                STEEL YIELD STRESS  =    40.000  KSI
      WT/FT OF COLUMN    =     .736  K/FT      UNIFORM SIZE COVER  =     2.000  IN
      COLUMN HEIGHT      =   22.000  FT        LOWER LIMIT NUMBER  =     8
      DEPTH TO FIXITY    =    4.000  FT        LOWER LIMIT SIZE    =     9
      COLUMN SPACING     =   16.000  FT        UPPER LIMIT NUMBER  =    14
      COLUMN RADIUS      =    1.250  FT        UPPER LIMIT SIZE    =    11
      COL. D PERP TO CAP =     .000  FT
      COL. D ALONG CAP   =     .000  FT
```

DEAD LOAD AND LIVE LOAD

```
REDUCTION FACTOR    =      .900
DEAD LOAD PER COL   =   196.207  KIPS
LL LANE             =    53.352  KIPS
LL TRUCK            =    57.277  KIPS
IMPACT FACTOR       =     1.264
LL+I                =    72.389  KIPS
```

COLUMN CAPACITY, DEFL. LIMIT MOMENT, & BEARING ECC. MOMENTS

```
Fc OF COLUMN        =    3600.0  PSI
MODULUS OF ELAS     =    3420.0  KSI
MOMENT OF INERTIA   =   39760.8  IN**4
K-FACTOR            =      1.25
Cm                  =      1.00
Pc                  =    3529.5  KIPS
0.7*Pc              =    2470.6  KIPS
DEFL LIM MOM PAR    =        .0  FT-KIP
DL ECCEN. MOMENT    =        .0  FT-KIP
LL ECCEN. MOMENT    =        .0  FT-KIP
```

GROUP I LOADING

```
CENTRIFUGAL FORCE (GROUP I)

    C      =   7.08     C.F.          =    4.054

    H Perp =  26.00     CF Mom Perp =   95.532
    H Par  =  39.00     CF Mom Par  =   66.822

STREAM FLOW (GROUP I)

    P      =   .024     SF Mom Perp =   14.520

  ***************** Bd = 1.00 ****************

    Pu   =  412.154     Mom Mag     =    1.200

    Sum Mom Perp =  171.713
    Sum Mom Par  =  104.262
```

MINIMUM ECCENTRICITY MOMENT

Min Mom = 123.669
Resultant= 200.888

CONTROLLING MOMENTS (GROUP I)

Mom Perp = 171.713
Mom Par = 104.262

***************** Bd = .75 ****************

Pu = 348.386 Mom Mag = 1.164

Sum Mom Perp = 166.554
Sum Mom Par = 101.129

MINIMUM ECCENTRICITY MOMENT

Min Mom = 101.394
Resultant= 194.852

CONTROLLING MOMENTS (GROUP I)

Mom Perp = 166.554
Mom Par = 101.129

GROUP II LOADING

WIND ON SUPERSTRUCTURE (GROUP II)

Wind Area Per Bent = 446.28

H Perp = 19.21
H Par = 32.21

Skew	Mom Perp	Mom Par
0.	129.492	101.247
15.	106.707	115.153
30.	91.692	135.133
45.	66.142	136.304
60.	21.082	116.932

WIND ON CAP (GROUP II)

Cap Area Perp	=	7.56
Cap Area Par	=	114.12
H Perp	=	14.38
H Par	=	27.37

Skew	Mom Perp	Mom Par
0.	1.314	17.604
15.	1.110	26.776
30.	.831	34.122
45.	.496	39.143
60.	.126	41.497

WIND ON COLUMN (GROUP II)

Column Area Perp	=	55.00
Column Area Par	=	55.00
H Perp	=	11.00
H Par	=	15.00

Skew	Mom Perp	Mom Par
0.	21.933	13.946
15.	18.538	21.212
30.	13.881	27.032
45.	8.277	31.010
60.	2.109	32.874

WIND REDUCTION (GROUP II)

Wind Red. Factor = .640

	100 MPH Wind		Reduced Wind	
Skew	Mom Perp	Mom Par	Mom Perp	Mom Par
0.	152.739	132.798	97.753	84.991
15.	126.356	163.141	80.868	104.410
30.	106.404	196.287	68.098	125.624
45.	74.915	206.457	47.946	132.132
60.	23.317	191.303	14.923	122.434

ADDITIONAL STREAM FLOW MOMENT PERP (GROUP II)

Skew	Mom Perp	Mom Par
0.	.000	
15.	.000	
30.	.639	
45.	6.243	
60.	12.411	

***************** Bd = 1.00 ****************

Pu = 255.069 Mom Mag = 1.115

Skew	Mom Perp	Mom Par
0.	141.708	123.208
15.	117.231	151.359
30.	99.647	182.112
45.	78.555	191.547
60.	39.625	177.488

***************** Bd = .75 ****************

Pu = 191.302 Mom Mag = 1.084

Skew	Mom Perp	Mom Par
0.	137.744	119.761
15.	113.951	147.125
30.	96.859	177.018
45.	76.358	186.189
60.	38.516	172.523

GROUP III LOADING

0.3 WIND (GROUP III)

Skew	Mom Perp	Mom Par
0.	45.822	39.839
15.	37.907	48.942
30.	31.921	58.886
45.	22.474	61.937
60.	6.995	57.391

WIND ON LIVE LOAD (GROUP III)

H Perp = 26.00
H Par = 39.00

Skew	Mom Perp	Mom Par
0.	50.661	35.436
15.	41.747	40.302
30.	35.872	47.295
45.	25.877	47.705
60.	8.248	40.925

LONGITUDINAL FORCE (GROUP III)

H Perp = 26.00 LF Mom Perp = 29.311
H Par = 39.00 LF Mom Par = 94.287

ADDITIONAL STREAM FLOW MOMENT PERP (GROUP III)

Skew	Mom Perp	Mom Par
0.	7.940	
15.	8.959	
30.	10.356	
45.	12.037	
60.	13.887	

***************** Bd = 1.00 ****************

Pu = 349.175 Mom Mag = 1.165

Skew	Mom Perp	Mom Par
0.	347.101	357.878
15.	323.164	379.027
30.	307.323	404.669
45.	280.434	409.908
60.	233.110	392.761

***************** Bd = .75 ****************

Pu = 285.408 Mom Mag = 1.131

Skew	Mom Perp	Mom Par
0.	336.972	347.434
15.	313.734	367.966
30.	298.355	392.860
45.	272.250	397.946
60.	226.308	381.300

************* SUMMARY OF COLUMN FORCES ************

	Axial	Mom Perp	Mom Par
GROUP I			
	413.	172.	105.
	349.	167.	102.
GROUP II			
	256.	142.	124.
	256.	118.	152.
	256.	100.	183.
	256.	79.	192.
	256.	40.	178.
	192.	138.	120.
	192.	114.	148.
	192.	97.	178.
	192.	77.	187.
	192.	39.	173.
GROUP III			
	350.	348.	358.
	350.	324.	380.
	350.	308.	405.
	350.	281.	410.
	350.	234.	393.
	286.	337.	348.
	286.	314.	368.
	286.	299.	393.
	286.	273.	398.
	286.	227.	382.

B.1.2 INPUT.PCA

```
JOB NUMBER:   2.3.2
DESCRIPTION:  3-column bent, 6 beams, 3 lanes
LOCATION:     Outahere, TX                            DATE:  1 December 1987
   1    1               30.00
              3420.0
     40.0
   1              2.0    8    9        14   11
   3         22          3
 413. 172. 105. 349. 167. 102. 256. 142. 124. 256. 118. 152. 256. 100. 183.
 256.  79. 192. 256.  40. 178. 192. 138. 120. 192. 114. 148. 192.  97. 178.
 192.  77. 187. 192.  39. 173. 350. 348. 358. 350. 324. 380. 350. 308. 405.
 350. 281. 410. 350. 234. 393. 286. 337. 348. 286. 314. 368. 286. 299. 393.
 286. 273. 398. 286. 227. 382.
```

B.1.3 OUTPUT.PCA

```
           TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2          STRENGTH DESIGN OF REINFORCED COLUMNS - 224114    VER 3.0 OCT 82


JOB NUMBER:   2.3.2

DESCRIPTION:  3-column bent, 6 beams, 3 lanes

PROB
LOCATION:     Outahere, TX                              DATE:  1 December 1987


TABLE 1.  CONTROL DATA

     PROBLEM TYPE (1=DESIGN, 2=INVESTIGATION)                    1
     REINFORCEMENT PATTERN (1=CIRCULAR, 2=RECTANGULAR)           1
     DIAMETER OF SECTION (INCHES)                            30.00


                                                  TABLE NUMBER
                                         2    3    4    5    6    7    8
     KEEP FROM PRECEDING PROB (1=YES)    0    0    0    0    0    0    0


TABLE 2.  CONSTANTS               * = DEFAULT VALUES

     CONCRETE (KSI)
         STRENGTH              3.60 *     STRESS BLOCK             3.06 *
         MODULUS OF ELAST     3420.       ULTIMATE STRAIN        0.0030 *

     CAPACITY REDUCTION FACTORS
         BENDING               0.90 *     COMPRESSION              0.70 *

     STEEL (KSI)
         YIELD STRENGTH         40.       MODULUS OF ELAST       29000. *

     REINFORCEMENT RATIOS (DECIMAL)
         MINIMUM              0.010 *     MAXIMUM                 0.080 *

     MINIMUM CLEAR SPAC (IN.)  2.25 *


TABLE 3.  REINFORCEMENT DATA
                                  * = DEFAULT VALUES

     REINFORCEMENT MODE   1  -  BARS EQUALLY DISTRIBUTED

          CLEAR COVER (INCHES)        2.00

                           LOWER LIMIT     UPPER LIMIT
          NO. OF BARS           8              14
          BAR SIZE NO.          9              11
```

TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2 STRENGTH DESIGN OF REINFORCED COLUMNS - 224114 VER 3.0 OCT 82

PROB (CONTD)
LOCATION: Outahere, TX DATE: 1 December 1987

TABLE 5. LOADING AND MOMENT CONTROL DATA

LOAD TYPE	3
NO. OF LOADS OR GROUPS	22
AXIS OPTION (1=X, 2=Y, 3=BOTH)	3

TABLE 8. AXIAL LOAD AND MOMENT GROUPS

APPLIED LOADS AND MOMENTS			COMPUTED LOADS AND MOMENTS			
LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	RATIO OF ULTIMATE TO APPLIED LOAD
413.	172.	105.	1027.	430.	262.	2.488
349.	167.	102.	942.	452.	276.	2.700
256.	142.	124.	772.	428.	374.	3.015
256.	118.	152.	756.	350.	450.	2.956
256.	100.	183.	704.	276.	504.	2.753
256.	79.	192.	708.	219.	532.	2.768
256.	40.	178.	794.	124.	553.	3.102
192.	138.	120.	601.	432.	376.	3.133
192.	114.	148.	587.	349.	453.	3.058
192.	97.	178.	535.	270.	496.	2.785
192.	77.	187.	537.	215.	523.	2.796
192.	39.	173.	623.	126.	561.	3.243
350.	348.	358.	368.	367.	378.	1.054
350.	324.	380.	368.	342.	401.	1.053
350.	308.	405.	359.	317.	416.	1.027
350.	281.	410.	370.	298.	434.	1.058
350.	234.	393.	412.	276.	463.	1.178
286.	337.	348.	290.	343.	354.	1.017
286.	314.	368.	291.	320.	375.	1.019
286.	299.	393.	284.	297.	390.	0.993
286.	273.	398.	291.	279.	407.	1.022
286.	227.	382.	329.	261.	439.	1.149

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	12	TOTAL AREA (SQ IN.)	12.00
BAR SIZE NO.	9	REINF RATIO (DECIMAL)	0.01698

B.2.1 FRAME.OUT

```
************************************************************************************

     TEXAS STATE DEPARTMENT OF HIGHWAYS & PUBLIC TRANSPORTATION:  BRIDGE DIVISION

     COLUMN ANALYSIS:  GROUPS I,II, & III              (AASHTO SPECIFICATIONS)

     AUGUST 1987                                                  VERSION 1.00

************************************************************************************

NAME OF INPUT FILE:  EXAMPLE PROBLEM #2


JOB NUMBER:    2.3.2
DESCRIPTION:   3-column bent, 6 beams, 3 lanes
LOCATION:      Outahere, TX                           DATE:  1 December 1987


      BENDING AXIS CODE  =    3

      NUMBER OF LANES    =    3                SLAB DEPTH          =     .625  FT
      SPAN LENGTH, AVG   =   64.500  FT        SLAB WIDTH          =   40.000  FT
      FWD SPAN LENGTH    =   60.000  FT        CAP DEPTH           =    2.750  FT
      BACK SPAN LENGTH   =   69.000  FT        CAP WIDTH           =    2.750  FT
      SKEW ANGLE         =   25.000  DEG       CAP LENGTH          =   41.500  FT

      NUMBER OF RAILS    =    2                PEDESTAL HEIGHT     =     .125  FT
      WT/FT OF RAIL      =     .330  K/FT      BEARING PAD HEIGHT  =     .083  FT
      RAIL HEIGHT        =    2.670  FT        HAUNCH              =     .083  FT

      NUMBER BEAMS, FWD  =    6                VELOCITY OF STREAM  =    6.000  FPS
      BEAM SPACING, FWD  =    7.500  FT        WATER DEPTH         =   22.000  FT
      WT/FT OF BEAM, FWD =     .516  K/FT      DEGREE OF CURVE     =    2.000  DEG
      BEAM DEPTH, FWD    =    3.333  FT        DESIGN SPEED        =   55.000  MPH
      BEARING ECC, FWD   =     .000  IN

      NUMBER BEAMS, BACK =    6                TEXAS DESIGN WIND   =   80.000  MPH
      BEAM SPACING, BACK =    7.500  FT        AASHTO HT ABOVE DK  =    6.000  FT
      WT/FT OF BEAM, BK  =     .516  K/FT      BENT CONT. MULT.    =    1.000
      BEAM DEPTH, BACK   =    3.333  FT        OUT OF PLANE DEFL   =     .000  IN
      BEARING ECC, BACK  =     .000  IN

      NUMBER OF COLUMNS  =    3                STEEL YIELD STRESS  =   40.000  KSI
      WT/FT OF COLUMN    =     .736  K/FT      UNIFORM SIZE COVER  =    2.000  IN
      COLUMN HEIGHT      =   22.000  FT        LOWER LIMIT NUMBER  =    8
      DEPTH TO FIXITY    =    4.000  FT        LOWER LIMIT SIZE    =    9
      COLUMN SPACING     =   16.000  FT        UPPER LIMIT NUMBER  =   14
      COLUMN RADIUS      =    1.250  FT        UPPER LIMIT SIZE    =   11
      COL. D PERP TO CAP =     .000  FT
      COL. D ALONG CAP   =     .000  FT
```

COLUMN CAPACITY, DEFL. LIMIT MOMENT, & BEARING ECC. MOMENTS

Fc OF COLUMN	=	3600.0	PSI
MODULUS OF ELAS	=	3420.0	KSI
MOMENT OF INERTIA	=	39760.8	IN**4
K-FACTOR	=	1.25	
Cm	=	1.00	
Pc	=	3529.5	KIPS
0.7*Pc	=	2470.6	KIPS
DEFL LIM MOM PAR	=	.0	FT-KIP
DL ECCEN. MOMENT	=	.0	FT-KIP
LL ECCEN. MOMENT	=	.0	FT-KIP

GROUP I LOADING

***************** Bd = 1.00 ***************

COL NO.	AXIAL	MOM PERP	MOM PAR
1	403.14	124.46	106.15
2	448.71	146.83	106.15
3	384.83	148.10	106.15

***************** Bd = .75 ***************

COL NO.	AXIAL	MOM PERP	MOM PAR
1	342.24	120.32	102.62
2	379.20	141.95	102.62
3	323.93	143.18	102.62

GROUP II LOADING

WIND ON SUPERSTRUCTURE (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	9.768	109.249
1	15.	8.050	90.026
1	30.	6.917	77.358
1	45.	4.990	55.802
1	60.	1.590	17.786

2	0.	.054	118.823
2	15.	.044	97.915
2	30.	.038	84.137
2	45.	.027	60.692
2	60.	.009	19.345
3	0.	-9.822	109.938
3	15.	-8.094	90.593
3	30.	-6.955	77.845
3	45.	-5.017	56.154
3	60.	-1.599	17.898

WIND ON CAP (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	.113	1.259
1	15.	.095	1.064
1	30.	.071	.797
1	45.	.042	.475
1	60.	.011	.121
2	0.	.001	1.369
2	15.	.001	1.157
2	30.	0.000	.866
2	45.	0.000	.517
2	60.	0.000	.132
3	0.	-.113	1.267
3	15.	-.096	1.071
3	30.	-.072	.802
3	45.	-.043	.478
3	60.	-.011	.122

WIND ON COLUMN (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	.872	20.334
1	15.	.737	17.187
1	30.	.552	12.869
1	45.	.329	7.674
1	60.	.084	1.955
2	0.	0.000	21.166
2	15.	0.000	17.890
2	30.	0.000	13.395
2	45.	0.000	7.988
2	60.	0.000	2.035

3	0.	-.872	20.334
3	15.	-.737	17.187
3	30.	-.552	12.869
3	45.	-.329	7.674
3	60.	-.084	1.955

WIND REDUCTION (GROUP II)

		100 MPH Wind		Reduced Wind	
Col No.	Skew	Axial	Mom Perp	Axial	Mom Perp
1	0.	10.753	130.842	6.882	83.739
1	15.	8.882	108.277	5.684	69.297
1	30.	7.540	91.023	4.826	58.255
1	45.	5.361	63.951	3.431	40.929
1	60.	1.685	19.863	1.078	12.712
2	0.	.054	141.358	.035	90.469
2	15.	.045	116.962	.029	74.856
2	30.	.038	98.398	.025	62.975
2	45.	.028	69.197	.018	44.286
2	60.	.009	21.512	.006	13.768
3	0.	-10.807	131.539	-6.917	84.185
3	15.	-8.926	108.851	-5.713	69.665
3	30.	-7.578	91.516	-4.850	58.570
3	45.	-5.389	64.306	-3.449	41.156
3	60.	-1.694	19.975	-1.084	12.784

ADDITIONAL STREAM FLOW FORCES (GROUP II)

Col No.	Skew	Axial	Mom Perp
1	0.	.000	.000
1	15.	.000	.000
1	30.	.025	.593
1	45.	.248	5.788
1	60.	.493	11.506
2	0.	.000	.000
2	15.	.000	.000
2	30.	0.000	.617
2	45.	0.000	6.025
2	60.	0.000	11.977

3	0.	.000	.000
3	15.	.000	.000
3	30.	-.025	.593
3	45.	-.248	5.788
3	60.	-.493	11.506

******************** Bd = 1.00 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	257.56	115.73	124.97
1	15.	255.13	94.58	153.41
1	30.	253.42	79.27	184.48
1	45.	250.88	61.50	194.01
1	60.	246.42	28.54	179.81
2	0.	278.11	132.53	124.97
2	15.	278.10	109.66	153.41
2	30.	278.09	93.16	184.48
2	45.	278.07	73.70	194.01
2	60.	278.05	37.71	179.81
3	0.	229.54	130.26	124.97
3	15.	231.98	108.99	153.41
3	30.	233.70	93.60	184.48
3	45.	236.26	75.70	194.01
3	60.	240.74	42.52	179.81

******************** Bd = .75 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	196.67	112.18	121.14
1	15.	194.23	91.67	148.71
1	30.	192.52	76.83	178.83
1	45.	189.98	59.61	188.07
1	60.	185.52	27.66	174.30
2	0.	208.60	128.45	121.14
2	15.	208.59	106.29	148.71
2	30.	208.58	90.29	178.83
2	45.	208.56	71.44	188.07
2	60.	208.54	36.55	174.30

3	0.	168.64	126.26	121.14
3	15.	171.08	105.64	148.71
3	30.	172.80	90.73	178.83
3	45.	175.36	73.38	188.07
3	60.	179.85	41.21	174.30

GROUP III LOADING

0.3 WIND (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	3.226	39.253
1	15.	2.665	32.483
1	30.	2.262	27.307
1	45.	1.608	19.185
1	60.	.506	5.959
2	0.	.016	42.407
2	15.	.013	35.089
2	30.	.011	29.520
2	45.	.008	20.759
2	60.	.003	6.454
3	0.	-3.242	39.462
3	15.	-2.678	32.655
3	30.	-2.273	27.455
3	45.	-1.617	19.292
3	60.	-.508	5.993

WIND ON LIVE LOAD (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	1.594	38.236
1	15.	1.813	31.508
1	30.	2.128	27.075
1	45.	2.146	19.530
1	60.	1.841	6.225
2	0.	.009	41.587
2	15.	.010	34.269
2	30.	.012	29.447
2	45.	.012	21.242
2	60.	.010	6.770

3	0.	-1.603	38.477
3	15.	-1.823	31.707
3	30.	-2.139	27.245
3	45.	-2.158	19.653
3	60.	-1.851	6.264

LONGITUDINAL FORCE (GROUP III)

Col No.	Axial	Mom Perp
1	1.978	22.122
2	.011	24.061
3	-1.989	22.262

ADDITIONAL STREAM FLOW FORCES (GROUP III)

Col No.	Skew	Axial	Mom Perp
1	0.	.316	7.362
1	15.	.356	8.306
1	30.	.412	9.601
1	45.	.479	11.160
1	60.	.552	12.875
2	0.	0.000	7.663
2	15.	0.000	8.645
2	30.	0.000	9.994
2	45.	0.000	11.616
2	60.	0.000	13.402
3	0.	-.316	7.362
3	15.	-.356	8.306
3	30.	-.412	9.601
3	45.	-.479	11.160
3	60.	-.552	12.875

******************** Bd = 1.00 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	351.29	265.60	363.62
1	15.	353.91	246.31	385.08
1	30.	353.98	233.53	411.10

1	45.	354.07	211.86	416.42
1	60.	354.17	173.73	399.02
2	0.	380.30	298.30	363.62
2	15.	380.32	277.32	385.08
2	30.	380.32	263.43	411.10
2	45.	380.32	239.85	416.42
2	60.	380.32	198.38	399.02
3	0.	315.94	286.32	363.62
3	15.	313.30	266.91	385.08
3	30.	313.22	254.05	411.10
3	45.	313.14	232.24	416.42
3	60.	313.04	193.87	399.02

******************** Bd = .75 ********************

Col No.	Skew	Axial	Mom Perp	Mom Par
1	0.	290.39	257.05	351.94
1	15.	293.01	238.38	372.71
1	30.	293.09	226.02	397.89
1	45.	293.17	205.04	403.03
1	60.	293.27	168.13	386.19
2	0.	310.79	288.70	351.94
2	15.	310.81	268.40	372.71
2	30.	310.81	254.95	397.89
2	45.	310.81	232.13	403.03
2	60.	310.81	191.99	386.19
3	0.	255.04	277.10	351.94
3	15.	252.40	258.32	372.71
3	30.	252.33	245.87	397.89
3	45.	252.24	224.76	403.03
3	60.	252.15	187.63	386.19

B.2.2 FRINPUT.PCA

```
JOB NUMBER:     2.3.2
DESCRIPTION:    3-column bent, 6 beams, 3 lanes
LOCATION:       Outahere, TX                            DATE:  1 December 1987
    1    1                  30.00
                3420.0
       40.0
    1              2.0     8     9          14    11
    3          66          3
  403. 124. 106. 449. 147. 106. 385. 148. 106. 342. 120. 103. 379. 142. 103.
  324. 143. 103. 258. 116. 125. 255.  95. 153. 253.  79. 184. 251.  61. 194.
  246.  29. 180. 278. 133. 125. 278. 110. 153. 278.  93. 184. 278.  74. 194.
  278.  38. 180. 230. 130. 125. 232. 109. 153. 234.  94. 184. 236.  76. 194.
  241.  43. 180. 197. 112. 121. 194.  92. 149. 193.  77. 179. 190.  60. 188.
  186.  28. 174. 209. 128. 121. 209. 106. 149. 209.  90. 179. 209.  71. 188.
  209.  37. 174. 169. 126. 121. 171. 106. 149. 173.  91. 179. 175.  73. 188.
  180.  41. 174. 351. 266. 364. 354. 246. 385. 354. 234. 411. 354. 212. 416.
  354. 174. 399. 380. 298. 364. 380. 277. 385. 380. 263. 411. 380. 240. 416.
  380. 198. 399. 316. 286. 364. 313. 267. 385. 313. 254. 411. 313. 232. 416.
  313. 194. 399. 290. 257. 352. 293. 238. 373. 293. 226. 398. 293. 205. 403.
  293. 168. 386. 311. 289. 352. 311. 268. 373. 311. 255. 398. 311. 232. 403.
  311. 192. 386. 255. 277. 352. 252. 258. 373. 252. 246. 398. 252. 225. 403.
  252. 188. 386.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
```

B.2.3 OUTPUT.PCA

```
          TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2         STRENGTH DESIGN OF REINFORCED COLUMNS - 224114     VER 3.0 OCT 82


JOB NUMBER:   2.3.2

DESCRIPTION:  3-column bent, 6 beams, 3 lanes

PROB
LOCATION:     Outahere, TX                                  DATE:  1 December 1987


TABLE 1.  CONTROL DATA

     PROBLEM TYPE (1=DESIGN, 2=INVESTIGATION)                       1
     REINFORCEMENT PATTERN (1=CIRCULAR, 2=RECTANGULAR)              1
     DIAMETER OF SECTION (INCHES)                               30.00


                                                   TABLE NUMBER
                                      2    3    4    5    6    7    8
     KEEP FROM PRECEDING PROB (1=YES) 0    0    0    0    0    0    0


TABLE 2.  CONSTANTS             * = DEFAULT VALUES

     CONCRETE (KSI)
        STRENGTH              3.60 *     STRESS BLOCK              3.06 *
        MODULUS OF ELAST     3420.       ULTIMATE STRAIN         0.0030 *

     CAPACITY REDUCTION FACTORS
        BENDING               0.90 *     COMPRESSION               0.70 *

     STEEL (KSI)
        YIELD STRENGTH         40.       MODULUS OF ELAST        29000. *

     REINFORCEMENT RATIOS (DECIMAL)
        MINIMUM              0.010 *     MAXIMUM                  0.080 *

     MINIMUM CLEAR SPAC (IN.)    2.25 *


TABLE 3.  REINFORCEMENT DATA
                                * = DEFAULT VALUES

     REINFORCEMENT MODE   1  -  BARS EQUALLY DISTRIBUTED

          CLEAR COVER (INCHES)        2.00

                          LOWER LIMIT      UPPER LIMIT
          NO. OF BARS          8                14
          BAR SIZE NO.         9                11
```

TEXAS DEPARTMENT OF HIGHWAYS AND PUBLIC TRANSPORTATION
PCA2 STRENGTH DESIGN OF REINFORCED COLUMNS - 224114 VER 3.0 OCT 82

PROB (CONTD)
LOCATION: Outahere, TX DATE: 1 December 1987

TABLE 5. LOADING AND MOMENT CONTROL DATA

LOAD TYPE	3
NO. OF LOADS OR GROUPS	66
AXIS OPTION (1=X, 2=Y, 3=BOTH)	3

TABLE 8. AXIAL LOAD AND MOMENT GROUPS

APPLIED LOADS AND MOMENTS			COMPUTED LOADS AND MOMENTS			
LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	LOAD (KIPS)	MOMENTS (K-FT) ABOUT X	ABOUT Y	RATIO OF ULTIMATE TO APPLIED LOAD
403.	124.	106.	1127.	347.	296.	2.795
449.	147.	106.	1128.	370.	266.	2.513
385.	148.	106.	1036.	398.	285.	2.692
342.	120.	103.	1050.	368.	316.	3.070
379.	142.	103.	1049.	393.	285.	2.769
324.	143.	103.	948.	420.	302.	2.927
258.	116.	125.	829.	373.	402.	3.213
255.	95.	153.	788.	294.	473.	3.090
253.	79.	184.	711.	222.	518.	2.812
251.	61.	194.	696.	169.	538.	2.773
246.	29.	180.	756.	89.	554.	3.075
278.	133.	125.	833.	399.	375.	2.996
278.	110.	153.	814.	322.	448.	2.927
278.	93.	184.	753.	253.	501.	2.715
278.	74.	194.	749.	200.	525.	2.699
278.	38.	180.	828.	113.	537.	2.981
230.	130.	125.	716.	405.	390.	3.114
232.	109.	153.	695.	327.	459.	2.996
234.	94.	184.	638.	256.	502.	2.726
236.	76.	194.	638.	206.	525.	2.705
241.	43.	180.	732.	131.	548.	3.041
197.	112.	121.	672.	382.	413.	3.412
194.	92.	149.	623.	296.	479.	3.213
193.	77.	179.	552.	220.	512.	2.860
190.	60.	188.	534.	168.	528.	2.808
186.	28.	174.	593.	89.	555.	3.190

209.	128.	121.	667.	409.	387.	3.193
209.	106.	149.	643.	326.	459.	3.078
209.	90.	179.	585.	252.	501.	2.799
209.	71.	188.	583.	198.	525.	2.791
209.	37.	174.	663.	117.	552.	3.173
169.	126.	121.	538.	401.	385.	3.183
171.	106.	149.	516.	320.	450.	3.018
173.	91.	179.	468.	246.	484.	2.705
175.	73.	188.	473.	197.	507.	2.699
180.	41.	174.	562.	128.	544.	3.124
351.	266.	364.	410.	310.	424.	1.167
354.	246.	385.	407.	283.	442.	1.150
354.	234.	411.	388.	257.	451.	1.097
354.	212.	416.	396.	237.	464.	1.117
354.	174.	399.	436.	214.	490.	1.229
380.	298.	364.	431.	337.	412.	1.132
380.	277.	385.	426.	310.	431.	1.120
380.	263.	411.	407.	283.	442.	1.074
380.	240.	416.	419.	264.	458.	1.102
380.	198.	399.	461.	241.	485.	1.215
316.	286.	364.	341.	310.	394.	1.081
313.	267.	385.	331.	283.	409.	1.061
313.	254.	411.	317.	258.	418.	1.016
313.	232.	416.	324.	241.	432.	1.038
313.	194.	399.	359.	222.	457.	1.146
290.	257.	352.	331.	293.	402.	1.141
293.	238.	373.	329.	267.	419.	1.122
293.	226.	398.	312.	242.	426.	1.069
293.	205.	403.	319.	223.	439.	1.089
293.	168.	386.	353.	202.	464.	1.202
311.	289.	352.	343.	319.	388.	1.102
311.	268.	373.	337.	292.	406.	1.086
311.	255.	398.	325.	267.	417.	1.046
311.	232.	403.	332.	248.	431.	1.069
311.	192.	386.	370.	228.	459.	1.189
255.	277.	352.	267.	291.	369.	1.049
252.	258.	373.	258.	265.	383.	1.027
252.	246.	398.	250.	245.	396.	0.994
252.	225.	403.	254.	227.	406.	1.007
252.	188.	386.	279.	208.	428.	1.108

THE REINFORCEMENT PATTERN FOR THE DESIGN IS

NO. OF BARS	9	TOTAL AREA (SQ IN.)	11.43
BAR SIZE NO.	10	REINF RATIO (DECIMAL)	0.01617

APPENDIX C. LISTING OF FORTRAN PROGRAM BENTCOL

```
      PROGRAM BENTCOL
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
      COMMON/TEN/ SPANF,SPANB
*
      COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
      COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *            FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
      COMMON/FR3/ TFORCE(3,20)
*
      COMMON/FONE/  YMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),
     *              DC(80,2)
*
      CHARACTER FNAME*64,CARD1*80,CARD2*80,CARD3*80
*
*     OPEN INPUT FILE
*
      WRITE(*,10)
 10   FORMAT(1X,'INPUT FILENAME-')
      READ(*,20) FNAME
 20   FORMAT(A)
      OPEN(3,FILE=FNAME,STATUS='OLD')
*
*     READ INPUT DATA
*
      READ(3,*)   ICODE
      READ(3,180) CARD1
      READ(3,180) CARD2
      READ(3,180) CARD3
      READ(3,*)   IAXIS
      READ(3,*)   PHI,NOLAN
      READ(3,*)   WRAIL,NORAL,RALHT
      READ(3,*)   SLBD,SLBW
      READ(3,*)   CAPD,CAPW,CAPL
      READ(3,*)   SPANF,WBMF,NOBMF,BMDF,ECCF,BMSPF
      READ(3,*)   SPANB,WBMB,NOBMB,BMDB,ECCB,BMSPB
      READ(3,*)   PDSTL,PAD,HNCH
      READ(3,*)   WCOL,FC,NOCOL,COLH,COLSP,COLK,CM,FIXD,DEFL2
      READ(3,*)   R,D1,D2
      READ(3,*)   CRV,DSMPH
      READ(3,*)   V,WTRD
      READ(3,*)   DSWIND,HDECK,CMULT
      READ(3,*)   FY,COVER,INUMB,ISIZE,JNUMB,JSIZE
      CLOSE(3)
*
```

```
*     ASSIGN LONG SPAN TO "SPAN1" AND SHORT SPAN TO SPAN2
*
      IF(SPANF.LT.SPANB)THEN
         SPAN1 = SPANB
         SPAN2 = SPANF
      ELSE
         SPAN1  = SPANF
         SPAN2  = SPANB
      ENDIF
*
*     ASSIGN DEEPER BEAM TO "BMD"
*
      IF(BMDF.LT.BMDB)THEN
         BMD = BMDB
      ELSE
         BMD = BMDF
      ENDIF
*
      SPAN    = (SPAN1+SPAN2)/2.
*
*     OPEN OUTPUT FILES
*
      OPEN(4,FILE='INPUT.PCA',STATUS='NEW')
      OPEN(5,FILE='APROX.OUT',STATUS='NEW')
      IF(ICODE.EQ.2) OPEN(6,FILE='FRAME.OUT',STATUS='NEW')
*
*     ECHO INPUT
      MM=5
      WRITE(5,99)
      IF(ICODE.EQ.2)THEN
         MM=6
         WRITE(6,99)
      ENDIF
      DO 2 NN=5,MM
      WRITE(NN,40)  FNAME
      WRITE(NN,*)
      WRITE(NN,180) CARD1
      WRITE(NN,180) CARD2
      WRITE(NN,180) CARD3
      WRITE(NN,*)
      WRITE(NN,*)
      WRITE(NN,50) IAXIS
      WRITE(NN,60) NOLAN,SLBD,SPAN,SLBW,
     *             SPANF,CAPD,SPANB,CAPW,PHI,CAPL
      WRITE(NN,70) NORAL,PDSTL,WRAIL,PAD,RALHT,HNCH
      WRITE(NN,80) NOBMF,V,BMSPF,WTRD,WBMF,CRV,BMDF,DSMPH,ECCF
      WRITE(NN,85) NOBMB,DSWIND,BMSPB,HDECK,WBMB,CMULT,BMDB,
     *             DEFL2,ECCB
      WRITE(NN,90) NOCOL,FY,WCOL,COVER,COLH,INUMB,FIXD,
     *             ISIZE,COLSP,JNUMB,R,JSIZE,D1,D2
  2   CONTINUE
*
      PHI    = PHI/57.29578
```

```
*
*
*     DEAD LOAD COMPUTATIONS FOR APPROXIMATE METHOD
*
      DLRAIL = WRAIL*SPAN*NORAL/NOCOL
      DLSLAB = SLBD*SLBW*SPAN*0.150/NOCOL
      DLBMF = NOBMF*WBMF*SPANF/NOCOL/2.
      DLBMB = NOBMB*WBMB*SPANB/NOCOL/2.
      DLCAP  = CAPW*CAPD*CAPL*0.150/NOCOL
      DLCOL  = WCOL*(COLH + FIXD)
      DLSUM  = DLRAIL + DLSLAB + DLBMF + DLBMB + DLCAP + DLCOL
*
*     DEAD LOAD COMPUTATIONS FOR BEARING ECCENTRICITY MOMENT
*
      DLF    = (WRAIL*NORAL + SLBD*SLBW*0.150 + NOBMF*WBMF)
     *          *(SPANF/2.)/NOCOL
      DLB    = (WRAIL*NORAL + SLBD*SLBW*0.150 + NOBMB*WBMB)
     *          *(SPANB/2.)/NOCOL
      ECCDL = ABS(DLF*ECCF-DLB*ECCB)/12.
*
*     LIVE LOAD COMPUTATIONS FOR APPROXIMATE METHOD
*
*     LIVE LOAD REDUCTION FACTOR
      RF = 1.
      IF(NOLAN.EQ.3) THEN
         RF = 0.90
      ELSEIF(NOLAN.GE.4) THEN
         RF = 0.75
      ENDIF
*
      RLLANE = (SPAN*0.64 + 18.)*NOLAN/NOCOL*RF
      RLLTRK = (32.*(SPAN1-14.)/SPAN1+32.
     *        +  8.*(SPAN2-14.)/SPAN2)*NOLAN/NOCOL*RF
      EMPAK = 50./(SPAN+125.)
      IF(EMPAK.GT.0.30) EMPAK = 0.30
      IF(RLLANE.GT.RLLTRK) THEN
         RLLI = RLLANE*(1.0+EMPAK)
      ELSE
         RLLI = RLLTRK*(1.0+EMPAK)
      ENDIF
*
*     LIVE LOAD COMPUTATION FOR BEARING ECCENTRICITY MOMENT
*
      FACT1  = 32.*(SPAN1-14.)/SPAN1 + 16.
      FACT2  =  8.*(SPAN2-14.)/SPAN2 + 16.
      RLL1  = RLLI*FACT1/(FACT1+FACT2)
      RLL2  = RLLI*FACT2/(FACT1+FACT2)
      IF(ECCF.LT.ECCB)THEN
         ECCLL = ABS(RLL1*ECCB-RLL2*ECCF)/12.
      ELSE
         ECCLL = ABS(RLL1*ECCF-RLL2*ECCB)/12.
      ENDIF
*
```

```
*     OUTPUT LOADING FOR APPROXIMATE METHOD
*
      WRITE(5,100) RF,DLSUM,RLLANE,RLLTRK,1.0+EMPAK,RLLI
*
*     COLUMN CAPACITY & DEFLECTION LIMIT MOMENT & CAP PROPERTIES
*
      YMOD   = 57.*SQRT(FC)
      ENRCAP = CAPW*(CAPD*12.)**3.
      ARCAP  = CAPW*CAPD*144.
      IF(R.EQ.0)THEN
         RM2DFL = 3.*YMOD*(D2*(D1*12.)**3.)*DEFL2/
     *            ((CAPD+COLH+FIXD)**2.)/12.**3.
         IF(D1.LE.D2)THEN
            ZINER = D2*(D1*12)**3.
         ELSE
            ZINER = D1*(D2*12)**3.
         ENDIF
         ENRCOL = D1*(D2*12)**3.
         ARCOL  = D1*D2*144.
      ELSE
         ZINER  = 3.1415927/4.*(R*12.)**4.
         ENRCOL = ZINER
         ARCOL  = 3.1215927*(R*12.)**2.
         RM2DFL = 3.*YMOD*ZINER*DEFL2/
     *            ((CAPD+COLH+FIXD)**2.)/12.**3.
      ENDIF
      EFFHT = COLK*(COLH+FIXD)*12.
      PC    = (3.1415927**2)*YMOD*ZINER/2.5/EFFHT**2.
      DO 3 NN=5,MM
 3    WRITE(NN,101) FC,YMOD,ZINER,COLK,CM,PC,0.7*PC,RM2DFL,
     *              ECCDL,ECCLL
*
*     FRAME ANALYSIS OPTION SELECTED WHEN ICODE = 2
*        - LATEST UPDATE REQUIRED FOR BOTH FORWARD AND BACK BEAMS
*        - TO ACCOMADATE THIS REQUIREMENT THE NODE GENERATOR IS
*          UTILIZED FOR A SECOND TIME TO PROVIDE AXIAL AND MOMENT
*          COEFFICIENTS FOR THE BACK BEAM LOCATIONS
*
      IF(ICODE.EQ.2)THEN
         NOBM = NOBMF
         BMSP = BMSPF
         CALL FRAME
         NOBM = NOBMB
         BMSP = BMSPB
         CALL FRAMEA
         CALL FRAME1
      ENDIF
*
C     TEMPORARY PRINTS FOR COLUMN AXIAL AND MOMENT COEFFICIENTS
C
C      DO 999 KKK=1,NOCOL
C      WRITE(9,*) KKK
C      DO 999 JJJ=1,6
```

```
C      WRITE(9,*) JJJ,ALOAD(1,KKK,JJJ),ALOAD(2,KKK,JJJ)
C 999  CONTINUE
C
*     CALL SUBROUTINES FOR EACH LOAD GROUP (I, II, III)
*
      CALL GROUP1
      CALL GROUP2
      IF(ICODE.EQ.2) CALL FRAME2
      CALL GROUP3
      IF(ICODE.EQ.2) CALL FRAME3
      IF(ICODE.EQ.2) CLOSE(6)
*
*.................OUTPUT SUMMARY OF COLUMNS
FORCES....................
*
      WRITE(5,902)
      DO 19 I=1,22
      IF(I.EQ.3)  WRITE(5,903)
      IF(I.EQ.13) WRITE(5,904)
      IF(I.EQ.8.OR.I.EQ.18) WRITE(5,*)
 19   WRITE(5,905) PCA2(I,1),PCA2(I,2),PCA2(I,3)
      CLOSE(5)
*
 902  FORMAT(///10X,'*************   SUMMARY OF COLUMN FORCES   ***
     **********',///
     *          25X,'Axial        Mom Perp        Mom Par'/
     *          25X,'-----------------------------------'/
     *          14X,'GROUP I')
 903  FORMAT(/14X,'GROUP II')
 904  FORMAT(/14X,'GROUP III')
 905  FORMAT(15X,3F15.0)
*
*
*...........OUTPUT PCA2 INPUT FILE (APPROXIMATE METHOD)..........
*
      WRITE(4,180) CARD1
      WRITE(4,180) CARD2
      WRITE(4,180) CARD3
      IF(D1.EQ.0.0)THEN
         WRITE(4,110) 2.*R*12.
      ELSE
         WRITE(4,120) D1*12.,D2*12.
      ENDIF
      WRITE(4,130) YMOD
      WRITE(4,140) FY
      WRITE(4,150) COVER,INUMB,ISIZE,JNUMB,JSIZE
      WRITE(4,160) IAXIS
      K=4
      DO 1 J=1,22,5
      IF(J.EQ.21) K=1
 1    WRITE(4,170) (PCA2(I,1),PCA2(I,2),PCA2(I,3),I=J,J+K)
 110  FORMAT(4X,'1',4X,'1',F20.2)
 120  FORMAT(4X,'1',4X,'2',F20.2,F10.2)
```

```
 130   FORMAT(F20.1)
 140   FORMAT(F10.1)
 150   FORMAT(4X,'1',F15.1,I5,I5,I10,I5)
 160   FORMAT(4X,'3               22',9X,I1)
 170   FORMAT(5(3F5.0))
 180   FORMAT(A80)
       CLOSE(4)
*
*.................OUTPUT PCA2 INPUT FILE (FRAME METHOD)...........

       IF(ICODE.EQ.2) THEN

       OPEN(4,FILE='FRINPUT.PCA',STATUS='NEW')
       WRITE(4,1801) CARD1
       WRITE(4,1801) CARD2
       WRITE(4,1801) CARD3
       IF(D1.EQ.0.0)THEN
          WRITE(4,1101) 2.*R*12.
       ELSE
          WRITE(4,1201) D1*12.,D2*12.
       ENDIF
       WRITE(4,1301) YMOD
       WRITE(4,1401) FY
       WRITE(4,1501) COVER,INUMB,ISIZE,JNUMB,JSIZE
       JJ = 22*NOCOL
       WRITE(4,1601) JJ,IAXIS
       K=4
       DO 4 J=1,JJ,5
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.2) K=3
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.3; K=0
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.4) K=2
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.5) K=4
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.6) K=1
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.7) K=3
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.8) K=0
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.9) K=2
       IF(J.EQ.(JJ-1).AND.NOCOL.EQ.10) K=4
 4     WRITE(4,1701) (FORCE(1,I),ABS(FORCE(2,I)),FORCE(3,I),I=J,J+K)
 1101  FORMAT(4X,'1',4X,'1',F20.2)
 1201  FORMAT(4X,'1',4X,'2',F20.2,F10.2)
 1301  FORMAT(F20.1)
 1401  FORMAT(F10.1)
 1501  FORMAT(4X,'1',F15.1,I5,I5,I10,I5)
 1601  FORMAT(4X,'3',I10,9X,I1)
 1701  FORMAT(5(3F5.0))
 1801  FORMAT(A80)
       CLOSE(4)

       ENDIF
*
*.................................................................
*
 30    FORMAT(1X,I19)
```

```
40    FORMAT('NAME OF INPUT FILE:  ',A50//)
50    FORMAT(6X,'BENDING AXIS CODE  =',I5/)
60    FORMAT(6X,'NUMBER OF LANES    =',I5,
     *      14X,'SLAB DEPTH         =',F9.3,'  FT',/
     *       6X,'SPAN LENGTH, AVG   =',F9.3,'  FT',
     *       6X,'SLAB WIDTH         =',F9.3,'  FT',/
     *       6X,'FWD SPAN LENGTH    =',F9.3,'  FT',
     *       6X,'CAP DEPTH          =',F9.3,'  FT',/
     *       6X,'BACK SPAN LENGTH   =',F9.3,'  FT',
     *       6X,'CAP WIDTH          =',F9.3,'  FT',/
     *       6X,'SKEW ANGLE         =',F9.3,'  DEG',
     *       5X,'CAP LENGTH         =',F9.3,'  FT',/)
70    FORMAT(6X,'NUMBER OF RAILS    =',I5,
     *      14X,'PEDESTAL HEIGHT    =',F9.3,'  FT',/
     *       6X,'WT/FT OF RAIL      =',F9.3,'  K/FT',
     *       4X,'BEARING PAD HEIGHT =',F9.3,'  FT',/
     *       6X,'RAIL HEIGHT        =',F9.3,'  FT',
     *       6X,'HAUNCH             =',F9.3,'  FT',/)
80    FORMAT(6X,'NUMBER BEAMS, FWD  =',I5,
     *      14X,'VELOCITY OF STREAM =',F9.3,'  FPS',/
     *       6X,'BEAM SPACING, FWD  =',F9.3,'  FT',
     *       6X,'WATER DEPTH        =',F9.3,'  FT',/
     *       6X,'WT/FT OF BEAM. FWD =',F9.3,'  K/FT',
     *       4X,'DEGREE OF CURVE    =',F9.3,'  DEG',/
     *       6X,'BEAM DEPTH, FWD    =',F9.3,'  FT',
     *       6X,'DESIGN SPEED       =',F9.3,'  MPH',/
     *       6X,'BEARING ECC. FWD   =',F9.3,'  IN',/)
85    FORMAT(6X,'NUMBER BEAMS, BACK =',I5,
     *      14X,'TEXAS DESIGN WIND  =',F9.3,'  MPH',/
     *       6X,'BEAM SPACING, BACK =',F9.3,'  FT',
     *       6X,'AASHTO HT ABOVE DK =',F9.3,'  FT',/
     *       6X,'WT/FT OF BEAM, BK  =',F9.3,'  K/FT',
     *       4X,'BENT CONT. MULT.   =',F9.3,/
     *       6X,'BEAM DEPTH, BACK   =',F9.3,'  FT',
     *       6X,'OUT OF PLANE DEFL  =',F9.3,'  IN',/
     *       6X,'BEARING ECC, BACK  =',F9.3,'  IN',/)
90    FORMAT(6X,'NUMBER OF COLUMNS  =',I5,
     *      14X,'STEEL YIELD STRESS =',F9.3,'  KSI',/
     *       6X,'WT/FT OF COLUMN    =',F9.3,'  K/FT',
     *       4X,'UNIFORM SIZE COVER =',F9.3,'  IN',/
     *       6X,'COLUMN HEIGHT      =',F9.3,'  FT',
     *       6X,'LOWER LIMIT NUMBER =',I5,/
     *       6X,'DEPTH TO FIXITY    =',F9.3,'  FT',
     *       6X,'LOWER LIMIT SIZE   =',I5,/
     *       6X,'COLUMN SPACING     =',F9.3,'  FT',
     *       6X,'UPPER LIMIT NUMBER =',I5,/
     *       6X,'COLUMN RADIUS      =',F9.3,'  FT',
     *       6X,'UPPER LIMIT SIZE   =',I5,/
     *       6X,'COL. D PERP TO CAP =',F9.3,'  FT',/
     *       6X,'COL. D ALONG CAP   =',F9.3,'  FT',///)
100    FORMAT(/4X,'DEAD LOAD AND LIVE LOAD'//
     *       10X,'REDUCTION FACTOR   =',F9.3/
     *       10X,'DEAD LOAD PER COL  =',F9.3,'  KIPS',/
```

```
     *        10X,'LL LANE             =',F9.3,'  KIPS',/
     *        10X,'LL TRUCK            =',F9.3,'  KIPS',/
     *        10X,'IMPACT FACTOR       =',F9.3/
     *        10X,'LL+I                =',F9.3,'  KIPS',//)
 101  FORMAT(/4X,'COLUMN CAPACITY, DEFL. LIMIT MOMENT,
     *& BEARING ECCENTRICITY MOMENTS'//
     *        10X,'Fc OF COLUMN        =',F9.1,'  PSI',/
     *        10X,'MODULUS OF ELAS     =',F9.1,'  KSI',/
     *        10X,'MOMENT OF INERTIA   =',F9.1,'  IN**4',/
     *        10X,'K-FACTOR            =',F9.2/
     *        10X,'Cm                  =',F9.2/
     *        10X,'Pc                  =',F9.1,'  KIPS',/
     *        10X,'0.7*Pc              =',F9.1,'  KIPS',/
     *        10X,'DEFL LIM MOM PAR    =',F9.1,'  FT-KIP',/
     *        10X,'DL ECCEN. MOMENT    =',F9.1,'  FT-KIP',/
     *        10X,'LL ECCEN. MOMENT    =',F9.1,'  FT-KIP',//)
 99   FORMAT('***************************************************',
     *        '***************************************'///
     *        '        TEXAS STATE DEPARTMENT OF HIGHWAYS & PUBLIC
     * TRANSPORTATION:  BRIDGE DIVISION'//
     *        '        COLUMN ANALYSIS:  GROUPS I,II, & III
     *            (AASHTO SPECIFICATIONS)'//
     *        '        AUGUST 1987
     *                        VERSION 1.00'///
     *        '***************************************************',
     *        '***************************************'//)
      END
*
*.........................GROUP I (APPROXIMATE)....................
*
      SUBROUTINE GROUP1
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
*
*     CENTRIFUGAL FORCE
      C       = .00117*(DSMPH**2)*CRV
      CF      = C/100.*RLLTRK
      H1      = (COLH+FIXD)/2.+CAPD+PDSTL+PAD+BMD+HNCH+SLBD+HDECK
      CFMOM1 = COS(PHI)*CF*H1
      H2      = (COLH+FIXD)/2.+H1
      CFMOM2 = SIN(PHI)*CF*H2
      WRITE(5,20) C,CF,H1,CFMOM1,H2,CFMOM2
*
*     STREAM FLOW
      IF(R.EQ.0)THEN
         P       = (1.375)*V*V/1000.
```

```
         SFMOM1 = P*D1*WTRD*(WTRD/2.)
      ELSE
         P      = (2./3.)*V*V/1000.
         SFMOM1 = P*(2.*R)*WTRD*(WTRD/2.)
      ENDIF
      WRITE(5,30) P,SFMOM1
*
*     MOMENT AND AXIAL LOAD
      BD = 1.0
      DO 1 I=1,2
      IF(I.EQ.2) BD = 0.75
      PU    = 1.3*BD*DLSUM + 2.17*RLLI
      DELTA = CM/(1.-PU/(.7*PC))
      RM1U  = 1.3*DELTA*(CFMOM1 + SFMOM1)
      RM2U  = 1.3*DELTA*(CFMOM2)
      WRITE(5,40) BD,PU,DELTA,RM1U,RM2U
*
*     COMPARE COMPUTED MOMENTS AGAINST MINIMUM ECCENTRICITY
      IF(R.EQ.0)THEN
         RMIN1   = DELTA*0.1*D2*PU
         RMIN2   = DELTA*0.1*D1*PU
         WRITE(5,50) RMIN1,RMIN2
         IF(RMIN1.GT.RM1U) RM1U = RMIN1
         IF(RMIN2.GT.RM2U) RM2U = RMIN2
      ELSE
         IF(IAXIS.EQ.1) RM2U = 0.
         IF(IAXIS.EQ.2) RM1U = 0.
         RMIN    = DELTA*0.1*(2.*R)*PU
         RESMU   = SQRT(RM1U*RM1U+RM2U*RM2U)
         WRITE(5,60) RMIN,RESMU
         IF(RMIN.GT.RESMU)THEN
            RM1U = RMIN/SQRT(2.0)
            RM2U = RM1U
            IF(IAXIS.EQ.1) RM1U = RMIN
            IF(IAXIS.EQ.2) RM2U = RMIN
         ENDIF
      ENDIF
*
*     OUTPUT GROUP I LOADING
      IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
      IF(IAXIS.EQ.1) RM2U = 0.
      IF(IAXIS.EQ.2) RM1U = 0.
      WRITE(5,70) RM1U,RM2U + (1.3*ECCDL + 2.17*ECCLL)*DELTA
      PCA2(I,1) = PU+.49
      PCA2(I,2) = RM1U+.49
      PCA2(I,3) = RM2U+.49 + (1.3*ECCDL + 2.17*ECCLL)*DELTA
 1    CONTINUE
*
 20   FORMAT(//4X,'GROUP I LOADING'//
     *       10X,'CENTRIFUGAL FORCE (GROUP I)'//
     *       14X,'C      =',F7.2,5X,'C.F.         =',F9.3//
     *       14X,'H Perp =',F7.2,5X,'CF Mom Perp =',F9.3/
     *       14X,'H Par  =',F7.2,5X,'CF Mom Par  =',F9.3//)
```

```
 30   FORMAT(10X,'STREAM FLOW (GROUP I)'//
     *        14X,'P       =',F7.3,5X,'SF Mom Perp =',F9.3//)
 40   FORMAT(/12X,'***************** Bd = ',F4.2,'
****************'//
     *        14X,'Pu    =',F9.3,5X,'Mom Mag      =',F9.3//
     *        14X,'Sum Mom Perp =',F9.3/
     *        14X,'Sum Mom Par  =',F9.3//)
 50   FORMAT(14X,'MINIMUM ECCENTRICITY MOMENTS'//
     *        18X,'Mom Perp =',F9.3/
     *        18X,'Mom Par  =',F9.3//)
 60   FORMAT(14X,'MINIMUM ECCENTRICITY MOMENT'//
     *        18X,'Min Mom  =',F9.3/
     *        18X,'Resultant=',F9.3//)
 70   FORMAT(14X,'CONTROLLING MOMENTS (GROUP I)'//
     *        18X,'Mom Perp =',F9.3/
     *        18X,'Mom Par  =',F9.3/)
      RETURN
      END
*
*........................GROUP II (APPROXIMATE).................
*
      SUBROUTINE GROUP2
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
      COMMON/TEN/ SPANF,SPANB
*
*     GROUP II WIND LOADING, 3.15.2.1.1
      WSKEW(1) =  0.
      WLAT(1)  =  0.050
      WLON(1)  =  0.
      WSKEW(2) = 15.
      WLAT(2)  =  0.044
      WLON(2)  =  0.006
      WSKEW(3) = 30.
      WLAT(3)  =  0.041
      WLON(3)  =  0.012
      WSKEW(4) = 45.
      WLAT(4)  =  0.033
      WLON(4)  =  0.016
      WSKEW(5) = 60.
      WLAT(5)  =  0.017
      WLON(5)  =  0.019
*
*     WIND ON SUPERSTRUCTURE
      WAREA=(RALHT+SLBD+HNCH+PAD+PDSTL)*SPAN
      WAREA=WAREA + BMDF*(SPANF/2.)
```

```
      WAREA=(WAREA + BMDB*(SPANB/2.))*CMULT
      H1   =(RALHT+SLBD+HNCH+BMD+PAD+PDSTL)/2.+CAPD+(COLH+FIXD)/2.
      H2   =H1+(COLH+FIXD)/2.
      WRITE(5,10) WAREA,H1,H2
      WRITE(5,20)
      DO 1 I=1,5
      RMOM1(I)=(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))*WAREA*H1/NOCOL
      RMOM2(I)=(WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))*WAREA*H2/NOCOL
      WRITE(5,30) WSKEW(I),RMOM1(I),RMOM2(I)
 1    CONTINUE
*
*     WIND ON CAP
      H1 = (CAPD/2.)+(COLH+FIXD)/2.
      H2 = H1+(COLH+FIXD)/2.
      WRITE(5,40) CAPD*CAPW,CAPD*CAPL,H1,H2
      WRITE(5,20)
      DO 2 I=1,5
         THETA     = PHI+WSKEW(I)/57.29578
         TEMP1     = COS(THETA)*0.04*CAPD*CAPW*H1/NOCOL
         TEMP2     = SIN(THETA)*0.04*CAPD*CAPL*H2/NOCOL
         RMOM1(I)  = RMOM1(I)+TEMP1
         RMOM2(I)  = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
 2    CONTINUE
*
*     WIND ON COLUMN
      H1 = COLH/2.
      H2 = COLH/2.+FIXD
      IF(R.EQ.0)THEN
         COLA1 = COLH*D1
         COLA2 = COLH*D2
         IF(IAXIS.EQ.2) COLA2 = ((NOCOL-1)*COLSP+D2)*COLH/NOCOL
      ELSE
         COLA1 = COLH*2.*R
         COLA2 = COLH*2.*R
         IF(IAXIS.EQ.2) COLA2 = ((NOCOL-1)*COLSP+2.*R)*COLH/NOCOL
      ENDIF
      WRITE(5,50) COLA1,COLA2,H1,H2
      WRITE(5,20)
*
*     TEST WATER PRESSURE AGAINST WIND (40 PSF)
      DO 3 I=1,5
         THETA     = PHI+WSKEW(I)/57.29578
         TEMP1     = COS(THETA)*COLA1*0.04*H1
         TEMP2     = SIN(THETA)*COLA2*0.04*H2
         RMOM1(I)  = RMOM1(I)+TEMP1
         RMOM2(I)  = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
         SFWD2(I)  = 0.
         SFWD3(I)  = 0.
         IF(SFMOM1.GT.TEMP1)      SFWD2(I)  = SFMOM1-TEMP1
         IF(SFMOM1.GT.TEMP1*0.3)  SFWD3(I)  = SFMOM1-TEMP1*0.3
 3    CONTINUE
```

```
*
*     WIND REDUCTION FACTOR
      WRF = DSWIND*DSWIND/10000.
      WRITE(5,60) WRF
*
*     OUTPUT SUM WIND MOMENTS
      DO 4 I=1,5
      WRITE(5,70) WSKEW(I),RMOM1(I),RMOM2(I),RMOM1(I)*WRF,
     *            RMOM2(I)*WRF
 4    CONTINUE
*
*     OUTPUT ADDITIONAL STREAM FLOW MOMENT PERP
      WRITE(5,80)
      WRITE(5,20)
      DO 5 I=1,5
      WRITE(5,30) WSKEW(I),SFWD2(I)
 5    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD  = 1.0
      K   = 2
      DO 6 J=1,2
      IF(J.EQ.2)THEN
         BD  = 0.75
         K   = 7
      ENDIF
      PU    = 1.3*BD*DLSUM
      DELTA = CM/(1.-PU/(.7*PC))
      WRITE(5,90) BD,PU,DELTA
      WRITE(5,20)
      DO 7 I=1,5
         RM1U = 1.3*DELTA*(RMOM1(I)*WRF+SFWD2(I))
         RM2U = 1.3*DELTA*RMOM2(I)*WRF
         IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
         IF(IAXIS.EQ.1) RM2U = 0.
         IF(IAXIS.EQ.2) RM1U = 0.
         WRITE(5,30) WSKEW(I),RM1U,RM2U + (1.3*ECCDL)*DELTA
         PCA2(I+K,1) = PU+.49
         PCA2(I+K,2) = RM1U+.49
         PCA2(I+K,3) = RM2U+.49 + (1.3*ECCDL)*DELTA
 7    CONTINUE
 6    CONTINUE
 10   FORMAT(//4X,'GROUP II LOADING'///
     *       10X,'WIND ON SUPERSTRUCTURE (GROUP II)'//
     *       14X,'Wind Area Per Bent =',F8.2//
     *       14X,'H Perp             =',F8.2/
     *       14X,'H Par              =',F8.2//)
 20   FORMAT(14X,'Skew',6X,'Mom Perp',7X,'Mom Par'/
     *       14X,'--------------------------------')
 30   FORMAT(14X,F4.0,2F14.3)
 40   FORMAT(//10X,'WIND ON CAP (GROUP II)'//
     *       14X,'Cap Area Perp      =',F8.2/
     *       14X,'Cap Area Par       =',F8.2//
     *       14X,'H Perp             =',F8.2/
```

```
     *        14X,'H Par                =',F8.2/)
 50   FORMAT(//10X,'WIND ON COLUMN (GROUP II)'//
     *        14X,'Column Area Perp   =',F8.2/
     *        14X,'Column Area Par    =',F8.2//
     *        14X,'H Perp             =',F8.2/
     *        14X,'H Par              =',F8.2/)
 60   FORMAT(//10X,'WIND REDUCTION (GROUP II)'//
     *        14X,'Wind Red. Factor   =',F8.3//
     *        29X,'100 MPH Wind',16X,'Reduced Wind'//
     *        14X,'Skew',6X,'Mom Perp',7X,'Mom Par',
     *                6X,'Mom Perp',7X,'Mom Par'/
     *        14X,'--------------------------------',
     *         6X,'---------------------')
 70   FORMAT(14X,F4.0,4F14.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW MOMENT PERP (GROUP
II)'/)
 90   FORMAT(//12X,'***************** Bd = ',F4.2,'
****************'//
     *        14X,'Pu   =',F9.3,5X,'Mom Mag     =',F9.3/)
      RETURN
      END
*
*.......................GROUP III (APPROXIMATE)...................
*
      SUBROUTINE GROUP3
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
*     GROUP III WIND LOADING, 3.15.2.1.2
      WLAT(1) = 0.100
      WLON(1) = 0.
      WLAT(2) = 0.088
      WLON(2) = 0.012
      WLAT(3) = 0.082
      WLON(3) = 0.024
      WLAT(4) = 0.066
      WLON(4) = 0.032
      WLAT(5) = 0.034
      WLON(5) = 0.038
*
*     0.3 WIND (USING SUM, REDUCED WIND FROM GROUP II)
      WRITE(5,10)
      WRITE(5,20)
      DO 1 I=1,5
         RMOM1(I) = RMOM1(I)*0.3
         RMOM2(I) = RMOM2(I)*0.3
```

```
         WRITE(5,30) WSKEW(I),RMOM1(I),RMOM2(I)
 1    CONTINUE
*
*     WIND ON LIVE LOAD (WL)
      H1 = (COLH+FIXD)/2.+CAPD+SLBD+HNCH+BMD+PAD+PDSTL+HDECK
      H2 = H1+(COLH+FIXD)/2.
      WRITE(5,40) H1,H2
      WRITE(5,20)
      DO 2 I=1,5
         TEMP1     = (WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *               *SPAN*H1/NOCOL
         TEMP2     = (WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))
     *               *SPAN*H2/NOCOL
         RMOM1(I) = RMOM1(I)+TEMP1
         RMOM2(I) = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
 2    CONTINUE
*
*     LONGITUDINAL FORCE (LF)
      RM1LF = SIN(PHI)*0.05*RLLANE*H1
      RM2LF = COS(PHI)*0.05*RLLANE*H2
      WRITE(5,50) H1,RM1LF,H2,RM2LF
      DO 3 I=1,5
         RMOM1(I) = RMOM1(I)+RM1LF
         RMOM2(I) = RMOM2(I)+RM2LF
 3    CONTINUE
*
*     OUTPUT ADDITIONAL STREAM FLOW MOMENT PERP
      WRITE(5,60)
      WRITE(5,20)
      DO 4 I=1,5
         WRITE(5,30) WSKEW(I),SFWD3(I)
 4    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD  = 1.0
      K   = 12
      DO 5 J=1,2
         IF(J.EQ.2)THEN
            BD  = 0.75
            K   = 17
         ENDIF
         PU    = 1.3*(BD*DLSUM+RLLI)
         DELTA = CM/(1.-PU/(.7*PC))
         WRITE(5,70) BD,PU,DELTA
         WRITE(5,20)
         DO 6 I=1,5
            RM1U = 1.3*DELTA*(RMOM1(I)+SFWD3(I)+CFMOM1)
            RM2U = 1.3*DELTA*(RMOM2(I)+CFMOM2)
            IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
            IF(IAXIS.EQ.1) RM2U = 0.
            IF(IAXIS.EQ.2) RM1U = 0.
            WRITE(5,30) WSKEW(I),RM1U,RM2U
     *                                  + (1.3*(ECCDL+ECCLL))*DELTA
```

```
            PCA2(I+K,1) = PU+.49
            PCA2(I+K,2) = RM1U+.49
            PCA2(I+K,3) = RM2U+.    + (1.3*(ECCDL+ECCLL))*DELTA
 6        CONTINUE
 5      CONTINUE
10      FORMAT(///4X,'GROUP III LOADING'//
     *         10X,'0.3 WIND (GROUP III)'/)
20      FORMAT(14X,'Skew',6X,'Mom Perp',7X,'Mom Par'/
     *         14X,'--------------------------------')
30      FORMAT(14X,F4.0,2F14.3)
40      FORMAT(//10X,'WIND ON LIVE LOAD (GROUP III)'//
     *         14X,'H Perp              =',F8.2/
     *         14X,'H Par               =',F8.2/)
50      FORMAT(//10X,'LONGITUDINAL FORCE (GROUP III)'//
     *         14X,'H Perp =',F7.2,5X,'LF Mom Perp =',F9.3/
     *         14X,'H Par  =',F7.2,5X,'LF Mom Par  =',F9.3)
60      FORMAT(//10X,'ADDITIONAL STREAM FLOW MOMENT PERP (GROUP
III)'/)
70      FORMAT(//12X,'***************** Bd = ',F4.2,'
****************'//
     *         14X,'Pu   =',F9.3,5X,'Mom Mag     =',F9.3/)
        RETURN
        END
*
*.........................GROUP I (FRAME ANALYSIS)................
*
        SUBROUTINE FRAME1
        COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
        COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
        COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
        COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
        COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
        COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
        COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
        COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
        COMMON/TEN/ SPANF,SPANB
*
        COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
        COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *              FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
        COMMON/FR3/ TFORCE(3,20)
*
*       COMPUTE DEAD AND LIVE LOAD PER BEAM
        DL1    = (WRAIL*(SPANF/2.0)*NORAL + SLBD*SLBW*
     *              (SPANF/2.0)*.15)/NOBMF + WBMF*(SPANF/2.0)
        DL2    = (WRAIL*(SPANB/2.0)*NORAL + SLBD*SLBW*
     *              (SPANB/2.0)*.15)/NOBMB + WBMB*(SPANB/2.0)
*
        DLCOL = WCOL*(COLH+FIXD)
        DL5   = CAPW*CAPD*.0125
        FRLLI = RLLI*NOCOL
        FTRK  = RLLTRK*NOCOL
        FLANE = RLLANE*NOCOL
```

```
*
      NOBM = NOBMF
      IF(NOBMF.LT.NOBMB) NOBM = NOBMB
*
*     CENTRIFUGAL FORCE
      C      = .00117*(DSMPH**2)*CRV
      CF     = C/100.*FTRK
      H1     = (COLH+FIXD)+CAPD+PDSTL+PAD+BMD+HNCH+SLBD+HDECK
      CFMOM  = COS(PHI)*CF*H1/Y2
      H2     = H1
      CFMOM2 = SIN(PHI)*CF/NOCOL*H2
*
*     STREAM FLOW
      IF(R.EQ.0)THEN
         P      = (1.375)*V*V/1000.
         SFMOM  = P*D1/12.
      ELSE
         P      = (2./3.)*V*V/1000.
         SFMOM  = P*(2.*R)/12.
      ENDIF
*
*     AXIAL LOAD AND MOMENTS
      WRITE(6,10)
      BD = 1.0
      DO 1 I=1,2
      IF(I.EQ.2) BD = 0.75
      WRITE(6,20) BD
      PMAX = 0.0
      DO 2 J=1,NOCOL
      FORCE(1,J)=1.3*(BD*(DL1*ALOAD(1,J,1)+DL2*ALOAD(1,J,6)+
     *              DL5*ALOAD(1,J,5)+DLCOL)+
     *             1.67*FRLLI/NOBM*ALOAD(1,J,1)+CFMOM*ALOAD(1,J,2)+
     *              SFMOM*ALOAD(1,J,4))
 2    IF(FORCE(1,J).GE.PMAX) PMAX = FORCE(1,J)
      DELTA = CM/(1.-PMAX/(.7*PC))
      DO 3 J=1,NOCOL
      FORCE(2,J)=-1.3*DELTA*(DL1*ALOAD(2,J,1)+DL2*ALOAD(2,J,6)
     *          +DL5*ALOAD(2,J,5)
     *          +1.67*FRLLI/NOBM*ALOAD(2,J,1)+CFMOM*ALOAD(2,J,2)
     *          +SFMOM*ALOAD(2,J,4))
 3    CONTINUE
      RM2U  = 1.3*DELTA*(CFMOM2)
*
*     COMPARE COMPUTED MOMENTS AGAINST MINIMUM ECCENTRICITY
      DO 4 J=1,NOCOL
      IF(R.EQ.0)THEN
         RMIN1    = DELTA*0.1*D2*FORCE(1,J)
         RMIN2    = DELTA*0.1*D1*FORCE(1,J)
         IF(RMIN1.GT.FORCE(2,J)) FORCE(2,J)  = RMIN1
         IF(RMIN2.GT.RM2U) RM2U = RMIN2
      ELSE
         IF(IAXIS.EQ.1) RM2U = 0.
         RMIN     = DELTA*0.1*(2.*R)*FORCE(1,J)
```

```
            RESMU   = SQRT(FORCE(2,J)*FORCE(2,J)+RM2U*RM2U)
            IF(RMIN.GT.RESMU)THEN
               FORCE(2,J) = RMIN/SQRT(2.0)
               RM2U = FORCE(2,J)
               IF(IAXIS.EQ.1) FORCE(2,J) = RMIN
               IF(IAXIS.EQ.2) RM2U = RMIN
            ENDIF
         ENDIF
*
*        OUTPUT GROUP I LOADING
         IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
         IF(IAXIS.EQ.1) RM2U = 0.0
 4       WRITE(6,30) J,FORCE(1,J),FORCE(2,J),
     *                  RM2U + (1.3*ECCDL + 2.17*ECCLL)*DELTA
*
         KK  = 1
         JJJ = 1
         IF(I.EQ.2) KK=NOCOL+1
         DO 5 JJ=KK,NOCOL*I
         TFORCE(1,JJ) = FORCE(1,JJJ)
         TFORCE(2,JJ) = FORCE(2,JJJ)
         JJJ=JJJ+1
 5       FORCE(3,JJ) = RM2U + (1.3*ECCDL + 2.17*ECCLL)*DELTA
*
 1       CONTINUE
*
         DO 6 I=1,2*NOCOL
         FORCE(1,I) = TFORCE(1,I)
  6      FORCE(2,I) = TFORCE(2,I)
*
 10      FORMAT(4X,'GROUP I LOADING')
 20      FORMAT(//12X,'***************** Bd = ',F4.2,'
***************'///
     *         14X,'COL NO.    AXIAL     MOM PERP     MOM PAR'/
     *         14X,'----------------------------------------')
 30      FORMAT(14X,I4,3F12.2)
         RETURN
         END
*
*.....................GROUP II (FRAME ANALYSIS)..................
*
         SUBROUTINE FRAME2
         COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
         COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
         COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
         COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
         COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
         COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
         COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
         COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
         COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
         COMMON/TEN/ SPANF,SPANB
*
```

```
      COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
      COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *            FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
*
*
*     WIND ON SUPERSTRUCTURE
      WRITE(6,11)
      WRITE(6,20)
      WAREA=(RALHT+SLBD+HNCH+PAD+PDSTL)*SPAN
      WAREA=WAREA + BMDF*(SPANF/2.)
      WAREA=(WAREA + BMDB*(SPANB/2.))*CMULT
      H1   =(RALHT+SLBD+HNCH+BMD+PAD+PDSTL)/2.+CAPD+(COLH+FIXD)
      DO 1 J=1,NOCOL
      WRITE(6,*)
      DO 1 I=1,5
      FRMOM1(I,J) = -(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *                *WAREA*H1/Y2*ALOAD(2,J,2)
      AXIAL(I,J)  =  (WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *                *WAREA*H1/Y2*ALOAD(1,J,2)
      WRITE(6,30) J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J)
 1    CONTINUE
*
*     WIND ON CAP
      WRITE(6,40)
      WRITE(6,20)
      H1 = (CAPD/2.)+(COLH+FIXD)
      DO 2 J=1,NOCOL
      WRITE(6,*)
      DO 2 I=1,5
         THETA     = PHI+WSKEW(I)/57.29578
         TEMP1     = -COS(THETA)*0.04*CAPD*CAPW*H1/Y2*ALOAD(2,J,2)
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
         TEMP2     =  COS(THETA)*0.04*CAPD*CAPW*H1/Y2*ALOAD(1,J,2)
         AXIAL(I,J)  = AXIAL(I,J)+TEMP2
         WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
 2    CONTINUE
*
*     WIND ON COLUMN
      WRITE(6,50)
      WRITE(6,20)
      IF(R.EQ.0)THEN
         COLA1 = D1/12.
      ELSE
         COLA1 = 2.*R/12.
      ENDIF
*
*     TEST WATER PRESSURE AGAINST WIND (40 PSF)
      DO 3 J=1,NOCOL
      WRITE(6,*)
      DO 3 I=1,5
         THETA     = PHI+WSKEW(I)/57.29578
         TEMP1     = -COS(THETA)*COLA1*.04*ALOAD(2,J,3)
         TEMP2     =  COS(THETA)*COLA1*.04*ALOAD(1,J,3)
```

```
          FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
          AXIAL(I,J)  = AXIAL(I,J)+TEMP2
          WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
          FSFWD2(I,J) = 0.
          FSFAX2(I,J) = 0.
          FSFWD3(I,J) = 0.
          FSFAX3(I,J) = 0.
          FSF   = -SFMOM*ALOAD(2,J,4)
          FSFAX =  SFMOM*ALOAD(1,J,4)
          IF(FSF.GT.TEMP1)THEN
             FSFWD2(I,J) = FSF-TEMP1
             FSFAX2(I,J) = FSFAX-TEMP2
          ENDIF
          IF(FSF.GT.TEMP1*0.3)THEN
             FSFWD3(I,J) = FSF-TEMP1*0.3
             FSFAX3(I,J) = FSFAX-TEMP2*0.3
          ENDIF
 3     CONTINUE
*
*      WIND REDUCTION FACTOR
       WRF = DSWIND*DSWIND/10000.
*
*      OUTPUT SUM WIND MOMENTS
       WRITE(6,60)
       DO 4 J=1,NOCOL
       WRITE(6,*)
       DO 4 I=1,5
       WRITE(6,70)J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J),AXIAL(I,J)*WRF,
      *                                              FRMOM1(I,J)*WRF
 4     CONTINUE
*
*      OUTPUT ADDITIONAL STREAM FLOW FORCES
       WRITE(6,80)
       WRITE(6,20)
       DO 5 J=1,NOCOL
       WRITE(6,*)
       DO 5 I=1,5
       WRITE(6,30) J,WSKEW(I),FSFAX2(I,J),FSFWD2(I,J)
 5     CONTINUE
*      AXIAL LOAD AND MOMENT MAGNIFIER
       BD  = 1.0
       K   = 2*NOCOL
       N   = K
       DO 6 J=1,2
       IF(J.EQ.2)THEN
          BD  = 0.75
          K   = 7*NOCOL
          N   = K
       ENDIF
       WRITE(6,*)
       WRITE(6,*)
       WRITE(6,90) BD
       PMAX  = 0.0
```

```
      DO 8 L=1,NOCOL
      DO 9 M=1,5
      FORCE(1,M+K) = 1.3*(BD*(DL1*ALOAD(1,L,1)+DL2*ALOAD(1,L,6)
     *               +DL5*ALOAD(1,L,5)
     *               +DLCOL)+AXIAL(M,L)+FSFAX2(M,L))
 9    IF(FORCE(1,M+K).GE.PMAX) PMAX = FORCE(1,M+K)
 8    K=K+5
      DELTA = CM/(1.-PMAX/(.7*PC))
      IF(J.EQ.1)THEN
         DELMY = DELTA/(CM/(1.-PCA2(3,1)/(.7*PC)))
      ELSE
         DELMY = DELTA/(CM/(1.-PCA2(8,1)/(.7*PC)))
      ENDIF
      DO 7 L=1,NOCOL
      WRITE(6,*)
      IF(J.EQ.1) LOC=3
      IF(J.EQ.2) LOC=8
      DO 10 M=1,5
         FORCE(2,M+N) = 1.3*DELTA*(-DL1*ALOAD(2,L,1)
     *                  -DL2*ALOAD(2,L,6)-DL5*ALOAD(2,L,5)
     *                  +FRMOM1(M,L)*WRF+FSFWD2(M,L))
         FORCE(3,M+N) = DELMY*PCA2(LOC,3)
         LOC=LOC+1
 10      WRITE(6,100) L,WSKEW(M),FORCE(1,M+N),FORCE(2,M+N),
     *                                     FORCE(3,M+N)
 7    N=N+5
 6    CONTINUE
 11   FORMAT(///4X,'GROUP II LOADING'///
     *       10X,'WIND ON SUPERSTRUCTURE (GROUP II)'/)
 20   FORMAT(14X,'Col No.',6X,'Skew',6X,' Axial ',5X,'Mom Perp'/
     *       14X,'-----------------------------------------')
 30   FORMAT(14X,I4,9X,F4.0,F12.3,F13.3)
 40   FORMAT(//10X,'WIND ON CAP (GROUP II)'/)
 50   FORMAT(//10X,'WIND ON COLUMN (GROUP II)'/)
 60   FORMAT(//10X,'WIND REDUCTION (GROUP II)'//
     *       40X,'100 MPH Wind',16X,'Reduced Wind'//
     *       14X,'Col No.',4X,'Skew',7X,' Axial ',5X,'Mom Perp',
     *                   7X,' Axial ',5X,'Mom Perp'/
     *       14X,'-----------------------------------------',
     *        8X,'-------------------')
 70   FORMAT(14X,I4,7X,F4.0,2F13.3,1X,2F13.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW FORCES (GROUP II)'/)
 90   FORMAT(///12X,'******************** Bd = ',F4.2,'
************
     **********'///
     *      14X,'Col No.   Skew       Axial     Mom Perp     Mom
Par'/
     *      14X,'---------------------------------------------------
-')
 100  FORMAT(14X,I4,6X,F4.0,3F12.2)
      RETURN
      END
*
```

```
*.......................GROUP III (FRAME ANALYSIS)................
*
      SUBROUTINE FRAME3
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCM2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
      COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *            FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
*
*     0.3 WIND (USING SUM WIND FROM FRAME GROUP II)
      WRITE(6,11)
      WRITE(6,20)
      DO 1 J=1,NOCOL
      WRITE(6,*)
      DO 1 I=1,5
         FRMOM1(I,J) = FRMOM1(I,J)*0.3
         AXIAL(I,J)  = AXIAL(I,J)*0.3
         WRITE(6,30) J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J)
 1    CONTINUE
*
*     WIND ON LIVE LOAD (WL)
      WRITE(6,40)
      WRITE(6,20)
      H1 = (COLH+FIXD)+CAPD+SLBD+HNCH+BMD+PAD+PDSTL+HDECK
      DO 2 J=1,NOCOL
      WRITE(6,*)
      DO 2 I=1,5
         TEMP1      =-(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *                *SPAN*H1/Y2*ALOAD(2,J,2)
         TEMP2      = (WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))
     *                *SPAN*H1/Y2*ALOAD(1,J,2)
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
         AXIAL(I,J)  = AXIAL(I,J)+TEMP2
         WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
 2    CONTINUE
*
*     LONGITUDINAL FORCE (LF)
      WRITE(6,50)
      WRITE(6,60)
      DO 3 J=1,NOCOL
      TEMP1 =-SIN(PHI)*0.05*FLANE*H1/Y2*ALOAD(2,J,2)
      TEMP2 = SIN(PHI)*0.05*FLANE*H1/Y2*ALOAD(1,J,2)
      WRITE(6,65) J,TEMP2,TEMP1
      DO 3 I=1,5
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
```

```
         AXIAL(I,J)   = AXIAL(1,J)+TEMP2
  3   CONTINUE
*
*
*     OUTPUT ADDITIONAL STREAM FLOW FORCES
      WRITE(6,80)
      WRITE(6,20)
      DO 5 J=1,NOCOL
      WRITE(6,*)
      DO 5 I=1,5
      WRITE(6,30) J,WSKEW(I),FSFAX3(I,J),FSFWD3(I,J)
  5   CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD   = 1.0
      K    = 12*NOCOL
      N    = K
      DO 6 J=1,2
      IF(J.EQ.2)THEN
         BD   = 0.75
         K    = 17*NOCOL
         N    = K
      ENDIF
      WRITE(6,90) BD
      PMAX  = 0.0
      DO 8 L=1,NOCOL
      DO 9 M=1,5
      FORCE(1,M+K)  = 1.3*(BD*(DL1*ALOAD(1,L,1)+DL2*ALOAD(1,L,6)
     *         +DL5*ALOAD(1,L,5)
     *         +DLCOL)+AXIAL(M,L)+FSFAX3(M,L)
     *         +FRLLI/NOBM*ALOAD(1,L,1)+CFMOM*ALOAD(1,L,2))
  9   IF(FORCE(1,M+K).GE.PMAX) PMAX = FORCE(1,M+K)
  8   K=K+5
      DELTA = CM/(1.-PMAX/(.7*PC))
      IF(J.EQ.1)THEN
         DELMY = DELTA/(CM/(1.-PCA2(13,1)/(.7*PC)))
      ELSE
         DELMY = DELTA/(CM/(1.-PCA2(18,1)/(.7*PC)))
      ENDIF
      DO 7 L=1,NOCOL
      WRITE(6,*)
      IF(J.EQ.1) LOC=13
      IF(J.EQ.2) LOC=18
      DO 10 M=1,5
      FORCE(2,M+N)  = 1.3*DELTA*(-DL1*ALOAD(2,L,1)-DL2*ALOAD(2,L,6)
     *                -DL5*ALOAD(2,L,5)
     *                -FRLLI/NOBM*ALOAD(2,L,1)-CFMOM*ALOAD(2,L,2)
     *                +FSFWD3(M,L)+FRMOM1(M,L))
      FORCE(3,M+N)  = DELMY*PCA2(LOC,3)
      LOC=LOC+1
 10   WRITE(6,100) L,WSKEW(M),FORCE(1,M+N),FORCE(2,M+N),
     *                                     FORCE(3,M+N)
  7   N=N+5
  6   CONTINUE
```

```
 11   FORMAT(///4X,'GROUP III LOADING'///
     *        10X,'0.3 WIND (GROUP III)'/)
 20   FORMAT(14X,'Col No.',6X,'Skew',6X,' Axial ',5X,'Mom Perp'/
     *        14X,'------------------------------------------')
 30   FORMAT(14X,I4,9X,F4.0,F12.3,F13.3)
 40   FORMAT(//10X,'WIND ON LIVE LOAD (GROUP III)'/)
 50   FORMAT(//10X,'LONGITUDINAL FORCE (GROUP III)'/)
 60   FORMAT(14X,'Col No.',6X,'Axial',5X,'Mom Perp'/
     *        14X,'-----------------------------')
 65   FORMAT(/14X,I4,2X,2F12.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW FORCES (GROUP III)'/)
 90   FORMAT(///12X,'********************* Bd = ',F4.2,'
************
     **********'///
     *       14X,'Col No.   Skew       Axial      Mom Perp      Mom
Par'/
     *       14X,'--------------------------------------------------
-')
 100  FORMAT(14X,I4,6X,F4.0,3F12.2)
      RETURN
      END
*
*.............................FRAME
ANALYSIS........................
*
      SUBROUTINE FRAME
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      DIMENSION  GF(240),JRL(240),GK(240,9),DL(6),NODE(80),
     * GK1(240,9),AMAT(6,80,5),EK(6,6),TLR(6,6),TRL(6,6),AXL(6,6)
     * ,FIXED(6,80)
*
      CALL  INPUT2( JRL,NODE)
      CALL  GSTIFF( GK )
      CALL  MODSTIF( GK,JRL,NACT,MBAND )
*
      DO 10 ILOAD = 1,5
      DO 22 IL=1,NACT
      DO 22 IM=1,MBAND
  22  GK1(IL,IM)=GK(IL,IM)
      DO 8 J=1,NACT
   8  GF(J)=0.0
      DO 1 I=1,NMEM
```

```
      DO 1 J=1,6
    1 FIXED(J,I) = 0.0

      CALL  FORCE(ILOAD,GF,FIXED,NODE)

C.........MODIFY FORCE VECTOR FOR PRESCRIBED DISPL..........
      DO 9 IA=1,NACT
      IF(JRL(IA).EQ.0) GO TO 9
      GF(IA)=0.0
    9 CONTINUE

      CALL  SOLVER( GK1,GF,NACT,MBAND )
C
      DO 25 IM= 1,NOCOL
      ENRTA = ENRCOL
      AREA  = ARCOL
      CALL  MEMBER(EMOD,ENRTA,AREA,ALENTH(IM),EK )
      CALL  TRANS(  DC(IM,1),DC(IM,2),TLR,TRL )
      CALL  MULT  (  TLR,EK,AXL,6,6,6 )
      CALL  MULT  (  AXL,TRL,EK,6,6,6 )
      DO 25 ID=1,6
      SUM = 0.0
      DO 26 K =1,6
   26 SUM = SUM + EK(ID,K)*GF(NDISP(IM,K))
   25 AMAT(ID,IM,ILOAD)  = SUM+FIXED(ID,IM)
   10 CONTINUE
C.........COMPUTE FORCES IN THE LOCAL SYSTEM.............
      DO 29 IM=1,NOCOL
      CALL  TRANS( DC(IM,1),DC(IM,2),TLR,TRL )
      DO 29 ILOAD = 1,5
      DO 33 ID=1,6
      SUM=0.0
      DO 34 K=1,6
   34 SUM=SUM+TRL(ID,K)*AMAT(K,IM,ILOAD)
   33 DL(ID)  = SUM
      DO 29 ID=1,6
   29 AMAT(ID,IM,ILOAD)  = DL(ID)
*
      DO 93 IL=1,5
      DO 93 IC=1,NOCOL
      ALOAD(1,IC,IL)  = -AMAT(4,IC,IL)
      COEFF=1.
      IF(IL.EQ.1) COEFF=-1.
      IF(IL.EQ.5) COEFF=-1.
 93   ALOAD(2,IC,IL)  =   COEFF*AMAT(6,IC,IL)/12.
      RETURN
      END
*
*......................FRAME ANLAYSIS FOR BACK BEAMS...............
*
      SUBROUTINE FRAMEA

      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
```

```
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      DIMENSION  GF(240),JRL(240),GK(240,9),DL(6),NODE(80),
     * GK1(240,9),AMAT(6,80,5),EK(6,6),TLR(6,6),TRL(6,6),AXL(6,6)
     * ,FIXED(6,80)
*
      CALL  INPUT2( JRL,NODE)
      CALL  GSTIFF( GK )
      CALL  MODSTIF( GK,JRL,NACT,MBAND )
      ILOAD = 1
      DO 22 IL=1,NACT
      DO 22 IM=1,MBAND
   22 GK1(IL,IM)=GK(IL,IM)
      DO 8 J=1,NACT
    8 GF(J)=0.0
      DO 1 I=1,NMEM
      DO 1 J=1,6
    1 FIXED(J,I) = 0.0
C
      CALL  FORCEA(ILOAD,GF,FIXED,NODE)

C.........MODIFY FORCE VECTOR FOR PRESCRIBED DISPL..........
      DO 9 IA=1,NACT
      IF(JRL(IA).EQ.0) GO TO 9
      GF(IA)=0.0
    9 CONTINUE
C
      CALL  SOLVER( GK1,GF,NACT,MBAND )
C
      DO 25 IM= 1,NOCOL
      ENRTA = ENRCOL
      AREA  = ARCOL
      CALL  MEMBER(EMOD,ENRTA,AREA,ALENTH(IM),EK )
      CALL  TRANS(  DC(IM,1),DC(IM,2),TLR,TRL )
      CALL  MULT (  TLR,EK,AXL,6,6,6 )
      CALL  MULT (  AXL,TRL,EK,6,6,6 )
      DO 25 ID=1,6
      SUM = 0.0
      DO 26 K =1,6
   26 SUM = SUM + EK(ID,K)*GF(NDISP(IM,K))
   25 AMAT(ID,IM,ILOAD) = SUM+FIXED(ID,IM)
C.........COMPUTE FORCES IN THE LOCAL SYSTEM.............
      DO 29 IM=1,NOCOL
      CALL  TRANS( DC(IM,1),DC(IM,2),TLR,TRL )
```

```
      DO 29 ILOAD = 1,1
      DO 33 ID=1,6
      SUM=0.0
      DO 34 K=1,6
   34 SUM=SUM+TRL(ID,K)*AMAT(K,IM,ILOAD)
   33 DL(ID) = SUM
      DO 29 ID=1,6
   29 AMAT(ID,IM,ILOAD) = DL(ID)
*
      IL = 6
      ILL= 1
      DO 93 IC=1,NOCOL
      ALOAD(1,IC,IL) = -AMAT(4,IC,ILL)
      COEFF=-1.
   93 ALOAD(2,IC,IL) =  COEFF*AMAT(6,IC,ILL)/12.
      RETURN
      END
C
C.........................NODE AND MEMBER GENERATOR.................
C
      SUBROUTINE INPUT2(JRL,NODE)
      DIMENSION  JRL(240),NODE(80)
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      XLB = (CAPL-(NOBM-1)*BMSP)/2.
      IF(XLB.LT.0.0) XLB=0.0
      XLC = (CAPL-(NOCOL-1)*COLSP)/2.
      Y2  = DFX + COLH + CAPD/2.
      XG(1) = 0.
      YG(1) = Y2
      KC = 1
      I  = 1
      XT = XLB
      IF(XT.LE.0.0001) GOTO 9
15    IF(XT.EQ.XLC) GOTO 7
      XX=ABS(XT-XLC)
      IF(XX.LE.0.01) GOTO 7
C      IF(XT-XLC) 5,7,6
      IF(XT.LT.XLC) GOTO 5
      IF(XT.GT.XLC) GOTO 6
 5    I = I+1
      XG(I) = XT
      YG(I) = Y2
```

```
      XX=ABS(XT-CAPL)
      IF(XX.LE.0.01) GO TO 44
      GOTO 9
 6    IF(XLC.LE.0.0001) GOTO 12
      I=I+1
      XG(I) = XLC
      YG(I) = Y2
12    I=I+1
      XG(I) = XLC
      YG(I) = 0.0
      XLC = XLC + COLSP
      KC  = KC+1
      IF(KC.GT.NOCOL) XLC=1000000.
      GOTO 15
 7    IF(XLC.LE.0.0001) GOTO 19
      I=I+1
      XG(I) = XT
      YG(I) = Y2
19    I=I+1
      XG(I) = XLC
      YG(I) = 0.0
      XX=ABS(XLC-CAPL)
      IF(XX.LE.0.01) GO TO 44
      XLC = XLC + COLSP
      KC  = KC+1
      IF(KC.GT.NOCOL) XLC=1000000.
 9    XT = XT + BMSP
      XX=ABS(XT-CAPL)
      IF(XX.LE.0.01) GOTO 15
      IF(XT.GT.CAPL) THEN
         XT = CAPL
C          I=I+1
C          XG(I) = CAPL
C          YG(I) = Y2
      ENDIF
      GOTO 15
44    NJOIN = I
      NMEM = 1
      DO 30 I=1,NJOIN-1
         IF(YG(I+1).EQ.0.0)THEN
            IEND(NMEM,1) = I
            IEND(NMEM,2) = I+1
            NMEM = NMEM+1
         ENDIF
30    CONTINUE
*
      I=1
35    J=1
      IF(I.LE.(NJOIN-1))THEN
         K=I+1
         XX=ABS(XG(I)-XG(I+1))
         IF(XX.LE.0.01.AND.K.EQ.NJOIN) GO TO 39
         IF(XX.LE.0.01) J=2
```

```
 30   FORMAT(10X,'STREAM FLOW (GROUP I)'//
     *        14X,'P      =',F7.3,5X,'SF Mom Perp =',F9.3//)
 40   FORMAT(/12X,'***************** Bd = ',F4.2,'
****************'//
     *        14X,'Pu   =',F9.3,5X,'Mom Mag     =',F9.3//
     *        14X,'Sum Mom Perp =',F9.3/
     *        14X,'Sum Mom Par  =',F9.3//)
 50   FORMAT(14X,'MINIMUM ECCENTRICITY MOMENTS'//
     *        18X,'Mom Perp =',F9.3/
     *        18X,'Mom Par  =',F9.3//)
 60   FORMAT(14X,'MINIMUM ECCENTRICITY MOMENT'//
     *        18X,'Min Mom  =',F9.3/
     *        18X,'Resultant=',F9.3//)
 70   FORMAT(14X,'CONTROLLING MOMENTS  (GROUP I)'//
     *        18X,'Mom Perp =',F9.3/
     *        18X,'Mom Par  =',F9.3/)
      RETURN
      END
*
*........................GROUP II (APPROXIMATE).................
*
      SUBROUTINE GROUP2
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
      COMMON/TEN/ SPANF,SPANB
*
*     GROUP II WIND LOADING, 3.15.2.1.1
      WSKEW(1) =  0.
      WLAT(1)  =  0.050
      WLON(1)  =  0.
      WSKEW(2) = 15.
      WLAT(2)  =  0.044
      WLON(2)  =  0.006
      WSKEW(3) = 30.
      WLAT(3)  =  0.041
      WLON(3)  =  0.012
      WSKEW(4) = 45.
      WLAT(4)  =  0.033
      WLON(4)  =  0.016
      WSKEW(5) = 60.
      WLAT(5)  =  0.017
      WLON(5)  =  0.019
*
*     WIND ON SUPERSTRUCTURE
      WAREA=(RALHT+SLBD+HNCH+PAD+PDSTL)*SPAN
      WAREA=WAREA + BMDF*(SPANF/2.)
```

```
      WAREA=(WAREA + BMDB*(SPANB/2.))*CMULT
      H1   =(RALHT+SLBD+HNCH+BMD+PAD+PDSTL)/2.+CAPD+(COLH+FIXD)/2.
      H2   =H1+(COLH+FIXD)/2.
      WRITE(5,10) WAREA,H1,H2
      WRITE(5,20)
      DO 1 I=1,5
      RMOM1(I)=(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))*WAREA*H1/NOCOL
      RMOM2(I)=(WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))*WAREA*H2/NOCOL
      WRITE(5,30) WSKEW(I),RMOM1(I),RMOM2(I)
 1    CONTINUE
*
*     WIND ON CAP
      H1 = (CAPD/2.)+(COLH+FIXD)/2.
      H2 = H1+(COLH+FIXD)/2.
      WRITE(5,40) CAPD*CAPW,CAPD*CAPL,H1,H2
      WRITE(5,20)
      DO 2 I=1,5
         THETA    = PHI+WSKEW(I)/57.29578
         TEMP1    = COS(THETA)*0.04*CAPD*CAPW*H1/NOCOL
         TEMP2    = SIN(THETA)*0.04*CAPD*CAPL*H2/NOCOL
         RMOM1(I) = RMOM1(I)+TEMP1
         RMOM2(I) = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
 2    CONTINUE
*
*     WIND ON COLUMN
      H1 = COLH/2.
      H2 = COLH/2.+FIXD
      IF(R.EQ.0)THEN
         COLA1 = COLH*D1
         COLA2 = COLH*D2
         IF(IAXIS.EQ.2) COLA2 = ((NOCOL-1)*COLSP+D2)*COLH/NOCOL
      ELSE
         COLA1 = COLH*2.*R
         COLA2 = COLH*2.*R
         IF(IAXIS.EQ.2) COLA2 = ((NOCOL-1)*COLSP+2.*R)*COLH/NOCOL
      ENDIF
      WRITE(5,50) COLA1,COLA2,H1,H2
      WRITE(5,20)
*
*     TEST WATER PRESSURE AGAINST WIND (40 PSF)
      DO 3 I=1,5
         THETA    = PHI+WSKEW(I)/57.29578
         TEMP1    = COS(THETA)*COLA1*0.04*H1
         TEMP2    = SIN(THETA)*COLA2*0.04*H2
         RMOM1(I) = RMOM1(I)+TEMP1
         RMOM2(I) = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
         SFWD2(I) = 0.
         SFWD3(I) = 0.
         IF(SFMOM1.GT.TEMP1)     SFWD2(I) = SFMOM1-TEMP1
         IF(SFMOM1.GT.TEMP1*0.3) SFWD3(I) = SFMOM1-TEMP1*0.3
 3    CONTINUE
```

```
*
*     WIND REDUCTION FACTOR
      WRF = DSWIND*DSWIND/10000.
      WRITE(5,60) WRF
*
*     OUTPUT SUM WIND MOMENTS
      DO 4 I=1,5
      WRITE(5,70) WSKEW(I),RMOM1(I),RMOM2(I),RMOM1(I)*WRF,
     *            RMOM2(I)*WRF
 4    CONTINUE
*
*     OUTPUT ADDITIONAL STREAM FLOW MOMENT PERP
      WRITE(5,80)
      WRITE(5,20)
      DO 5 I=1,5
      WRITE(5,30) WSKEW(I),SFWD2(I)
 5    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD  = 1.0
      K   = 2
      DO 6 J=1,2
      IF(J.EQ.2)THEN
         BD  = 0.75
         K   = 7
      ENDIF
      PU    = 1.3*BD*DLSUM
      DELTA = CM/(1.-PU/(.7*PC))
      WRITE(5,90) BD,PU,DELTA
      WRITE(5,20)
      DO 7 I=1,5
         RM1U = 1.3*DELTA*(RMOM1(I)*WRF+SFWD2(I))
         RM2U = 1.3*DELTA*RMOM2(I)*WRF
         IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
         IF(IAXIS.EQ.1) RM2U = 0.
         IF(IAXIS.EQ.2) RM1U = 0.
         WRITE(5,30) WSKEW(I),RM1U,RM2U + (1.3*ECCDL)*DELTA
         PCA2(I+K,1) = PU+.49
         PCA2(I+K,2) = RM1U+.49
         PCA2(I+K,3) = RM2U+.49 + (1.3*ECCDL)*DELTA
 7    CONTINUE
 6    CONTINUE
 10   FORMAT(//4X,'GROUP II LOADING'///
     *       10X,'WIND ON SUPERSTRUCTURE (GROUP II)'//
     *       14X,'Wind Area Per Bent =',F8.2//
     *       14X,'H Perp             =',F8.2/
     *       14X,'H Par              =',F8.2//)
 20   FORMAT(14X,'Skew',6X,'Mom Perp',7X,'Mom Par'/
     *       14X,'--------------------------------')
 30   FORMAT(14X,F4.0,2F14.3)
 40   FORMAT(//10X,'WIND ON CAP (GROUP II)'//
     *       14X,'Cap Area Perp      =',F8.2/
     *       14X,'Cap Area Par       =',F8.2//
     *       14X,'H Perp             =',F8.2/
```

```
     *        14X,'H Par               =',F8.2/)
 50   FORMAT(//10X,'WIND ON COLUMN (GROUP II)'//
     *        14X,'Column Area Perp   =',F8.2/
     *        14X,'Column Area Par    =',F8.2//
     *        14X,'H Perp             =',F8.2/
     *        14X,'H Par              =',F8.2/)
 60   FORMAT(//10X,'WIND REDUCTION (GROUP II)'//
     *        14X,'Wind Red. Factor   =',F8.3//
     *        29X,'100 MPH Wind',16X,'Reduced Wind'//
     *        14X,'Skew',6X,'Mom Perp',7X,'Mom Par',
     *                   6X,'Mom Perp',7X,'Mom Par'/
     *        14X,'--------------------------------',
     *         6X,'----------------------')
 70   FORMAT(14X,F4.0,4F14.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW MOMENT PERP (GROUP
II)'/)
 90   FORMAT(//12X,'**************** Bd = ',F4.2,'
****************'//
     *        14X,'Pu   =',F9.3,5X,'Mom Mag     =',F9.3/)
      RETURN
      END
*
*.......................GROUP III (APPROXIMATE)..................
*
      SUBROUTINE GROUP3
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
*     GROUP III WIND LOADING, 3.15.2.1.2
      WLAT(1) = 0.100
      WLON(1) = 0.
      WLAT(2) = 0.088
      WLON(2) = 0.012
      WLAT(3) = 0.082
      WLON(3) = 0.024
      WLAT(4) = 0.066
      WLON(4) = 0.032
      WLAT(5) = 0.034
      WLON(5) = 0.038
*
*     0.3 WIND (USING SUM, REDUCED WIND FROM GROUP II)
      WRITE(5,10)
      WRITE(5,20)
      DO 1 I=1,5
         RMOM1(I) = RMOM1(I)*0.3
         RMOM2(I) = RMOM2(I)*0.3
```

```
         WRITE(5,30) WSKEW(I),RMOM1(I),RMOM2(I)
 1    CONTINUE
*
*     WIND ON LIVE LOAD (WL)
      H1 = (COLH+FIXD)/2.+CAPD+SLBD+HNCH+BMD+PAD+PDSTL+HDECK
      H2 = H1+(COLH+FIXD)/2.
      WRITE(5,40) H1,H2
      WRITE(5,20)
      DO 2 I=1,5
         TEMP1     = (WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *                *SPAN*H1/NOCOL
         TEMP2     = (WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))
     *                *SPAN*H2/NOCOL
         RMOM1(I) = RMOM1(I)+TEMP1
         RMOM2(I) = RMOM2(I)+TEMP2
         WRITE(5,30) WSKEW(I),TEMP1,TEMP2
 2    CONTINUE
*
*     LONGITUDINAL FORCE (LF)
      RM1LF = SIN(PHI)*0.05*RLLANE*H1
      RM2LF = COS(PHI)*0.05*RLLANE*H2
      WRITE(5,50) H1,RM1LF,H2,RM2LF
      DO 3 I=1,5
         RMOM1(I) = RMOM1(I)+RM1LF
         RMOM2(I) = RMOM2(I)+RM2LF
 3    CONTINUE
*
*     OUTPUT ADDITIONAL STREAM FLOW MOMENT PERP
      WRITE(5,60)
      WRITE(5,20)
      DO 4 I=1,5
         WRITE(5,30) WSKEW(I),SFWD3(I)
 4    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD  = 1.0
      K   = 12
      DO 5 J=1,2
         IF(J.EQ.2)THEN
            BD  = 0.75
            K   = 17
         ENDIF
         PU    = 1.3*(BD*DLSUM+RLLI)
         DELTA = CM/(1.-PU/(.7*PC))
         WRITE(5,70) BD,PU,DELTA
         WRITE(5,20)
         DO 6 I=1,5
            RM1U = 1.3*DELTA*(RMOM1(I)+SFWD3(I)+CFMOM1)
            RM2U = 1.3*DELTA*(RMOM2(I)+CFMOM2)
            IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
            IF(IAXIS.EQ.1) RM2U = 0.
            IF(IAXIS.EQ.2) RM1U = 0.
            WRITE(5,30) WSKEW(I),RM1U,RM2U
     *                                  + (1.3*(ECCDL+ECCLL))*DELTA
```

```
            PCA2(I+K,1) = PU+.49
            PCA2(I+K,2) = RM1U+.49
            PCA2(I+K,3) = RM2U+.49 + (1.3*(ECCDL+ECCLL))*DELTA
 6        CONTINUE
 5      CONTINUE
10      FORMAT(///4X,'GROUP III LOADING'//
     *         10X,'0.3 WIND (GROUP III)'/)
20      FORMAT(14X,'Skew',6X,'Mom Perp',7X,'Mom Par'/
     *         14X,'-------------------------------')
30      FORMAT(14X,F4.0,2F14.3)
40      FORMAT(//10X,'WIND ON LIVE LOAD (GROUP III)'//
     *         14X,'H Perp              =',F8.2/
     *         14X,'H Par               =',F8.2/)
50      FORMAT(//10X,'LONGITUDINAL FORCE (GROUP III)'//
     *         14X,'H Perp =',F7.2,5X,'LF Mom Perp =',F9.3/
     *         14X,'H Par  =',F7.2,5X,'LF Mom Par  =',F9.3)
60      FORMAT(//10X,'ADDITIONAL STREAM FLOW MOMENT PERP (GROUP
III)'/)
70      FORMAT(//12X,'**************** Bd = ',F4.2,'
****************'//
     *         14X,'Pu   =',F9.3,5X,'Mom Mag     =',F9.3/)
        RETURN
        END
*
*.........................GROUP I (FRAME ANALYSIS).................
*
        SUBROUTINE FRAME1
        COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
        COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
        COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
        COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
        COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
        COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
        COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
        COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
        COMMON/TEN/ SPANF,SPANB
*
        COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
        COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *              FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
        COMMON/FR3/ TFORCE(3,20)
*
*       COMPUTE DEAD AND LIVE LOAD PER BEAM
        DL1    = (WRAIL*(SPANF/2.0)*NORAL + SLBD*SLBW*
     *              (SPANF/2.0)*.15)/NOBMF + WBMF*(SPANF/2.0)
        DL2    = (WRAIL*(SPANB/2.0)*NORAL + SLBD*SLBW*
     *              (SPANB/2.0)*.15)/NOBMB + WBMB*(SPANB/2.0)
*
        DLCOL = WCOL*(COLH+FIXD)
        DL5   = CAPW*CAPD*.0125
        FRLLI = RLLI*NOCOL
        FTRK  = RLLTRK*NOCOL
        FLANE = RLLANE*NOCOL
```

```
*
      NOBM = NOBMF
      IF(NOBMF.LT.NOBMB) NOBM = NOBMB
*
*     CENTRIFUGAL FORCE
      C      = .00117*(DSMPH**2)*CRV
      CF     = C/100.*FTRK
      H1     = (COLH+FIXD)+CAPD+PDSTL+PAD+BMD+HNCH+SLBD+HDECK
      CFMOM  = COS(PHI)*CF*H1/Y2
      H2     = H1
      CFMOM2 = SIN(PHI)*CF/NOCOL*H2
*
*     STREAM FLOW
      IF(R.EQ.0)THEN
         P       = (1.375)*V*V/1000.
         SFMOM   = P*D1/12.
      ELSE
         P       = (2./3.)*V*V/1000.
         SFMOM   = P*(2.*R)/12.
      ENDIF
*
*     AXIAL LOAD AND MOMENTS
      WRITE(6,10)
      BD = 1.0
      DO 1 I=1,2
      IF(I.EQ.2) BD = 0.75
      WRITE(6,20) BD
      PMAX = 0.0
      DO 2 J=1,NOCOL
      FORCE(1,J)=1.3*(BD*(DL1*ALOAD(1,J,1)+DL2*ALOAD(1,J,6)+
     *              DL5*ALOAD(1,J,5)+DLCOL)+
     *             1.67*FRLLI/NOBM*ALOAD(1,J,1)+CFMOM*ALOAD(1,J,2)+
     *              SFMOM*ALOAD(1,J,4))
 2    IF(FORCE(1,J).GE.PMAX) PMAX = FORCE(1,J)
      DELTA = CM/(1.-PMAX/(.7*PC))
      DO 3 J=1,NOCOL
      FORCE(2,J)=-1.3*DELTA*(DL1*ALOAD(2,J,1)+DL2*ALOAD(2,J,6)
     *          +DL5*ALOAD(2,J,5)
     *          +1.67*FRLLI/NOBM*ALOAD(2,J,1)+CFMOM*ALOAD(2,J,2)
     *          +SFMOM*ALOAD(2,J,4))
 3    CONTINUE
      RM2U   = 1.3*DELTA*(CFMOM2)
*
*     COMPARE COMPUTED MOMENTS AGAINST MINIMUM ECCENTRICITY
      DO 4 J=1,NOCOL
      IF(R.EQ.0)THEN
         RMIN1    = DELTA*0.1*D2*FORCE(1,J)
         RMIN2    = DELTA*0.1*D1*FORCE(1,J)
         IF(RMIN1.GT.FORCE(2,J)) FORCE(2,J)  = RMIN1
         IF(RMIN2.GT.RM2U) RM2U = RMIN2
      ELSE
         IF(IAXIS.EQ.1) RM2U = 0.
         RMIN     = DELTA*0.1*(2.*R)*FORCE(1,J)
```

```
            RESMU    = SQRT(FORCE(2,J)*FORCE(2,J)+RM2U*RM2U)
            IF(RMIN.GT.RESMU)THEN
               FORCE(2,J) = RMIN/SQRT(2.0)
               RM2U = FORCE(2,J)
               IF(IAXIS.EQ.1) FORCE(2,J) = RMIN
               IF(IAXIS.EQ.2) RM2U = RMIN
            ENDIF
         ENDIF
*
*        OUTPUT GROUP I LOADING
         IF(RM2DFL.GT.RM2U) RM2U = RM2DFL
         IF(IAXIS.EQ.1) RM2U = 0.0
 4       WRITE(6,30) J,FORCE(1,J),FORCE(2,J),
     *                 RM2U + (1.3*ECCDL + 2.17*ECCLL)*DELTA
*
         KK  = 1
         JJJ = 1
         IF(I.EQ.2) KK=NOCOL+1
         DO 5 JJ=KK,NOCOL*I
         TFORCE(1,JJ) = FORCE(1,JJJ)
         TFORCE(2,JJ) = FORCE(2,JJJ)
         JJJ=JJJ+1
 5       FORCE(3,JJ) = RM2U + (1.3*ECCDL + 2.17*ECCLL)*DELTA
*
 1       CONTINUE
*
         DO 6 I=1,2*NOCOL
         FORCE(1,I) = TFORCE(1,I)
  6      FORCE(2,I) = TFORCE(2,I)
*
 10      FORMAT(4X,'GROUP I LOADING')
 20      FORMAT(//12X,'***************** Bd = ',F4.2,'
***************'///
     *         14X,'COL NO.    AXIAL      MOM PERP     MOM PAR'/
     *         14X,'-----------------------------------------')
 30      FORMAT(14X,I4,3F12.2)
         RETURN
         END
*
*......................GROUP II (FRAME ANALYSIS)..................
*
         SUBROUTINE FRAME2
         COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
         COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
         COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
         COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
         COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
         COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
         COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
         COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
         COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
         COMMON/TEN/ SPANF,SPANB
*
```

```
      COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
      COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *            FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
*
*
*     WIND ON SUPERSTRUCTURE
      WRITE(6,11)
      WRITE(6,20)
      WAREA=(RALHT+SLBD+HNCH+PAD+PDSTL)*SPAN
      WAREA=WAREA + BMDF*(SPANF/2.)
      WAREA=(WAREA + BMDB*(SPANB/2.))*CMULT
      H1   =(RALHT+SLBD+HNCH+BMD+PAD+PDSTL)/2.+CAPD+(COLH+FIXD)
      DO 1 J=1,NOCOL
      WRITE(6,*)
      DO 1 I=1,5
      FRMOM1(I,J) = -(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *               *WAREA*H1/Y2*ALOAD(2,J,2)
      AXIAL(I,J)  =  (WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *               *WAREA*H1/Y2*ALOAD(1,J,2)
      WRITE(6,30) J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J)
 1    CONTINUE
*
*     WIND ON CAP
      WRITE(6,40)
      WRITE(6,20)
      H1 = (CAPD/2.)+(COLH+FIXD)
      DO 2 J=1,NOCOL
      WRITE(6,*)
      DO 2 I=1,5
         THETA      = PHI+WSKEW(I)/57.29578
         TEMP1      = -COS(THETA)*0.04*CAPD*CAPW*H1/Y2*ALOAD(2,J,2)
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
         TEMP2      =  COS(THETA)*0.04*CAPD*CAPW*H1/Y2*ALOAD(1,J,2)
         AXIAL(I,J)  = AXIAL(I,J)+TEMP2
         WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
 2    CONTINUE
*
*     WIND ON COLUMN
      WRITE(6,50)
      WRITE(6,20)
      IF(R.EQ.0)THEN
         COLA1 = D1/12.
      ELSE
         COLA1 = 2.*R/12.
      ENDIF
*
*     TEST WATER PRESSURE AGAINST WIND (40 PSF)
      DO 3 J=1,NOCOL
      WRITE(6,*)
      DO 3 I=1,5
         THETA      = PHI+WSKEW(I)/57.29578
         TEMP1      = -COS(THETA)*COLA1*.04*ALOAD(2,J,3)
         TEMP2      =  COS(THETA)*COLA1*.04*ALOAD(1,J,3)
```

```
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
         AXIAL(I,J)  = AXIAL(I,J)+TEMP2
         WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
         FSFWD2(I,J) = 0.
         FSFAX2(I,J) = 0.
         FSFWD3(I,J) = 0.
         FSFAX3(I,J) = 0.
         FSF    = -SFMOM*ALOAD(2,J,4)
         FSFAX =   SFMOM*ALOAD(1,J,4)
         IF(FSF.GT.TEMP1)THEN
            FSFWD2(I,J) = FSF-TEMP1
            FSFAX2(I,J) = FSFAX-TEMP2
         ENDIF
         IF(FSF.GT.TEMP1*0.3)THEN
            FSFWD3(I,J) = FSF-TEMP1*0.3
            FSFAX3(I,J) = FSFAX-TEMP2*0.3
         ENDIF
 3    CONTINUE
*
*     WIND REDUCTION FACTOR
      WRF = DSWIND*DSWIND/10000.
*
*     OUTPUT SUM WIND MOMENTS
      WRITE(6,60)
      DO 4 J=1,NOCOL
      WRITE(6,*)
      DO 4 I=1,5
      WRITE(6,70)J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J),AXIAL(I,J)*WRF,
     *                                              FRMOM1(I,J)*WRF
 4    CONTINUE
*
*     OUTPUT ADDITIONAL STREAM FLOW FORCES
      WRITE(6,80)
      WRITE(6,20)
      DO 5 J=1,NOCOL
      WRITE(6,*)
      DO 5 I=1,5
      WRITE(6,30) J,WSKEW(I),FSFAX2(I,J),FSFWD2(I,J)
 5    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD   = 1.0
      K    = 2*NOCOL
      N    = K
      DO 6 J=1,2
      IF(J.EQ.2)THEN
         BD   = 0.75
         K    = 7*NOCOL
         N    = K
      ENDIF
      WRITE(6,*)
      WRITE(6,*)
      WRITE(6,90) BD
      PMAX  = 0.0
```

```
      DO 8 L=1,NOCOL
      DO 9 M=1,5
      FORCE(1,M+K) = 1.3*(BD*(DL1*ALOAD(1,L,1)+DL2*ALOAD(1,L,6)
     *               +DL5*ALOAD(1,L,5)
     *               +DLCOL)+AXIAL(M,L)+FSFAX2(M,L))
 9    IF(FORCE(1,M+K).GE.PMAX) PMAX = FORCE(1,M+K)
 8    K=K+5
      DELTA = CM/(1.-PMAX/(.7*PC))
      IF(J.EQ.1)THEN
         DELMY = DELTA/(CM/(1.-PCA2(3,1)/(.7*PC)))
      ELSE
         DELMY = DELTA/(CM/(1.-PCA2(8,1)/(.7*PC)))
      ENDIF
      DO 7 L=1,NOCOL
      WRITE(6,*)
      IF(J.EQ.1) LOC=3
      IF(J.EQ.2) LOC=8
      DO 10 M=1,5
         FORCE(2,M+N) = 1.3*DELTA*(-DL1*ALOAD(2,L,1)
     *                  -DL2*ALOAD(2,L,6)-DL5*ALOAD(2,L,5)
     *                  +FRMOM1(M,L)*WRF+FSFWD2(M,L))
         FORCE(3,M+N) = DELMY*PCA2(LOC,3)
         LOC=LOC+1
 10      WRITE(6,100) L,WSKEW(M),FORCE(1,M+N),FORCE(2,M+N),
     *                                   FORCE(3,M+N)
 7    N=N+5
 6    CONTINUE
 11   FORMAT(///4X,'GROUP II LOADING'///
     *       10X,'WIND ON SUPERSTRUCTURE (GROUP II)'/)
 20   FORMAT(14X,'Col No.',6X,'Skew',6X,' Axial ',5X,'Mom Perp'/
     *       14X,'------------------------------------------')
 30   FORMAT(14X,I4,9X,F4.0,F12.3,F13.3)
 40   FORMAT(//10X,'WIND ON CAP (GROUP II)'/)
 50   FORMAT(//10X,'WIND ON COLUMN (GROUP II)'/)
 60   FORMAT(//10X,'WIND REDUCTION (GROUP II)'//
     *       40X,'100 MPH Wind',16X,'Reduced Wind'//
     *       14X,'Col No.',4X,'Skew',7X,' Axial ',5X,'Mom Perp',
     *                  7X,' Axial ',5X,'Mom Perp'/
     *       14X,'--------------------------------------------',
     *        8X,'-------------------')
 70   FORMAT(14X,I4,7X,F4.0,2F13.3,1X,2F13.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW FORCES (GROUP II)'/)
 90   FORMAT(///12X,'******************** Bd = ',F4.2,'
************
     **********'///
     *      14X,'Col No.   Skew       Axial     Mom Perp     Mom
Par'/
     *      14X,'-------------------------------------------------
-')
 100  FORMAT(14X,I4,6X,F4.0,3F12.2)
      RETURN
      END
*
```

```
*........................GROUP III (FRAME ANALYSIS)................
*
      SUBROUTINE FRAME3
      COMMON/ONE/ SPAN,PHI,NOLAN,WRAIL,NORAL,RALHT,SLBD,SLBW
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,FIXD,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WTRD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/FIV/ RF,DLSUM,RLLANE,RLLTRK,RLLI,PC,SFMOM1,WRF
      COMMON/SIX/ WSKEW(5),WLAT(5),WLON(5),RMOM1(5),RMOM2(5)
      COMMON/SEV/ DSWIND,HDECK,CMULT,SFWD2(5),SFWD3(5),IAXIS
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      COMMON/FR1/ FORCE(3,220),DL1,DL2,DLCOL,DL5,FRLLI,FTRK,SFMOM
      COMMON/FR2/ FRMOM1(5,80),AXIAL(5,80),FSFWD2(5,80),FLANE,
     *            FSFWD3(5,80),FSFAX2(5,80),FSFAX3(5,80),CFMOM
*
*     0.3 WIND (USING SUM WIND FROM FRAME GROUP II)
      WRITE(6,11)
      WRITE(6,20)
      DO 1 J=1,NOCOL
      WRITE(6,*)
      DO 1 I=1,5
         FRMOM1(I,J) = FRMOM1(I,J)*0.3
         AXIAL(I,J)  = AXIAL(I,J)*0.3
         WRITE(6,30) J,WSKEW(I),AXIAL(I,J),FRMOM1(I,J)
 1    CONTINUE
*
*     WIND ON LIVE LOAD (WL)
      WRITE(6,40)
      WRITE(6,20)
      H1 = (COLH+FIXD)+CAPD+SLBD+HNCH+BMD+PAD+PDSTL+HDECK
      DO 2 J=1,NOCOL
      WRITE(6,*)
      DO 2 I=1,5
         TEMP1      =-(WLAT(I)*COS(PHI)-WLON(I)*SIN(PHI))
     *                *SPAN*H1/Y2*ALOAD(2,J,2)
         TEMP2      = (WLAT(I)*SIN(PHI)+WLON(I)*COS(PHI))
     *                *SPAN*H1/Y2*ALOAD(1,J,2)
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
         AXIAL(I,J)  = AXIAL(I,J)+TEMP2
         WRITE(6,30) J,WSKEW(I),TEMP2,TEMP1
 2    CONTINUE
*
*     LONGITUDINAL FORCE (LF)
      WRITE(6,50)
      WRITE(6,60)
      DO 3 J=1,NOCOL
      TEMP1 =-SIN(PHI)*0.05*FLANE*H1/Y2*ALOAD(2,J,2)
      TEMP2 = SIN(PHI)*0.05*FLANE*H1/Y2*ALOAD(1,J,2)
      WRITE(6,65) J,TEMP2,TEMP1
      DO 3 I=1,5
         FRMOM1(I,J) = FRMOM1(I,J)+TEMP1
```

```
      AXIAL(I,J)   = AXIAL(1,J)+TEMP2
 3    CONTINUE
*
*
*     OUTPUT ADDITIONAL STREAM FLOW FORCES
      WRITE(6,80)
      WRITE(6,20)
      DO 5 J=1,NOCOL
      WRITE(6,*)
      DO 5 I=1,5
      WRITE(6,30) J,WSKEW(I),FSFAX3(I,J),FSFWD3(I,J)
 5    CONTINUE
*     AXIAL LOAD AND MOMENT MAGNIFIER
      BD   = 1.0
      K    = 12*NOCOL
      N    = K
      DO 6 J=1,2
      IF(J.EQ.2)THEN
         BD   = 0.75
         K    = 17*NOCOL
         N    = K
      ENDIF
      WRITE(6,90) BD
      PMAX  = 0.0
      DO 8 L=1,NOCOL
      DO 9 M=1,5
      FORCE(1,M+K)  = 1.3*(BD*(DL1*ALOAD(1,L,1)+DL2*ALOAD(1,L,6)
     *         +DL5*ALOAD(1,L,5)
     *         +DLCOL)+AXIAL(M,L)+FSFAX3(M,L)
     *         +FRLLI/NOBM*ALOAD(1,L,1)+CFMOM*ALOAD(1,L,2))
 9    IF(FORCE(1,M+K).GE.PMAX) PMAX = FORCE(1,M+K)
 8    K=K+5
      DELTA = CM/(1.-PMAX/(.7*PC))
      IF(J.EQ.1)THEN
         DELMY = DELTA/(CM/(1.-PCA2(13,1)/(.7*PC)))
      ELSE
         DELMY = DELTA/(CM/(1.-PCA2(18,1)/(.7*PC)))
      ENDIF
      DO 7 L=1,NOCOL
      WRITE(6,*)
      IF(J.EQ.1) LOC=13
      IF(J.EQ.2) LOC=18
      DO 10 M=1,5
      FORCE(2,M+N)  = 1.3*DELTA*(-DL1*ALOAD(2,L,1)-DL2*ALOAD(2,L,6)
     *                -DL5*ALOAD(2,L,5)
     *                -FRLLI/NOBM*ALOAD(2,L,1)-CFMOM*ALOAD(2,L,2)
     *                +FSFWD3(M,L)+FRMOM1(M,L))
      FORCE(3,M+N)  = DELMY*PCA2(LOC,3)
      LOC=LOC+1
10    WRITE(6,100) L,WSKEW(M),FORCE(1,M+N),FORCE(2,M+N),
     *                                     FORCE(3,M+N)
 7    N=N+5
 6    CONTINUE
```

```
 11   FORMAT(///4X,'GROUP III LOADING'///
     *         10X,'0.3 WIND (GROUP III)'/)
 20   FORMAT(14X,'Col No.',6X,'Skew',6X,' Axial ',5X,'Mom Perp'/
     *         14X,'------------------------------------------')
 30   FORMAT(14X,I4,9X,F4.0,F12.3,F13.3)
 40   FORMAT(//10X,'WIND ON LIVE LOAD (GROUP III)'/)
 50   FORMAT(//10X,'LONGITUDINAL FORCE (GROUP III)'/)
 60   FORMAT(14X,'Col No.',6X,'Axial',5X,'Mom Perp'/
     *         14X,'----------------------------')
 65   FORMAT(/14X,I4,2X,2F12.3)
 80   FORMAT(//10X,'ADDITIONAL STREAM FLOW FORCES (GROUP III)'/)
 90   FORMAT(///12X,'********************** Bd = ',F4.2,'
************
     **********'///
     *       14X,'Col No.   Skew       Axial     Mom Perp      Mom
Par'/
     *       14X,'-----------------------------------------------------
-')
 100  FORMAT(14X,I4,6X,F4.0,3F12.2)
      RETURN
      END
*
*.............................FRAME
ANALYSIS.........................
*
      SUBROUTINE FRAME
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMT,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      DIMENSION  GF(240),JRL(240),GK(240,9),DL(6),NODE(80),
     * GK1(240,9),AMAT(6,80,5),EK(6,6),TLR(6,6),TRL(6,6),AXL(6,6)
     * ,FIXED(6,80)
*
      CALL  INPUT2( JRL,NODE)
      CALL  GSTIFF( GK )
      CALL  MODSTIF( GK,JRL,NACT,MBAND )
*
      DO 10 ILOAD = 1,5
      DO 22 IL=1,NACT
      DO 22 IM=1,MBAND
   22 GK1(IL,IM)=GK(IL,IM)
      DO 8 J=1,NACT
    8 GF(J)=0.0
      DO 1 I=1,NMEM
```

```
      DO 1 J=1,6
    1 FIXED(J,I) = 0.0

      CALL  FORCE(ILOAD,GF,FIXED,NODE)

C.........MODIFY FORCE VECTOR FOR PRESCRIBED DISPL..........
      DO 9 IA=1,NACT
      IF(JRL(IA).EQ.0) GO TO 9
      GF(IA)=0.0
    9 CONTINUE

      CALL  SOLVER( GK1,GF,NACT,MBAND )
C
      DO 25 IM= 1,NOCOL
      ENRTA = ENRCOL
      AREA  = ARCOL
      CALL  MEMBER(EMOD,ENRTA,AREA,ALENTH(IM),EK )
      CALL  TRANS(  DC(IM,1),DC(IM,2),TLR,TRL )
      CALL  MULT (  TLR,EK,AXL,6,6,6 )
      CALL  MULT (  AXL,TRL,EK,6,6,6 )
      DO 25 ID=1,6
      SUM = 0.0
      DO 26 K =1,6
   26 SUM = SUM + EK(ID,K)*GF(NDISP(IM,K))
   25 AMAT(ID,IM,ILOAD) = SUM+FIXED(ID,IM)
   10 CONTINUE
C.........COMPUTE FORCES IN THE LOCAL SYSTEM.............
      DO 29 IM=1,NOCOL
      CALL  TRANS( DC(IM,1),DC(IM,2),TLR,TRL )
      DO 29 ILOAD = 1,5
      DO 33 ID=1,6
      SUM=0.0
      DO 34 K=1,6
   34 SUM=SUM+TRL(ID,K)*AMAT(K,IM,ILOAD)
   33 DL(ID) = SUM
      DO 29 ID=1,6
   29 AMAT(ID,IM,ILOAD) = DL(ID)
*
      DO 93 IL=1,5
      DO 93 IC=1,NOCOL
      ALOAD(1,IC,IL) = -AMAT(4,IC,IL)
      COEFF=1.
      IF(IL.EQ.1) COEFF=-1.
      IF(IL.EQ.5) COEFF=-1.
  93  ALOAD(2,IC,IL) =  COEFF*AMAT(6,IC,IL)/12.
      RETURN
      END
*
*......................FRAME ANLAYSIS FOR BACK BEAMS...............
*
      SUBROUTINE FRAMEA

      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
```

```
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      DIMENSION  GF(240),JRL(240),GK(240,9),DL(6),NODE(80),
     * GK1(240,9),AMAT(6,80,5),EK(6,6),TLR(6,6),TRL(6,6),AXL(6,6)
     * ,FIXED(6,80)
*
      CALL  INPUT2( JRL,NODE)
      CALL  GSTIFF( GK )
      CALL  MODSTIF( GK,JRL,NACT,MBAND )
      ILOAD = 1
      DO 22 IL=1,NACT
      DO 22 IM=1,MBAND
   22 GK1(IL,IM)=GK(IL,IM)
      DO 8 J=1,NACT
    8 GF(J)=0.0
      DO 1 I=1,NMEM
      DO 1 J=1,6
    1 FIXED(J,I) = 0.0
C
      CALL  FORCEA(ILOAD,GF,FIXED,NODE)

C.........MODIFY FORCE VECTOR FOR PRESCRIBED DISPL..........
      DO 9 IA=1,NACT
      IF(JRL(IA).EQ.0) GO TO 9
      GF(IA)=0.0
    9 CONTINUE
C
      CALL  SOLVER( GK1,GF,NACT,MBAND )
C
      DO 25 IM= 1,NOCOL
      ENRTA = ENRCOL
      AREA  = ARCOL
      CALL  MEMBER(EMOD,ENRTA,AREA,ALENTH(IM),EK )
      CALL  TRANS(  DC(IM,1),DC(IM,2),TLR,TRL )
      CALL  MULT (  TLR,EK,AXL,6,6,6 )
      CALL  MULT (  AXL,TRL,EK,6,6,6 )
      DO 25 ID=1,6
      SUM = 0.0
      DO 26 K =1,6
   26 SUM = SUM + EK(ID,K)*GF(NDISP(IM,K))
   25 AMAT(ID,IM,ILOAD) = SUM+FIXED(ID,IM)
C.........COMPUTE FORCES IN THE LOCAL SYSTEM.............
      DO 29 IM=1,NOCOL
      CALL  TRANS( DC(IM,1),DC(IM,2),TLR,TRL )
```

```
      DO 29 ILOAD = 1,1
      DO 33 ID=1,6
      SUM=0.0
      DO 34 K=1,6
  34  SUM=SUM+TRL(ID,K)*AMAT(K,IM,ILOAD)
  33  DL(ID) = SUM
      DO 29 ID=1,6
  29  AMAT(ID,IM,ILOAD) = DL(ID)
*
      IL = 6
      ILL= 1
      DO 93 IC=1,NOCOL
      ALOAD(1,IC,IL) = -AMAT(4,IC,ILL)
      COEFF=-1.
 93   ALOAD(2,IC,IL) =  COEFF*AMAT(6,IC,ILL)/12.
      RETURN
      END
C
C.......................NODE AND MEMBER GENERATOR.................
C
      SUBROUTINE INPUT2(JRL,NODE)
      DIMENSION  JRL(240),NODE(80)
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE /
EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      XLB = (CAPL-(NOBM-1)*BMSP)/2.
      IF(XLB.LT.0.0) XLB=0.0
      XLC = (CAPL-(NOCOL-1)*COLSP)/2.
      Y2  = DFX + COLH + CAPD/2.
      XG(1) = 0.
      YG(1) = Y2
      KC = 1
      I  = 1
      XT = XLB
      IF(XT.LE.0.0001) GOTO 9
15    IF(XT.EQ.XLC) GOTO 7
      XX=ABS(XT-XLC)
      IF(XX.LE.0.01) GOTO 7
C      IF(XT-XLC) 5,7,6
      IF(XT.LT.XLC) GOTO 5
      IF(XT.GT.XLC) GOTO 6
 5    I = I+1
      XG(I) = XT
      YG(I) = Y2
```

```
      XX=ABS(XT-CAPL)
      IF(XX.LE.0.01) GO TO 44
      GOTO 9
 6    IF(XLC.LE.0.0001) GOTO 12
      I=I+1
      XG(I) = XLC
      YG(I) = Y2
12    I=I+1
      XG(I) = XLC
      YG(I) = 0.0
      XLC = XLC + COLSP
      KC  = KC+1
      IF(KC.GT.NOCOL) XLC=1000000.
      GOTO 15
 7    IF(XLC.LE.0.0001) GOTO 19
      I=I+1
      XG(I) = XT
      YG(I) = Y2
19    I=I+1
      XG(I) = XLC
      YG(I) = 0.0
      XX=ABS(XLC-CAPL)
      IF(XX.LE.0.01) GO TO 44
      XLC = XLC + COLSP
      KC  = KC+1
      IF(KC.GT.NOCOL) XLC=1000000.
 9    XT = XT + BMSP
      XX=ABS(XT-CAPL)
      IF(XX.LE.0.01) GOTO 15
      IF(XT.GT.CAPL) THEN
         XT = CAPL
C         I=I+1
C         XG(I) = CAPL
C         YG(I) = Y2
      ENDIF
      GOTO 15
44    NJOIN = I
      NMEM = 1
      DO 30 I=1,NJOIN-1
         IF(YG(I+1).EQ.0.0)THEN
            IEND(NMEM,1) = I
            IEND(NMEM,2) = I+1
            NMEM = NMEM+1
         ENDIF
30    CONTINUE
*
      I=1
35    J=1
      IF(I.LE.(NJOIN-1))THEN
         K=I+1
         XX=ABS(XG(I)-XG(I+1))
         IF(XX.LE.0.01.AND.K.EQ.NJOIN) GO TO 39
         IF(XX.LE.0.01) J=2
```

```
            IEND(NMEM,1) = I
            IEND(NMEM,2) = I+J
            NMEM = NMEM +1
 39         I=I+J
            GOTO 35
       ENDIF
*
       XT = XLB
       K  = 0
       DO 45 I=1,NJOIN
       XX=ABS(XT-XG(I))
       YY=ABS(YG(I)-Y2)
       IF(XX.GT.0.01) GO TO 45
       IF(YY.GT.0.01) GO TO 45
       K = K+1
       NODE(K) = I
       XT = XT+BMSP
 45    CONTINUE
       NMEM = NMEM-1
       NACT = 3*NJOIN
C
C      TEMPORARY PRINTS FOR NODE AND MEMBER GENERATOR
C
C        DO 209 JJJ=1,NJOIN
C 209    WRITE(9,219) JJJ,XG(JJJ),YG(JJJ)
C 219    FORMAT(1X,'NODE =',I5,8X,'X =',F5.1,8X,'Y =',F5.1)
C
C
C        DO 409 III=1,NMEM
C 409    WRITE(9,419) III,IEND(III,1),IEND(III,2)
C 419    FORMAT(1X,'MEM. NO.=',I5,7X,'I-END =',I5,7X,'J-END =',I5)
C        PAUSE
C
C
*
       DO 55 IN=1,NACT
  55   JRL(IN) = 0
       DO 54 I= 1,NJOIN
       IF(YG(I).NE.0.0) GO TO 54
       JRL(3*I-2) = 1
       JRL(3*I-1) = 1
       JRL(3*I  ) = 1
   54 CONTINUE
C  -----------------------------------------------------------------
       DO 13 I=1,NMEM
       XL= (XG(IEND(I,2))-XG(IEND(I,1)))*12.
       YL= (YG(IEND(I,2))-YG(IEND(I,1)))*12.
       ALENTH(I) = SQRT(XL*XL+YL*YL)
       DC(I,1) = XL/ALENTH(I)
       DC(I,2) = YL/ALENTH(I)
  13   CONTINUE
C      CALCULATE MAXIMUM SEMI-BANDWIDTH
       MBAND=0
```

```
      DO 31 I=1,NMEM
      ISUM= 3*IABS(IEND(I,2)-IEND(I,1)+1)
      IF(ISUM.LE.MBAND) GO TO 31
      MBAND=ISUM
   31 CONTINUE
      DO 32 IM=1,NMEM
      NDISP(IM,1) = 3*IEND(IM,1)-2
      NDISP(IM,2) = 3*IEND(IM,1)-1
      NDISP(IM,3) = 3*IEND(IM,1)
      NDISP(IM,4) = 3*IEND(IM,2)-2
      NDISP(IM,5) = 3*IEND(IM,2)-1
      NDISP(IM,6) = 3*IEND(IM,2)
   32 CONTINUE
      RETURN
      END
C
C......................LOAD CASES 1,2,3,4 & 5......................
C
      SUBROUTINE  FORCE(ILOAD,GF,F,NODE)
      DIMENSION   GF(240),NODE(80),F(6,80),TLR(6,6),TRL(6,6)
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE / EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),
DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)

      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM

          IF(ILOAD.EQ.1) THEN
      DO 3 IB=1,NOBM
      JJ=3*NODE(IB)-1
    3 GF(JJ) = -1.0
          ENDIF
          IF(ILOAD.EQ.2) THEN
      DO 11 I=1,NJOIN
      IF(XG(I).NE.CAPL) GO TO 11
      IF(YG(I).EQ.0.0) GO TO 11
      K=I
   11 CONTINUE
      JJ = 3*K-2
      GF(JJ) = -1.0
          ENDIF
          IF(ILOAD.EQ.3) THEN
      DO 13 I=1,NOCOL
      START =(CAPD/2.)*12.
      END   =DFX*12.
      CALL  TRANS( DC(I,1),DC(I,2),TLR,TRL )
      CALL  FIXED(-1.0,ALENTH(I),START,END,F(1,I),TLR)
      DO 13 J=1,6
```

```
   13 GF(NDISP(I,J)) = GF(NDISP(I,J))-F(J,I)
           ENDIF
           IF(ILOAD.EQ.4) THEN
      DO 14 I=1,NOCOL
      START = ALENTH(I)-WATD*12.-DFX*12.
      END   = DFX*12.
      CALL  TRANS( DC(I,1),DC(I,2),TLR,TRL )
      CALL  FIXED(-1.0,ALENTH(I),START,END,F(1,I),TLR )
      DO 14 J=1,6
   14 GF(NDISP(I,J)) = GF(NDISP(I,J))-F(J,I)
            ENDIF
      IF(ILOAD.EQ.5) THEN
      DO 16 I=NOCOL+1,NMEM
      START =0.0
      END   =0.0
      CALL  TRANS( DC(I,1),DC(I,2),TLR,TRL )
      CALL  FIXED(-1.0,ALENTH(I),START,END,F(1,I),TLR)
      DO 16 J=1,6
   16 GF(NDISP(I,J)) = GF(NDISP(I,J))-F(J,I)
      ENDIF
      RETURN
      END
C
C...........................LOAD CASE 6..........................
C
      SUBROUTINE  FORCEA(ILOAD,GF,F,NODE)
      DIMENSION   GF(240),NODE(80),F(6,80),TLR(6,6),TRL(6,6)
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE / EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),
DC(80,2)
      COMMON /FTWO / XG(80),YG(80)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
*
      COMMON/TWO/ CAPD,CAPW,CAPL,BMDF,BMDB,BMD,PDSTL,PAD,HNCH
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      COMMON/FOR/ R,D1,D2,CRV,DSMPH,V,WATD,P,CFMOM1,CFMOM2,RM2DFL
      COMMON/ATE/ PCA2(22,3),ALOAD(2,80,6),Y2,BMSPF,BMSPB,BMSP
      COMMON/NIN/ ECCDL,ECCLL,WBMF,WBMB,NOBMF,NOBMB,NOBM
*
      DO 3 IB=1,NOBM
      JJ=3*NODE(IB)-1
  3   GF(JJ) = -1.0
      RETURN
      END
C ----------------------------------------------------------------
      SUBROUTINE FIXED(Q,AL,A,B,FR,TLR)
      DIMENSION  F(6),TLR(6,6),FR(6)

      C1 = AL-(A+B)
      A1 = A+C1/2.0
      B1 = B+C1/2.0
      DEN = 12.0*AL*AL
      X1 =-Q*C1*(12.*A1*B1*B1+C1*C1*(AL-3.*B1))/DEN
```

```
      X2 = Q*C1*(12.*B1*A1*A1+C1*C1*(AL-3.*A1))/DEN
      FAC = (X1+X2)/AL
      F(1) = 0.0
      F(4) = 0.0
      F(3) = X1
      F(6) = X2
      F(2) = FAC-Q*C1*B1/AL
      F(5) =-FAC-Q*C1*A1/AL

C.......TRANSFER TO GLOBAL SYSTEM.......................
      DO 2 I=1,6
      SUM = 0.0
      DO 4 J=1,6
 4    SUM = SUM+TLR(I,J)*F(J)
 2    FR(I) = SUM
      RETURN
      END

C  ================================================================
      SUBROUTINE GSTIFF(GK)
      DIMENSION GK(240,9),EK(6,6),TRL(6,6),TLR(6,6),AXL(6,6)
      COMMON /FZERO/ NMEM,NJOIN,NACT,MBAND
      COMMON /FONE / EMOD,ENRCAP,ENRCOL,ARCAP,ARCOL,ALENTH(80),
DC(80,2)
      COMMON /FTHR / IEND(80,2),NDISP(80,6)
      COMMON/THR/ WCOL,FC,NOCOL,COLH,COLK,CM,DFX,COLSP,DEFL2
      DO 2 I=1,NACT
      DO 2 J=1,MBAND
  2   GK(I,J)=0.0
      DO 4 IM = 1,NMEM
               IF(IM.LE.NOCOL) THEN
      ENRTA = ENRCOL
      AREA  = ARCOL
                     ELSE
      ENRTA = ENRCAP
      AREA  = ARCAP
                    ENDIF

      CALL  MEMBER( EMOD,ENRTA,AREA,ALENTH(IM),EK )
      CALL  TRANS  ( DC(IM,1),DC(IM,2),TLR,TRL )
      CALL  MULT   ( TLR,EK,AXL,6,6,6 )
      CALL  MULT   ( AXL,TRL,EK,6,6,6 )

      DO 6 II=1,6
      DO 6 JJ=1,6
      I=NDISP(IM,II)
      J=NDISP(IM,JJ)
      IF(I-J) 9,8,6
  8   J=1
      GO TO 7
  9   J=J-I+1
      IF(J.GT.MBAND) GO TO 6
  7   GK(I,J) = GK(I,J)+EK(II,JJ)
```

```
    6   CONTINUE
    4   CONTINUE
        RETURN
        END
C ==================================================================
        SUBROUTINE MODSTIF( GK,JRL,N,IBW )
        DIMENSION GK(240,9),JRL(240)
        DO 10 I=1,N
        IF(JRL(I).EQ.0) GO TO 10
        JJ=I
        I1=JJ
        JR=N-JJ+1
        GK(I1,1)= 1.0
        IF(JR.EQ.1) GO TO 30
        IF(JR-IBW) 6,8,8
    6   JJJ=JR
        GO TO 9
    8   JJJ=IBW
    9   DO 2 J1=2,JJJ
        GK(I1,J1)=0.0
    2   CONTINUE
    30  DO 4 J1=2,IBW
        I1=I1-1
        IF(I1.EQ.0) GO TO 10
        GK(I1,J1) = 0.0
    4   CONTINUE
    10  CONTINUE
        RETURN
        END
C ==================================================================
        SUBROUTINE MEMBER( EMOD,ENRTA,AR,AL,EK )
        DIMENSION  EK(6,6)
        EE = EMOD*ENRTA/AL
        BB = EE/AL
        CC = BB/AL
        DD = EMOD*AR/AL
        DO 2 I=1,6
        DO 2 J=1,6
    2   EK(I,J) =0.0
        EK(1,1) = DD
        EK(1,4) =-DD
        EK(2,2) = 12.0*CC
        EK(2,3) = 6.0*BB
        EK(2,5) =-EK(2,2)
        EK(2,6) = EK(2,3)
        EK(3,3) = 4.0*EE
        EK(3,5) =-EK(2,3)
        EK(3,6) = 2.0*EE
        EK(4,4) = DD
        EK(5,5) = EK(2,2)
        EK(5,6) = EK(3,5)
        EK(6,6) = EK(3,3)
        DO 4 I=1,6
```

```
      DO 4 J=1,6
 4    EK(J,I)=EK(I,J)
      RETURN
      END
C ==================================================================
      SUBROUTINE TRANS( CX,CY,TL,TR )
      DIMENSION TL(6,6) , TR(6,6)
      DO 2 I=1,6
      DO 2 J=1,6
  2   TL(I,J) = 0.0
      TL(1,1) = CX
      TL(1,2) =-CY
      TL(2,1) = CY
      TL(2,2) = CX
      TL(3,3) = 1.
      TL(4,4) = CX
      TL(4,5) =-CY
      TL(5,4) = CY
      TL(5,5) = CX
      TL(6,6) = 1.0
      DO 4 I = 1,6
      DO 4 J = 1,6
 4    TR(I,J)= TL(J,I)
      RETURN
      END

C ==================================================================
      SUBROUTINE MULT ( A , B , R , L , M , N )
      DIMENSION  A(L,M) , B(M,N) , R(L,N)
      DO 2  I = 1 , L
      DO 2  J = 1 , N
      R(I,J) = 0.0
      DO 2 K  = 1 , M
      R(I,J) = R(I,J) + A(I,K) * B(K,J)
 2    CONTINUE
      RETURN
      END
C
=====================================================================
C  TO SOLVE  BANDED MATRIX USING GAUSS ELIMINATION METHOD
C  note: MATRIX "A" IS DESTROYED DURING OPERATION
C      : RESULTS ARE STORED IN RHS VECTOR AFTER SOLUTION
C  ------------------------------------------------------
      SUBROUTINE  SOLVER( A,RHS,N,IBW )
      DIMENSION A(240,9) , RHS(480/9)
      NM1 =N-1
      IBWM1 = IBW-1
      DO 3 I = 1 , NM1
      IF(A(I,1).EQ.0.0) GO TO 3
      DO 2 J = 1, IBWM1
      IJ = I+J
      IF(IJ.GT.N) GO TO 3
      L1 = IBW-J
```

```
      CONST = A(I,J+1)/A(I,1)
      DO 1  K = 1,L1
1     A(IJ,K) = A(IJ,K) - A(I,K+J)*CONST
2     RHS(IJ) = RHS(IJ) - RHS(I)*CONST
3     CONTINUE
      RHS(N) = RHS(N)/A(N,1)
      K = N-1
7     IF(A(K,1).NE.0.0) GO TO 8
      RHS(K) = 0.0
      GO TO 10
8     RHS(K) = RHS(K)/A(K,1)
      L2 = IBW
      IF(L2+K-1.GT.N) L2=N-K+1
      DO 9 J = 2,L2
9     RHS(K) = RHS(K) - A(K,J)*RHS(J+K-1)/A(K,1)
10    K = K-1
      IF(K.GT.0) GO TO 7
      RETURN
      END
```

APPENDIX D. LISTING OF PASCAL GRAPHICAL INTERFACE BCINPUT

```
Program bcinput(input,output);

Const
  Glb_scr_text    :  array[1..41] of string[80] =
                         ('Number of Lanes      = [       ]      ',
                          'Span Length, Long    = [       ]  ft  ',
                          'Span Length, Short   = [       ]  ft  ',
                          'Skew Angle           = [       ]  deg ',
                          'Number of Rails      = [       ]      ',
                          'Wt/ft of Rail        = [       ]  k/ft',
                          'Rail Height          = [       ]  ft  ',
                          'Number of Beams      = [       ]      ',
                          'Wt/ft of Beam        = [       ]  k/ft',
                          'Beam Depth           = [       ]  ft  ',
                          'Beam Spacing         = [       ]  ft  ',
                          'Number of Columns    = [       ]      ',
                          'Wt/ft of Column      = [       ]  k/ft',
                          'Column Height        = [       ]  ft  ',
                          'Depth to Fixity      = [       ]  ft  ',
                          'Column Spacing       = [       ]  ft  ',
                          'Column Radius        = [       ]  ft  ',
                          'Col. D Perp to Cap   = [       ]  ft  ',
                          'Col. D Along Cap     = [       ]  ft  ',
                          'Fc of Concrete       = [       ]  psi ',
                          'Column K-factor      = [       ]      ',
                          'Cm                   = [       ]      ',
                          'Slab Depth           = [       ]  ft  ',
                          'Slab Width           = [       ]  ft  ',
                          'Cap Depth            = [       ]  ft  ',
                          'Cap Width            = [       ]  ft  ',
                          'Cap Length           = [       ]  ft  ',
                          'Pedestal Height      = [       ]  ft  ',
                          'Bearing Pad Height   = [       ]  ft  ',
                          'Haunch               = [       ]  ft  ',
                          'Velocity of Stream   = [       ]  fps ',
                          'Water Depth          = [       ]  ft  ',
                          'Degree of Curve      = [       ]  deg ',
                          'Design Speed         = [       ]  mph ',
                          'Texas Design Wind    = [       ]  mph ',
                          'AASHTO Ht Above Dk   = [       ]  ft  ',
                          'Bent Cont. Mult.     = [       ]      ',
                          'Out-of-plane Defl.   = [       ]  in  ',
                          'Approx/Frame Code    = [       ]      ',
                          'Bending Axis Code    = [       ]      ',
                          'Name of Input File   = [             ]');

Var
  Glb_fld_strg : array[1..44] of string[80];
  fld_strg     : integer;
  x_prev       : integer;
  y_prev       : integer;
  finished     : boolean;
```

```
{$I typedef.sys}                    { These files must be }
{$I graphix.sys}                    { included and in this order }
{$I kernel.sys}
{$I screen.inc}


{........................span input screen......................}

Procedure Span;

Const
  C_max_field = 4;

  fld_x         :  array[1..c_max_field] of integer = (35,30,33,64);
  fld_y         :  array[1..c_max_field] of integer = (8,10,24,24);
  fld_length    :  array[1..c_max_field] of integer = (5,5,5,5);
  scr_x         :  array[1..c_max_field] of integer = (22,17,20,50);

Var
  key         :  char;
  fld_strg    :  array[1..c_max_field] of string[80];
  loop        :  integer;
  scr_text    :  array[1..c_max_field] of string[80];
  field       :  integer;


Begin

  InitGraphic;
  DefineWindow(1,1,1,XMaxGlb-1,YMaxGlb-1);
  DefineHeader(1,'SPAN VIEW');
  SetHeaderOn;
  DefineWorld(1,2.5,31.5,47.5,3.5);
  SelectWindow(1);
  SelectWorld(1);
  SetBackground(0);
  Drawborder;

  {horiz. lines}
  DrawLine(5,30,45,30);
  DrawLine(5,30.05,45,30.05);
  DrawLine(5,28,45,28);
  DrawLine(5,25,45,25);
  DrawLine(5,28.4,45,28.4);
  DrawLine(5,27.5,45,27.5);
  DrawLine(5,25.5,45,25.5);

  DrawLine(8,23,10,23);
  DrawLine(29,23,31,23);
  DrawLine(40,23,42,23);
```

```
  DrawLine(7.5,10,10.5,10);
  DrawLine(28.5,10,31.5,10);
  DrawLine(39.5,10,42.5,10);

  DrawLine(8,22.5,8,21);DrawLine(10,22.5,10,21);


DrawLine(7,21.5,8,21.5);DrawLine(10,21.5,11,21.5);DrawLine(27,23,28
.5,23);

  {section cut lines}
  DrawLine(5,30.4,5,27);              DrawLine(45,30.4,45,27);
  DrawLine(5,27,5.5,26.5);            DrawLine(45,27,45.5,26.5);
  DrawLine(5.5,26.5,4.5,26.5);        DrawLine(45.5,26.5,44.5,26.5);
  DrawLine(4.5,26.5,5,26);            DrawLine(44.5,26.5,45,26);
  DrawLine(5,26,5,24.5);              DrawLine(45,26,45,24.5);

  {beam lines}
  DrawLine(9,28,9,25);DrawLine(30,28,30,25);DrawLine(41,28,41,25);

  {cap lines}
  DrawLine(8,25,8,23);
  DrawLine(10,25,10,23);
  DrawLine(27.5,25,27.5,23);
  DrawLine(29,25,29,23);
  DrawLine(31,25,31,23);
  DrawLine(40,25,40,23);
  DrawLine(42,25,42,23);

  {column lines}
  DrawLine(8.3,23,8.3,10);
  DrawLine(9.7,23,9.7,10);
  DrawLine(29.3,23,29.3,10);
  DrawLine(30.7,23,30.7,10);
  DrawLine(40.3,23,40.3,10);
  DrawLine(41.7,23,41.7,10);

  {fixity}
  DrawLine(7.5,9.5,8,10);
  DrawLine(8.5,9.5,9,10);
  DrawLine(9.5,9.5,10,10);
  DrawLine(28.5,9.5,29,10);
  DrawLine(29.5,9.5,30,10);
  DrawLine(30.5,9.5,31,10);
  DrawLine(39.5,9.5,40,10);
  DrawLine(40.5,9.5,41,10);
  DrawLine(41.5,9.5,42,10);

  {dimension lines}

DrawLine(9,9,9,6);DrawLine(30,9,30,6);DrawLine(41,9,41,6);DrawLine(
9,7,41,7);
  DrawLine(9,7,9.5,7.3);DrawLine(9,7,9.5,6.7);
```

```
DrawLine(29.5,7.3,30.5,6.7);DrawLine(29.5,6.7,30.5,7.3);
DrawLine(41,7,40.5,7.3);DrawLine(41,7,40.5,6.7);


{close box}
writexy(1,1,$1f,' "F1" to close ');
DrawLine(1,31.5,47.6,31.5);


SetWindowModeOff;


{input fields}

scr_text[1]  := 'Cap Depth = [     ] ft';
scr_text[2]  := 'Cap Width = [     ] ft';
scr_text[3]  := 'Long Span = [     ] ft';
scr_text[4]  := 'Short Span = [     ] ft';

fld_strg[1]  := Glb_fld_strg[25];
fld_strg[2]  := Glb_fld_strg[26];
fld_strg[3]  := Glb_fld_strg[2];
fld_strg[4]  := Glb_fld_strg[3];

key := chr(00);
field := 1;

for loop := 1 to c_max_field do
  begin
    writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
    writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
  end;

While (key<> c_F1) do
  begin
    input_field(fld_x[field],fld_y[field],fld_length[field],31,
fld_strg[field],key);

    case key of
      c_down  :  begin
                   if field < c_max_field then
                       field := field + 1
                   else
                       field := 1
                 end;
      c_up    :  begin
                   if field > 1 then
                       field := field - 1
                   else
                       field := c_max_field
                 end;
      c_tab   :  begin
                   if field < c_max_field then
```

```
                        field := field + 1
                    else
                        field := 1
                  end;
        c_bktab : begin
                    if field > 1 then
                        field := field - 1
                    else
                        field := c_max_field
                 end;
        c_enter :  begin
                    if field < c_max_field then
                        field := field + 1
                    else
                        field := 1
                  end;
      end;

    end;

  SetWindowModeOn;
  LeaveGraphic;

  Glb_fld_strg[25] := fld_strg[1];
  Glb_fld_strg[26] := fld_strg[2];
  Glb_fld_strg[2]  := fld_strg[3];
  Glb_fld_strg[3]  := fld_strg[4];

End;


{.......................elevation input screen..................}

Procedure Elevation;

Const
  C_max_field = 11;

  fld_x    :  array[1..c_max_field] of integer = (67,67,67,67,67,67,
                                                   67,34,63,34,63);
  fld_y    :  array[1..c_max_field] of integer = (2,3,4,5,6,7,
                                                  9,11,14,17,21);
  fld_length :  array[1..c_max_field] of integer = (5,5,5,5,5,5,
                                                     5,5,5,5,5);
  scr_x    :  array[1..c_max_field] of integer = (45,45,45,45,45,45,
                                                   45,21,41,21,41);

Var
  key         :  char;
  fld_strg    :  array[1..c_max_field] of string[80];
  loop        :  integer;
  scr_text    :  array[1..c_max_field] of string[80];
  field       :  integer;
```

```
begin
  InitGraphic;
  DefineWindow(1, 1, 0, XMaxGlb-1, YMaxGlb-0);
  DefineHeader(1,'BENT COLUMN ELEVATION');
  SetHeaderOn;
  DefineWorld(1,-0.25,-1.2,27.7,-20);
  SelectWindow(1);
  SelectWorld(1);
  SetBackground(0);
  DrawBorder;

  {horiz. lines}
  Drawline(1.0,-2,10.0,-2);
  Drawline(1.0,-3.25,10,-3.25);
  Drawline(1.0,-3.50,10,-3.50);
  Drawline(1.0,-3.65,10,-3.65);
  Drawline(2.0,-3.90,5.25,-3.90);
  Drawline(5.75,-3.90,9,-3.90);
  Drawline(2.0,-5.20,5.25,-5.20);Drawline(5.75,-5.20,9,-5.20);
  Drawline(2.0,-5.45,5.28,-5.45);Drawline(5.75,-5.45,9,-5.45);
  Drawline(4.7,-5.60,6.3,-5.60);
  Drawline(4.7,-5.75,6.3,-5.75);
  Drawline(4.7,-7.00,6.33,-7.00);
  Drawline(1.0,-12.75,5.0,-12.75);Drawline(6.0,-12.75,10.0,-12.75);
  Drawline(1.0,-14.25,5.0,-14.25);Drawline(6.0,-14.25,10.0,-14.25);
  Drawline(1.0,-14.3,5.0,-14.3);Drawline(6.0,-14.3,10.0,-14.3);
  Drawline(5.00,-15.0,6.00,-15.0);
  Drawline(4.5,-18.5,6.5,-18.5);

  {vert. lines}
  Drawline(5.25,-3.65,5.25,-5.45);Drawline(5.75,-3.65,5.75,-5.45);
  Drawline(4.85,-5.45,4.85,-5.60);
  Drawline(5.15,-5.45,5.15,-5.60);
  Drawline(5.85,-5.45,5.85,-5.60);
  Drawline(6.15,-5.45,6.15,-5.60);
  Drawline(4.70,-5.60,4.70,-7.00);Drawline(6.30,-5.60,6.30,-7.00);
  Drawline(5.00,-7.00,5.00,-18.50);Drawline(6.00,-7.00,6.00,-
18.50);

  {section cut lines}
  DrawLine(1,-1.8,1,-2.7);           DrawLine(10,-1.8,10,-2.7);
  DrawLine(1,-2.7,1.25,-2.8);        DrawLine(10,-2.7,10.25,-2.8);
  DrawLine(1.25,-2.8,0.75,-2.8);     DrawLine(10.25,-2.8,9.75,-2.8);
  DrawLine(0.75,-2.8,1,-3);          DrawLine(9.75,-2.8,10,-3);
  DrawLine(1,-3,1,-3.9);             DrawLine(10,-3,10,-3.9);

  DrawLine(2,-3.65,2,-4.4);          DrawLine(9,-3.65,9,-4.4);
  DrawLine(2,-4.4,2.25,-4.5);        DrawLine(9,-4.4,9.25,-4.5);
  DrawLine(2.25,-4.5,1.75,-4.5);     DrawLine(9.25,-4.5,8.75,-4.5);
  DrawLine(1.75,-4.5,2,-4.7);        DrawLine(8.75,-4.5,9,-4.7);
  DrawLine(2,-4.7,2,-5.75);          DrawLine(9,-4.7,9,-5.49);
```

```
  {free surface}
  DrawLine(1.75,-12.35,2.25,-12.35);
  DrawLine(1.75,-12.35,2,-12.75);
  DrawLine(2.25,-12.35,2,-12.75);

  {fixity}
  DrawLine(4.75,-18.5,4.5,-19);
  DrawLine(5.5,-18.5,5.25,-19);
  DrawLine(6.25,-18.5,6,-19);

  {cap dimension lines}
  DrawLine(4,-7.75,4.7,-7.75);DrawLine(6.3,-7.75,7,-7.75);
  DrawLine(4.7,-7.25,4.7,-8);DrawLine(6.3,-7.25,6.3,-8);

  {dimension lines}
  Drawline(11,-2,15,-2);DrawLine(7,-5.75,15,-5.75);DrawLine(7,-
7,15,-7);
  DrawLine(11,-12.75,13,-12.75);DrawLine(7,-15,15,-15);DrawLine(7,-
18.5,15,-18.5);
  DrawLine(14.7,-2,14.7,-18.5);DrawLine(12.7,-12.75,12.7,-15);

  {arrows}
  DrawLine(14.7,-2,14.5,-2.33);DrawLine(14.7,-2,14.95,-2.33);
  DrawLine(14.45,-5.42,14.95,-6.08);DrawLine(14.45,-6.08,14.95,-
5.42);
  DrawLine(14.45,-6.7,14.95,-7.33);DrawLine(14.45,-7.3,14.95,-
6.67);
  DrawLine(14.45,-14.7,14.95,-15.33);DrawLine(14.45,-15.25,14.95,-
14.67);
  DrawLine(14.7,-18.5,14.45,-18.17);DrawLine(14.7,-18.5,14.95,-
18.17);


  {close box}
  writexy(1,1,$1f,' "F1" to close ');
  DrawLine(-0.25,-1.35,4.9,-1.35);

  SetWindowModeOff;        {screen coordinates in absolute}

  {input fields}

  scr_text[1]  := Glb_scr_text[7];
  scr_text[2]  := Glb_scr_text[23];
  scr_text[3]  := Glb_scr_text[10];
  scr_text[4]  := Glb_scr_text[30];
  scr_text[5]  := Glb_scr_text[29];
  scr_text[6]  := Glb_scr_text[28];
  scr_text[7]  := Glb_scr_text[25];
  scr_text[8]  := 'Cap Width = [      ] ft';
  scr_text[9]  := Glb_scr_text[14];
  scr_text[10] := 'Water Depth=[      ] ft';
  scr_text[11] := Glb_scr_text[15];
```

```
fld_strg[1]  := Glb_fld_strg[7];
fld_strg[2]  := Glb_fld_strg[23];
fld_strg[3]  := Glb_fld_strg[10];
fld_strg[4]  := Glb_fld_strg[30];
fld_strg[5]  := Glb_fld_strg[29];
fld_strg[6]  := Glb_fld_strg[28];
fld_strg[7]  := Glb_fld_strg[25];
fld_strg[8]  := Glb_fld_strg[26];
fld_strg[9]  := Glb_fld_strg[14];
fld_strg[10] := Glb_fld_strg[32];
fld_strg[11] := Glb_fld_strg[15];


key := chr(00);
field := 1;

for loop := 1 to c_max_field do
  begin
    writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
    writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
  end;

While (key<> c_F1) do
  begin
    input_field(fld_x[field],fld_y[field],fld_length[field],31,
fld_strg[field],key);

    case key of
      c_down  :  begin
                   if field < c_max_field then
                       field := field + 1
                   else
                       field := 1
                 end;
      c_up    :  begin
                   if field > 1 then
                       field := field - 1
                   else
                       field := c_max_field
                 end;
      c_enter :  begin
                   if field < c_max_field then
                       field := field + 1
                   else
                       field := 1
                 end;
    end;

  end;

SetWindowModeOn;
LeaveGraphic;
```

```
  Glb_fld_strg[7]  := fld_strg[1];
  Glb_fld_strg[23] := fld_strg[2];
  Glb_fld_strg[10] := fld_strg[3];
  Glb_fld_strg[30] := fld_strg[4];
  Glb_fld_strg[29] := fld_strg[5];
  Glb_fld_strg[28] := fld_strg[6];
  Glb_fld_strg[25] := fld_strg[7];
  Glb_fld_strg[26] := fld_strg[8];
  Glb_fld_strg[14] := fld_strg[9];
  Glb_fld_strg[32] := fld_strg[10];
  Glb_fld_strg[15] := fld_strg[11];

end; { Elevation }


{..........................plan input screen.....................}

Procedure Plan;

Const
  C_max_field = 8;

  fld_x      :  array[1..c_max_field] of integer =
                                         (30,30,18,30,43,67,56,70);
  fld_y      :  array[1..c_max_field] of integer =
                                         (7,11,14,18,24.3,12,15);
  fld_length :  array[1..c_max_field] of integer =
                                         (5,5,5,5,5,5,5,5);
  scr_x      :  array[1..c_max_field] of integer =
                                         (15,15,3,15,28,52,53,55);

Var
  key        :  char;
  fld_strg   :  array[1..c_max_field] of string[80];
  loop       :  integer;
  scr_text   :  array[1..c_max_field] of string[80];
  field      :  integer;

Begin

  InitGraphic;
  DefineWindow(1,1,1,XMaxGlb-1,YMaxGlb-1);
  DefineHeader(1,'PLAN VIEW');
  SetHeaderOn;
  DefineWorld(1,59,63.4,163,-2.1);
  SelectWindow(1);
  SelectWorld(1);
  SetBackground(0);
  Drawborder;

  {cap}
  Drawline(114.9,9.8,114.9,55.2);Drawline(114.9,9.8,121.1,9.8);
  Drawline(121.1,9.8,121.1,55.2);Drawline(121.1,55.2,114.7,55.2);
```

```
  Drawline(115,10,115,55);Drawline(115,10,121,10);
  Drawline(121,10,121,55);Drawline(121,55,114.9,55);

  Drawline(115.1,10,115.1,55);Drawline(115,10.2,121,10.2);
  Drawline(120.9,10,120.9,55);Drawline(121,54.8,115,54.8);


  {dimension lines}
  Drawline(70,10,113,10);Drawline(71,10,71,55);Drawline(70,55,113,
55);

  Drawline(110,15,113,15);Drawline(111,15,111,22);Drawline(110,22,
113,22);

  Drawline(110,32.5,113,32.5);Drawline(111,32.5,111,42);Drawline(
110,42,113,42);

  Drawline(110,47,113,47);Drawline(111,47,111,49.5);Drawline(110,
49.5,113,49.5);

  Drawline(115.5,5,115.5,8);Drawline(115.5,6,120.5,6);Drawline(
120.5,5,120.5,8);

  Drawline(123,18.5,126,18.5);Drawline(125,18.5,125,47);Drawline(
123,47,126,47);

  Drawline(122,6.3,138.7,2.8);Drawline(137.5,3.2,149.5,51);Drawline
(122,56,151,50.67);

  {circular column}
  SetAspect(1);
  DrawCircle(118,47,0.165);

  {rectangular column}
  SetLineStyle(4);
  Drawline(115.5,15,120.5,15);
  Drawline(115.5,15,115.5,22);
  Drawline(115.5,22,120.5,22);
  Drawline(120.5,15,120.5,22);

  {centerline}
  Drawline(118,8,118,61);

  {skew anlge dimension line}
  SetLineStyle(0);
  Drawline(118,32.5,125,60.5);
  Drawline(118,60,124.8,59.5);

  {beams}
  DrawLine(116,13.33,120,12.67);DrawLine(116,13.6,120,12.8);
  DrawLine(116,24.45,120,23.5);DrawLine(116,24.55,120,23.7);
  DrawLine(116,33.33,120,32.13);DrawLine(116,33.1,120,31.95);
```

```
DrawLine(116,42.13,120,41.2);DrawLine(116,42.33,120,41.4);
DrawLine(116,52.23,120,51.3);DrawLine(116,52.4,120,51.5);

{close box}
writexy(1,1,$1f,' "F1" to close ');
DrawLine(59,63.4,100,63.4);

SetWindowModeOff;        {screen coordinates in absolute}

{input fields}

scr_text[1]  := 'Col. Radius = [      ] ft';
scr_text[2]  := 'Beam Spacing= [      ] ft';
scr_text[3]  := ' Cap Length = [      ] ft';
writexy(15,17,$1f,'  Col Depth');
scr_text[4]  := '  Along Cap = [      ] ft';
writexy(28,23,$1f,'Col. Depth');
scr_text[5]  := 'Perp to Cap = [      ] ft';
scr_text[6]  := ' Skew Angle = [      ] deg';
writexy(53,11,$1f,'Col. Spacing');
scr_text[7]  := '= [      ] ft';
scr_text[8]  := ' Slab Width = [      ] ft';

fld_strg[1]  := Glb_fld_strg[17];
fld_strg[2]  := Glb_fld_strg[11];
fld_strg[3]  := Glb_fld_strg[27];
fld_strg[4]  := Glb_fld_strg[19];
fld_strg[5]  := Glb_fld_strg[18];
fld_strg[6]  := Glb_fld_strg[4];
fld_strg[7]  := Glb_fld_strg[16];
fld_strg[8]  := Glb_fld_strg[24];

key := chr(00);
field := 1;

for loop := 1 to c_max_field do
  begin
    writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
    writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
  end;

While (key<> c_F1) do
  begin
    input_field(fld_x[field],fld_y[field],fld_length[field],31,
    fld_strg[field],key);

    case key of
      c_down  :  begin
                   if field < c_max_field then
                       field := field + 1
                   else
                       field := 1
                 end;
```

```
        c_up    :  begin
                      if field > 1 then
                          field := field - 1
                      else
                          field := c_max_field
                    end;
        c_tab   :   begin
                      if field = 1 then field := 6;
                      if field = 2 then field := 7;
                      if field = 3 then field := 8;
                      if field = 4 then field := 8;
                    end;
        c_bktab : begin
                      if field = 6 then field := 1;
                      if field = 7 then field := 2;
                      if field = 8 then field := 4;
                   end;
        c_enter :  begin
                      if field < c_max_field then
                          field := field + 1
                      else
                          field := 1
                    end;
        c_enter :  begin
                      if field < c_max_field then
                          field := field + 1
                      else
                          field := 1
                    end;
      end;

    end;

  SetWindowModeOn;
  LeaveGraphic;

  Glb_fld_strg[17] := fld_strg[1];
  Glb_fld_strg[11] := fld_strg[2];
  Glb_fld_strg[27] := fld_strg[3];
  Glb_fld_strg[19] := fld_strg[4];
  Glb_fld_strg[18] := fld_strg[5];
  Glb_fld_strg[4]  := fld_strg[6];
  Glb_fld_strg[16] := fld_strg[7];
  Glb_fld_strg[24] := fld_strg[8];

End;


{......................full sheet input screen...................}

Procedure FullSheet;

Const
```

```
C_max_field = 44;
fld_x :  array[1..c_max_field] of integer = (23,23,23,23,23,23,23
                                            ,23,23,23,23,23,23,23
                                            ,23,23,23,23,23,23,23
                                            ,23,62,62,62,62,62,62
                                            ,62,62,62,62,62,62,62
                                            ,62,62,62,62,62,62
                                                 ,11,11,11);
fld_y :  array[1..c_max_field] of integer = (1,2,3,4
                                            ,6,7,8
                                            ,10,11,12,13
                                            ,15,16,17,18,19,20,21
                                            ,22,23,24,25
                                            ,1,2,3,4,5
                                            ,7,8,9
                                            ,11,12,13,14
                                            ,16,17,18,19
                                            ,21,22,23
                                            ,12,14,16);
fld_length :  array[1..c_max_field] of integer =
                                          (5,5,5,5,5,5,5,5,5,5
                                          ,5,5,5,5,5,5,5,5,5,5
                                          ,5,5,5,5,5,5,5,5,5,5
                                          ,5,5,5,5,5,5,5,5,5,5,11
                                          ,60,60,60);

scr_x      :  array[1..c_max_field] of integer = (1,1,1,1,1,1,1
                                                 ,1,1,1,1,1,1,1
                                                 ,1,1,1,1,1,1,1
                                           ,1,40,40,40,40,40,40
                                           ,40,40,40,40,40,40,40
                                           ,40,40,40,40,40,40
                                                    ,10,10,10);


Var
  key        :  char;
  fld_strg   :  array[1..c_max_field] of string[80];
  scr_text   :  array[1..44] of string[80];
  tmp_strg   :  array[1..40] of string[5];
  field      :  integer;
  loop       :  integer;
  FilVar     :  text;
  FileName   :  string[11];
  tmp        :  string[1];


Begin

  for loop := 1 to c_max_field - 3 do
    begin
```

```
    scr_text[loop] := Glb_scr_text[loop];
    fld_strg[loop] := Glb_fld_strg[loop];
  end;

field := 1;
key := chr(00);
clear_screen(0);

for loop := 1 to c_max_field-4 do
  begin
    writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
    writexy(fld_x[loop],fld_y[loop],$1f,fld_strg[loop]);
  end;

begin
  writexy(scr_x[41],fld_y[41],$1A,scr_text[41]);
  writexy(fld_x[41],fld_y[41],28,fld_strg[41]);
end;

{close box}
writexy(59,25,28,' "F1" to close ');


While (key <> c_F1) do
  begin

    input_field(fld_x[field],fld_y[field],fld_length[field],31,
    fld_strg[field],key);

    case key of
      c_tab    :  begin
                    if field = 5  then field := field + 23;
                    if field = 8  then field := field + 23;
                    if field = 12 then field := field + 23;
                    if field < 18 then field := field + 22;
                    if field = 18 then field := field + 21;
                    if field = 19 then field := field + 21;
                    if field = 20 then field := field + 21;
                    if field = 21 then field := field + 20;
                    if field = 22 then field := field + 19;
                  end;
      c_bktab  :  begin
                    if field = 41 then field := field - 19;
                    if field = 40 then field := field - 21;
                    if field = 39 then field := field - 21;
                    if field > 22 then field := field - 22;
                  end;
      c_down    :  begin
                    if field < c_max_field - 3 then
                        field := field + 1
                    else
                        field := 1;
                  end;
```

```
        c_enter   :  begin
                      if field < c_max_field - 3 then
                          field := field + 1
                      else
                          field := 1;
                    end;
        c_up      :  begin
                      if field > 1 then
                          field := field - 1
                      else
                          field := c_max_field - 3;
                    end;
      end;
    end;

  for loop := 1 to c_max_field - 3 do
    begin
      Glb_fld_strg[loop] := fld_strg[loop];
    end;

End;


{...........................file exist?.........................}

Type
  FileName  = string[11];

Function Open(var fp:text; name: Filename): boolean;
  begin
    Assign(fp,Name);
    {$I-}
    reset(fp);
    {$I+}
    If IOresult <> 0 then
     begin
      Open := False;
      close(fp);
     end
    else
      Open := True;
  end { Open };

{..............................open..............................}

Procedure OpenFile;

Const
  c_max_field = 44;

Var
  key         :  char;
  fld_strg    :  array[1..c_max_field] of string[80];
```

```
  fld_x       :  array[1..c_max_field] of integer;
  fld_y       :  array[1..c_max_field] of integer;
  fld_length  :  array[1..c_max_field] of integer;
  scr_x       :  array[1..c_max_field] of integer;
  scr_text    :  array[1..44] of string[80];
  tmp_strg    :  array[1..40] of string[5];
  tmp_card    :  array[1..3]  of string[60];
  field       :  integer;
  loop        :  integer;
  FilVar      :  text;
  FileName    :  string[11];
  tmp         :  string[1];

Label Cancel,EditName;

Begin

  fld_x[41]       := 42;
  fld_y[41]       := 19;
  fld_length[41]  := 11;
  scr_x[41]       := 20;

  scr_text[41]    := 'EXISTING File Name = [           ]';

  fld_strg[41]    := Glb_fld_strg[41];


  key := chr(00);
  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'               OPEN EXISTING FILE
');

  begin
    writexy(15,12,$1f,' Follow standard DOS filename procedure. ');
    writexy(scr_x[41],fld_y[41],$1A,scr_text[41]);
  end;

  EditName:
    begin
      writexy(20,22,0,'
');
      writexy(20,23,0,'
');
      writexy(20,24,0,'
');
    end;

  while (key <> c_F1) do
    begin
      input_field(fld_x[41],fld_y[41],fld_length[41],$1f,
      fld_strg[41],key);
    end;
```

```
  FileName := fld_strg[41];

  if Open(FilVar,FileName) then
    begin
      assign(FilVar,FileName);
      reset(FilVar);
      readln(FilVar,tmp_strg[39]);
      readln(FilVar,tmp_card[1]);
      readln(FilVar,tmp_card[2]);
      readln(FilVar,tmp_card[3]);
      readln(FilVar,tmp_strg[40]);
      readln(FilVar,tmp_strg[2],tmp,tmp_strg[3],tmp,tmp_strg[4],
tmp,tmp_strg[1]);
      readln(FilVar,tmp_strg[6],tmp,tmp_strg[5],tmp,tmp_strg[7]);
      readln(FilVar,tmp_strg[23],tmp,tmp_strg[24]);
      readln(FilVar,tmp_strg[25],tmp,tmp_strg[26],tmp,
tmp_strg[27]);
      readln(FilVar,tmp_strg[9],tmp,tmp_strg[8],tmp,tmp_strg[10],
tmp,tmp_strg[11]);
      readln(FilVar,tmp_strg[28],tmp,tmp_strg[29],tmp,
tmp_strg[30]);
      readln(FilVar,tmp_strg[13],tmp,tmp_strg[20],tmp,tmp_strg[12],
tmp,tmp_strg[14],tmp,tmp_strg[16],tmp,tmp_strg[21],tmp,tmp_strg[22]
,tmp, tmp_strg[15],tmp,tmp_strg[38]);
      readln(FilVar,tmp_strg[17],tmp,tmp_strg[18],tmp,
tmp_strg[19]);
      readln(FilVar,tmp_strg[33],tmp,tmp_strg[34]);
      readln(FilVar,tmp_strg[31],tmp,tmp_strg[32]);
      readln(FilVar,tmp_strg[35],tmp,tmp_strg[36],tmp,
tmp_strg[37]);

      close(FilVar);

      for loop := 1 to 40 do
        begin
          fld_strg[loop] := tmp_strg[loop];
        end;
      for loop := 1 to c_max_field - 3 do
        begin
          Glb_fld_strg[loop] := fld_strg[loop];
        end;

      Glb_fld_strg[42] := tmp_card[1];
      Glb_fld_strg[43] := tmp_card[2];
      Glb_fld_strg[44] := tmp_card[3];
    end
  else
    begin
      key := chr(00);
      writexy(20,22,$1f,' File Not Found. ');
      writexy(51,22, 28,' C');
```

```
      writexy(53,22,$1f,  'ancel   ');
      writexy(51,23, 28,' E');
      writexy(53,23,$1f,  'dit Name');

      repeat
        read(KBD,key);
        if key = 'C' then
          begin
            goto Cancel;
          end;
        if key = 'E' then
          begin
            goto EditName;
          end;
        until key in ['C','E'];
    end;

  Glb_fld_strg[41] := fld_strg[41];

  Cancel:

End;


{..............................save.............................}

Procedure SaveFile;

Const
  c_max_field = 44;

Var
  key          :  char;
  fld_strg     :  array[1..c_max_field] of string[80];
  fld_x        :  array[1..c_max_field] of integer;
  fld_y        :  array[1..c_max_field] of integer;
  fld_length   :  array[1..c_max_field] of integer;
  scr_x        :  array[1..c_max_field] of integer;
  scr_text     :  array[1..44] of string[80];
  tmp_strg     :  array[1..40] of string[5];
  field        :  integer;
  loop         :  integer;
  FilVar       :  text;
  FileName     :  string[11];
  tmp          :  string[1];

Label Cancel,EditName,Ok;


Begin

  fld_x[41]       := 41;
  fld_y[41]       := 19;
```

```
  fld_length[41] := 11;
  scr_x[41]      := 20;

  scr_text[41]   := 'SAVE AS File Name = [            ]';

  for loop := 1 to c_max_field do
    begin
      fld_strg[loop] := Glb_fld_strg[loop];
    end;

  clear_screen(0);

  for loop := 1 to c_max_field - 3 do
    begin
      if fld_strg[loop] = '     ' then
        begin
          writexy(17,10,$1f,' Input not completed, some input
fields blank! ');
          writexy(35,13,28, ' C');
          writexy(37,13,$1f,  'ancel Save ');
          writexy(35,14,$1f,'             ');
          writexy(35,15,28, ' O');
          writexy(37,15,$1f,  'k, Proceed ');

          repeat
            read(KBD,key);
            if key = 'C' then
              begin
                goto Cancel;
              end;
            if key = 'O' then
              begin
                goto Ok;
              end;
            until key in ['C','O'];
        end;
      end;

  Ok:

  key := chr(00);
  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                           SAVE FILE
');

  begin
    writexy(15,12,$1f,' Follow standard DOS filename procedure. ');
    writexy(scr_x[41],fld_y[41],$1A,scr_text[41]);
  end;

  EditName:
    begin
```

```
      writexy(20,22,0,'
');
      writexy(20,23,0,'
');
      writexy(20,24,0,'
');
    end;

  while (key <> c_F1) do
    begin
      input_field(fld_x[41],fld_y[41],fld_length[41],$1f,
      fld_strg[41],key);
    end;

  key := chr(00);
  FileName := fld_strg[41];
  if Open(FilVar,FileName) then
    begin
      writexy(20,22,$1f,' File Already Exists. ');
      writexy(51,22, 28,' R');
      writexy(53,22,$1f,  'eplace  ');
      writexy(51,23, 28,' C');
      writexy(53,23,$1f,  'ancel   ');
      writexy(51,24, 28,' E');
      writexy(53,24,$1f,  'dit Name');

      repeat
        read(KBD,key);
        if key = 'R' then
          begin
          end;
        if key = 'C' then
          begin
            goto Cancel;
          end;
        if key = 'E' then
          begin
            goto EditName;
          end;
        until key in ['R','C','E'];
    end;

  begin
    FileName := fld_strg[41];
    assign(Filvar,FileName);
    rewrite(FilVar);
    writeln(FilVar,fld_strg[39]);
    writeln(FilVar,fld_strg[42]);
    writeln(FilVar,fld_strg[43]);
    writeln(FilVar,fld_strg[44]);
    writeln(FilVar,fld_strg[40]);
    writeln(FilVar,fld_strg[2],' ',fld_strg[3],' ',fld_strg[4],'
',fld_strg[1]);
```

```
    writeln(FilVar,fld_strg[6],' ',fld_strg[5],' ',fld_strg[7]);
    writeln(FilVar,fld_strg[23],' ',fld_strg[24]);
    writeln(FilVar,fld_strg[25],' ',fld_strg[26],' ',fld_strg[27]);
    writeln(FilVar,fld_strg[9],' ',fld_strg[8],' ',fld_strg[10],'
',fld_strg[11]);
    writeln(FilVar,fld_strg[28],' ',fld_strg[29],' ',fld_strg[30]);
    writeln(FilVar,fld_strg[13],' ',fld_strg[20],' ',fld_strg[12],'
',fld_strg[14]
              ,' ',fld_strg[16],' ',fld_strg[21],' ',fld_strg[22],'
',fld_strg[15]
              ,' ',fld_strg[38]);
    writeln(FilVar,fld_strg[17],' ',fld_strg[18],' ',fld_strg[19]);
    writeln(FilVar,fld_strg[33],' ',fld_strg[34]);
    writeln(FilVar,fld_strg[31],' ',fld_strg[32]);
    writeln(FilVar,fld_strg[35],' ',fld_strg[36],' ',fld_strg[37]);

    close(FilVar)
  end;

  Glb_fld_strg[41] := fld_strg[41];

  Cancel:

End;


{...........................new file.............................}

Procedure NewFile;

Const
  C_max_field = 1;

  fld_x          :  array[1..c_max_field] of integer = (39);
  fld_y          :  array[1..c_max_field] of integer = (19);
  fld_length     :  array[1..c_max_field] of integer = (11);
  scr_x          :  array[1..c_max_field] of integer = (22);

Var
  key         :  char;
  fld_strg    :  array[1..c_max_field] of string[80];
  loop        :  integer;
  scr_text    :  array[1..c_max_field] of string[80];
  field       :  integer;
  FileName    :  string[11];
  FilVar      :  text;

Label Cancel,EditName;

Begin

  scr_text[1]   := 'NEW File Name = [           ]';
```

```
  key := chr(00);
  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                          NEW FILE
');

  begin
    writexy(15,12,$1f,' Follow standard DOS filename procedure. ');
    writexy(scr_x[1],fld_y[1],$1A,scr_text[1]);
  end;

  EditName:
    begin
      writexy(20,22,0,'
');
      writexy(20,23,0,'
');
      writexy(20,24,0,'
');
    end;

  while (key <> c_F1) do
    begin
      input_field(fld_x[1],fld_y[1],fld_length[1],$1f,
      fld_strg[1],key);
    end;

  key := chr(00);
  FileName := fld_strg[1];
  if Open(FilVar,FileName) then
    begin
      writexy(22,22,$1f,' File Already Exists. ');
      writexy(51,22, 28,' R');
      writexy(53,22,$1f,  'eplace  ');
      writexy(51,23, 28,' C');
      writexy(53,23,$1f,  'ancel   ');
      writexy(51,24, 28,' E');
      writexy(53,24,$1f,  'dit Name');

      close(FilVar);

      repeat
        read(KBD,key);
        if key = 'R' then
          begin
          end;
        if key = 'C' then
          begin
            goto Cancel;
          end;
        if key = 'E' then
          begin
            goto EditName;
```

```
        end;
      until key in ['R','C','E'];
  end;

Glb_fld_strg[41] := fld_strg[1];

Glb_fld_strg[1]  := '     ';
Glb_fld_strg[2]  := '     ';
Glb_fld_strg[3]  := '     ';
Glb_fld_strg[4]  := '     ';
Glb_fld_strg[5]  := '2    ';
Glb_fld_strg[6]  := '     ';
Glb_fld_strg[7]  := '     ';
Glb_fld_strg[8]  := '     ';
Glb_fld_strg[9]  := '     ';
Glb_fld_strg[10] := '     ';
Glb_fld_strg[11] := '     ';
Glb_fld_strg[12] := '     ';
Glb_fld_strg[13] := '     ';
Glb_fld_strg[14] := '     ';
Glb_fld_strg[15] := '     ';
Glb_fld_strg[16] := '     ';
Glb_fld_strg[17] := '     ';
Glb_fld_strg[18] := '0.0  ';
Glb_fld_strg[19] := '0.0  ';
Glb_fld_strg[20] := '3600.';
Glb_fld_strg[21] := '1.25 ';
Glb_fld_strg[22] := '1.0  ';
Glb_fld_strg[23] := '     ';
Glb_fld_strg[24] := '     ';
Glb_fld_strg[25] := '     ';
Glb_fld_strg[26] := '     ';
Glb_fld_strg[27] := '     ';
Glb_fld_strg[28] := '0.125';
Glb_fld_strg[29] := '0.083';
Glb_fld_strg[30] := '0.083';
Glb_fld_strg[31] := '     ';
Glb_fld_strg[32] := '     ';
Glb_fld_strg[33] := '     ';
Glb_fld_strg[34] := '55.0 ';
Glb_fld_strg[35] := '80.0 ';
Glb_fld_strg[36] := '6.0  ';
Glb_fld_strg[37] := '1.0  ';
Glb_fld_strg[38] := '0.0  ';
Glb_fld_strg[39] := '2    ';
Glb_fld_strg[40] := '3    ';

Glb_fld_strg[42] := 'Line Number 1';
Glb_fld_strg[43] := 'Line Number 2';
Glb_fld_strg[44] := 'Line Number 3';

Cancel:
```

```
End;

{........................analysis method.......................}

Procedure Method;

Const
  C_max_field = 1;

  fld_x         :  array[1..c_max_field] of integer = (48);
  fld_y         :  array[1..c_max_field] of integer = (19);
  fld_length    :  array[1..c_max_field] of integer = (5);
  scr_x         :  array[1..c_max_field] of integer = (27);

Var
  key        :  char;
  fld_strg   :  array[1..c_max_field] of string[80];
  loop       :  integer;
  scr_text   :  array[1..c_max_field] of string[80];
  field      :  integer;

Begin

  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                METHOD OF ANALYSIS
');


  scr_text[1] := 'Approx/Frame Code = [     ]';

  fld_strg[1] := Glb_fld_strg[39];

  key := chr(00);
  field := 1;

  writexy(18, 8,$1f,' Code = 1:  TSDHPT Approximate Analysis only
');
  writexy(18,10,$1f,' Code = 2:  Both Approximate and Frame
Analysis ');

  for loop := 1 to c_max_field do
    begin
      writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
      writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
    end;

  While (key<> c_F1) do
    begin
      input_field(fld_x[field],fld_y[field],fld_length[field],31,
      fld_strg[field],key);

      case key of
```

```
        c_down  :  begin
                     if field < c_max_field then
                         field := field + 1
                     else
                         field := 1
                   end;
        c_up    :  begin
                     if field > 1 then
                         field := field - 1
                     else
                         field := c_max_field
                   end;
        c_enter :  begin
                     if field < c_max_field then
                         field := field + 1
                     else
                         field := 1
                   end;
      end;
    end;

  Glb_fld_strg[39] := fld_strg[1];

End;


{.......................bending axis option......................}

Procedure OptionAxis;

Const
  C_max_field = 1;

  fld_x           :  array[1..c_max_field] of integer = (48);
  fld_y           :  array[1..c_max_field] of integer = (19);
  fld_length      :  array[1..c_max_field] of integer = (5);
  scr_x           :  array[1..c_max_field] of integer = (27);

Var
  key         :  char;
  fld_strg    :  array[1..c_max_field] of string[80];
  loop        :  integer;
  scr_text    :  array[1..c_max_field] of string[80];
  field       :  integer;

Begin

  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                    BENDING AXIS CODE
');
```

```
  scr_text[1] := 'Bending Axis Code = [      ]';

  fld_strg[1] := Glb_fld_strg[40];

  key := chr(00);
  field := 1;

  writexy(18, 8,$1f,' Code = 1:  In-plane (x-axis) bending only
');
  writexy(18,10,$1f,' Code = 2:  Out-of-plane (y-axis) bending only
');
  writexy(18,12,$1f,' Code = 3:  Biaxial bending (both axes)
');

  for loop := 1 to c_max_field do
    begin
      writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
      writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
    end;

  While (key<> c_F1) do
    begin
      input_field(fld_x[field],fld_y[field],fld_length[field],31,
      fld_strg[field],key);

      case key of
        c_down  :  begin
                     if field < c_max_field then
                         field := field + 1
                     else
                         field := 1
                   end;
        c_up    :  begin
                     if field > 1 then
                         field := field - 1
                     else
                         field := c_max_field
                   end;
        c_enter :  begin
                     if field < c_max_field then
                         field := field + 1
                     else
                         field := 1
                   end;
      end;
    end;

  Glb_fld_strg[40] := fld_strg[1];

End;

{..............................cards.............................}
```

```
Procedure Cards;

Const
  C_max_field = 3;

  fld_x          :  array[1..c_max_field] of integer = (11,11,11);
  fld_y          :  array[1..c_max_field] of integer = (12,14,16);
  fld_length     :  array[1..c_max_field] of integer = (60,60,60);
  scr_x          :  array[1..c_max_field] of integer = (10,10,10);

Var
  key        :  char;
  fld_strg   :  array[1..c_max_field] of string[80];
  loop       :  integer;
  scr_text   :  array[1..c_max_field] of string[80];
  field      :  integer;

Begin

  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                PROBLEM DESCRIPTION
');

  scr_text[1] := '[
]';
  scr_text[2] := '[
]';
  scr_text[3] := '[
]';

  fld_strg[1] := Glb_fld_strg[42];
  fld_strg[2] := Glb_fld_strg[43];
  fld_strg[3] := Glb_fld_strg[44];

  key := chr(00);
  field := 1;

  for loop := 1 to c_max_field do
    begin
      writexy(scr_x[loop],fld_y[loop],$1A,scr_text[loop]);
      writexy(fld_x[loop],fld_y[loop], 28,fld_strg[loop]);
    end;

  While (key<> c_F1) do
    begin
      input_field(fld_x[field],fld_y[field],fld_length[field],31,
      fld_strg[field],key);

      case key of
        c_down  :  begin
                      if field < c_max_field then
```

```
                        field := field + 1
                    else
                        field := 1
                  end;
        c_up    :  begin
                    if field > 1 then
                        field := field - 1
                    else
                        field := c_max_field
                  end;
        c_enter :  begin
                    if field < c_max_field then
                        field := field + 1
                    else
                        field := 1
                  end;
      end;
    end;

  Glb_fld_strg[42] := fld_strg[1];
  Glb_fld_strg[43] := fld_strg[2];
  Glb_fld_strg[44] := fld_strg[3];

End;


{..............................help...............................}

Procedure PleaseHelp;

Var
  key       : char;
  nothing   : string[80];

Begin

  clear_screen(0);
  writexy(1,1,28,' "F1" to close  ');
  writexy(16,1,$1f,'                    HELP SCREEN
');

  writexy(5, 6,$1f,' 1.  Cursor movements are controlled by arrow
keys, tab, shift-tab,       ');
  writexy(5, 7,$1f,'     backspace, and enter keys.
');
  writexy(5,10,$1f,' 2.  Items listed in each menu are selected by
pressing the first letter  ');
  writexy(5,11,$1f,'     of the key word (shown in red).
');
  writexy(5,14,$1f,' 3.  All windows are closed by pressing the F1
function key.  Each screen ');
  writexy(5,15,$1f,'     must be closed before entering another
window.                        ');
```

```
  writexy(5,18,$1f,' 4.  Naming of files must follow standard DOS
filename procedures.  Also, ');
  writexy(5,19,$1f,'     the filename must begin in the first space
of the input field.       ');

  nothing := '';
  key := chr(00);
  While (key <> c_F1) do
    begin
      input_field(79,25,0,31,nothing,key);
  end;

End;


{..............................menus.............................}

Procedure MenuIBM;

Const
  c_menu           = 5;
  menu_x           : array[1..c_menu] of integer = (1,10,20,44,60);

Var
  key              : char;
  loop             : integer;
  menu_strg        : array[1..c_menu] of string[80];
  field            : integer;


Begin

  menu_strg[1] := '   HOME    ';
  menu_strg[2] := '  FILES     ';
  menu_strg[3] := '   OPTIONS/DESCRIPTION   ';
  menu_strg[4] := ' INPUT SCREENS   ';
  menu_strg[5] := '  HELP    ';

  x_prev := 0;
  y_prev := 0;

  key := chr(00);

  field := 1;

  while not finished do
    begin
      clear_screen(0);

      for loop := 1 to c_menu do
        begin
          writexy(menu_x[loop],1,$1a,menu_strg[loop]);
        end;
```

```
case field of
  2         : begin
               writexy(menu_x[2],2,$1f,'          ');
               writexy(menu_x[2],3, 28,'   N');
               writexy(menu_x[2]+4,3,$1f,  'ew     ');
               writexy(menu_x[2],4, 28,'   O');
               writexy(menu_x[2]+4,4,$1f,  'pen    ');
               writexy(menu_x[2],5, 28,'   S');
               writexy(menu_x[2]+4,5,$1f,  'ave    ');
               writexy(menu_x[2],6,$1f,'__________');
               writexy(menu_x[2],7,$1f,'          ');
               writexy(menu_x[2],8, 28,'   Q');
               writexy(menu_x[2]+4,8,$1f,  'uit    ');
               writexy(menu_x[2],9,$1f,'          ');
            end;
  3         : begin
               writexy(menu_x[3],2,$1f,'
');

               writexy(menu_x[3],3, 28,'  P');
               writexy(menu_x[3]+3,3,$1f,  'rocedure for
Analysis ');

               writexy(menu_x[3],4, 28,'  O');
               writexy(menu_x[3]+3,4,$1f,  'ption of Bending
Axis ');

               writexy(menu_x[3],5, 28,'  F');
               writexy(menu_x[3]+3,5,$1f,  'ull 3-Card
Description');

               writexy(menu_x[3],6,$1f,'
');
            end;
  4         : begin
               writexy(menu_x[4],2,$1f,'                  ');
               writexy(menu_x[4],3, 28,'   S');
               writexy(menu_x[4]+4,3,$1f,  'pan           ');
               writexy(menu_x[4],4, 28,'   E');
               writexy(menu_x[4]+4,4,$1f,  'levation      ');
               writexy(menu_x[4],5, 28,'   P');
               writexy(menu_x[4]+4,5,$1f,  'lan           ');
               writexy(menu_x[4],6,$1f,'                  ');
               writexy(menu_x[4],7, 28,'   F');
               writexy(menu_x[4]+4,7,$1f,  'ull Sheet     ');
               writexy(menu_x[4],8,$1f,'                  ');
            end;
  5         : begin
               writexy(menu_x[5],2,$1f,'          ');
               writexy(menu_x[5],3, 28,'  P');
               writexy(menu_x[5]+3,3,$1f, 'lease  ');
               writexy(menu_x[5],4,$1f,'          ');
            end;
end;

menu(menu_x[field],1,$1f,menu_strg[field],key);
```

```
case key of
  c_right   : begin
                if field < c_menu then
                     field := field + 1
                else
                     field := 1
                end;

  c_left    : begin
                if field > 1 then
                     field := field - 1
                else
                     field := c_menu
                end;
end;

if field = 2 then
  begin
    case key of
      'N'   : NewFile;
      'O'   : OpenFile;
      'S'   : SaveFile;
      'Q'   : finished := true;
    end;
end;

if field = 3 then
  begin
    case key of
      'P'   : Method;
      'O'   : OptionAxis;
      'F'   : Cards;
    end;
end;

if field = 4 then
  begin
    case key of
      'S'   : Span;
      'E'   : Elevation;
      'P'   : Plan;
      'F'   : FullSheet;
    end;
end;

if field = 5 then
  begin
    case key of
      'P'   : PleaseHelp;
    end;
end;
```

```
  end;
end;

{........................main program:  bcinput..................}

Begin

  Glb_fld_strg[1]  := '     ';
  Glb_fld_strg[2]  := '     ';
  Glb_fld_strg[3]  := '     ';
  Glb_fld_strg[4]  := '     ';
  Glb_fld_strg[5]  := '2    ';
  Glb_fld_strg[6]  := '     ';
  Glb_fld_strg[7]  := '     ';
  Glb_fld_strg[8]  := '     ';
  Glb_fld_strg[9]  := '     ';
  Glb_fld_strg[10] := '     ';
  Glb_fld_strg[11] := '     ';
  Glb_fld_strg[12] := '     ';
  Glb_fld_strg[13] := '     ';
  Glb_fld_strg[14] := '     ';
  Glb_fld_strg[15] := '     ';
  Glb_fld_strg[16] := '     ';
  Glb_fld_strg[17] := '     ';
  Glb_fld_strg[18] := '0.0  ';
  Glb_fld_strg[19] := '0.0  ';
  Glb_fld_strg[20] := '3600.';
  Glb_fld_strg[21] := '1.25 ';
  Glb_fld_strg[22] := '1.0  ';
  Glb_fld_strg[23] := '     ';
  Glb_fld_strg[24] := '     ';
  Glb_fld_strg[25] := '     ';
  Glb_fld_strg[26] := '     ';
  Glb_fld_strg[27] := '     ';
  Glb_fld_strg[28] := '0.125';
  Glb_fld_strg[29] := '0.083';
  Glb_fld_strg[30] := '0.083';
  Glb_fld_strg[31] := '     ';
  Glb_fld_strg[32] := '     ';
  Glb_fld_strg[33] := '     ';
  Glb_fld_strg[34] := '55.0 ';
  Glb_fld_strg[35] := '80.0 ';
  Glb_fld_strg[36] := '6.0  ';
  Glb_fld_strg[37] := '1.0  ';
  Glb_fld_strg[38] := '0.0  ';
  Glb_fld_strg[39] := '2    ';
  Glb_fld_strg[40] := '3    ';
  Glb_fld_strg[41] := '           ';
  Glb_fld_strg[42] := 'Line Number 1';
  Glb_fld_strg[43] := 'Line Number 2';
  Glb_fld_strg[44] := 'Line Number 3';
```

```
  finished := false;

  while finished = false
    begin
      MenuIBM;
  end;

  clear_screen(WHITE);
  cursor_type(c_normal);
end.
```

LISTING OF INCLUDE FILE: {$I SCREEN.INC}

```
Type
   work_strg  =  string[80];

   REGISTER = RECORD
                 AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER;
              END;
Const
  c_normal   = 0;
  c_off      = 1;
  c_block    = 2;



  Function aMonoChrome : boolean;
      Var
        PARAMETER          : REGISTER;

    Begin
      PARAMETER.AX := $0F00;
      INTR($10,PARAMETER);
      IF (PARAMETER.AX AND $00FF) = 7 THEN aMONOCHROME := TRUE
                                      ELSE aMONOCHROME := FALSE;
     end;

  Procedure WriteXY(x,y,color : integer; strg : work_strg);

    begin
      GotoXY(x,y);
      textcolor(color and $0F);
      textbackground( (color and $F0) shr 4 );
      write(strg);
    end;

  Procedure DrawBox(x1,y1,x2,y2,color : integer);
    begin
      draw(x1,y1,x2,y1,color);
      draw(x1,y1,x1,y2,color);
      draw(x1,y2,x2,y2,color);
      draw(x2,y1,x2,y2,color);
    end;

  Procedure Circle(x,y,radius,color : integer);

      Var
        loop : integer;
        temp : integer;
```

```
      Const
        Aspect = 2;

    begin
      for loop := 1 to radius do
        begin
          temp := round(sqrt(sqr(radius) - sqr(loop)));
          plot(x - loop,y + temp,color);
          plot(x - loop,y - temp,color);
          plot(x + loop,y + temp,color);
          plot(x + loop,y - temp,color);
        end;
      for loop := 1 to radius do
        begin
          temp := round(sqrt(sqr(radius) - sqr(loop)));
          plot(x + trunc(temp * aspect),y - loop,color);
          plot(x - trunc(temp * aspect),y - loop,color);
          plot(x + trunc(temp * aspect),y + loop,color);
          plot(x - trunc(temp * aspect),y + loop,color);
        end;
    end;

PROCEDURE SCROLL_UP(X1,Y1,X2,Y2,LINES,ATTR : INTEGER);
      Var
        PARAMETER        : REGISTER;
 BEGIN
   PARAMETER.AX:=$0600 + lines;
   PARAMETER.CX:= (Y1-1) * 256 + (X1-1);
   PARAMETER.DX:= (Y2-1) * 256 + (X2-1);
   PARAMETER.BX:= ATTR * 256;
   INTR($10,PARAMETER);
 END;

PROCEDURE SCROLL_DN(X1,Y1,X2,Y2,LINES,ATTR : INTEGER);
      Var
        PARAMETER        : REGISTER;
 BEGIN
   PARAMETER.AX:=$0700 + LINES;
   PARAMETER.CX:= (Y1-1) * 256 + (X1-1);
   PARAMETER.DX:= (Y2-1) * 256 + (X2-1);
   PARAMETER.BX:= ATTR * 256;
   INTR($10,PARAMETER);
 END;

PROCEDURE LOCATE_CURSOR(XPOS,YPOS : INTEGER);

 Var
   loop   : integer;

 Begin
   gotoxy(xpos,ypos);
   SetColorBlack;
```

```
   if x_prev <> 0 then                                  {graphics cursor}
     begin
       DrawLine((x_prev-3)*8,y_prev*14-2,(x_prev+3)*8,y_prev*14-2);
       DrawLine((x_prev-3)*8,y_prev*14-1,(x_prev+3)*8,y_prev*14-1);
     end;

   for loop := 1 to 3 do
     begin
       SetColorBlack;
       if loop = 2 then SetColorWhite;
       DrawLine(((xpos+loop-2)*8-7),(ypos*14-2),((xpos+loop-
2)*8),(ypos*14-2));
       DrawLine(((xpos+loop-2)*8-7),(ypos*14-1),((xpos+loop-
2)*8),(ypos*14-1));
     end;

   x_prev := xpos;
   y_prev := ypos;

   SetColorWhite;
 End;


PROCEDURE CURSOR_TYPE(CRS_TYPE : INTEGER);
      Var
        PARAMETER         : REGISTER;
{  CRS_TYPE =  0  -  Normal cursor

               1  -  Cursor  off

               2  -  Block cursor  }


 BEGIN
   PARAMETER.AX := $C100;

   IF CRS_TYPE = 0 THEN
    BEGIN
      IF aMONOCHROME THEN
        BEGIN
          PARAMETER.CX := $0A0B;
        END
      ELSE
        BEGIN
          PARAMETER.CX := $0607;
        END;
    END;

   IF CRS_TYPE = 1 THEN
    BEGIN
     PARAMETER.CX :=$1f00;{Set cursor start/stop to wrap illegally}
                          {to turn cursor off                      }
    END;
```

```
    IF CRS_TYPE = 2 THEN
     BEGIN
       IF aMONOCHROME THEN
         BEGIN
           PARAMETER.CX := $000B;
         END
       ELSE
         BEGIN
           PARAMETER.CX := $0007;
         END;
     END;

     INTR($10,PARAMETER);
 END;


PROCEDURE CLEAR_SCREEN(ATTR : INTEGER);
      Var
        PARAMETER         : REGISTER;
  BEGIN
   PARAMETER.AX:=$0600;
   PARAMETER.CX:=$0000;
   PARAMETER.DX:=$184F;
   PARAMETER.BX:= ATTR * 256;
   INTR($10,PARAMETER);
  END;

Const
  c_F1              = 'Ä';     {    128;    }
  c_F2              = 'Å';     {    129;    }
  c_F3              = 'Ç';     {    130;    }
  c_F4              = 'É';     {    131;    }
  c_F5              = 'Ñ';     {    132;    }
  c_F6              = 'Ö';     {    133;    }
  c_F7              = 'Ü';     {    134;    }
  c_F8              = 'á';     {    135;    }
  c_F9              = 'à';     {    136;    }
  c_F10             = 'â';     {    137;    }
  c_up              = 'ä';     {    138;    }
  c_right           = 'ã';     {    139;    }
  c_down            = 'å';     {    140;    }
  c_left            = 'ç';     {    141;    }
  c_home            = 'é';     {    142;    }
  c_end             = 'è';     {    143;    }
  c_ins             = 'ê';     {    144;    }
  c_del             = 'ë';     {    145;    }
  c_pgup            = 'î';     {    148;    }
  c_pgdn            = 'ï';     {    149;    }
  c_bktab           = 'ñ';     {    150;    }
  c_tab             = 'ó';     {    151;    }
  c_bkspace         = ' ';     {      8;    }
  c_esc             = ' ';     {     27;    }
```

```
  c_enter           = 'ŏ';     {   152;    }
Function INKEY : char;

  VAR  KEY_FOUND   :  BOOLEAN;
       cmd         :  char;

BEGIN
  KEY_FOUND := FALSE;
  CMD := CHR(0);
  WHILE NOT KEY_FOUND DO
    BEGIN
      IF KEYPRESSED THEN
        BEGIN
          READ(KBD,CMD);
          IF CMD = CHR(27) THEN
            BEGIN
              IF KEYPRESSED THEN
                BEGIN
                  READ(KBD,CMD);
                     CASE CMD OF
                     'K' : CMD := CHR(141);
                     'M' : CMD := CHR(139);
                     'H' : CMD := CHR(138);
                     'P' : CMD := CHR(140);
                     'I' : CMD := CHR(148);
                     'Q' : CMD := CHR(149);
                ';'..'D' : CMD := CHR(69+ORD(CMD));
                     'G' : CMD := CHR(142);
                     'O' : CMD := CHR(143);
                     'R' : CMD := CHR(144);
                     'S' : CMD := CHR(145);
                     ' ' : CMD := CHR(146);
                     ' ' : CMD := CHR(147);
                     ' ' : CMD := CHR(150);
                     ELSE CMD := CHR(0);
                     END;
                END
              ELSE
                BEGIN
                  CMD := CHR(27);
                END;
            END;
          if (cmd = chr(9)) then cmd := chr(151);
          if (cmd = chr(13)) then cmd := chr(152);
          IF (CMD <> CHR(0)) THEN KEY_FOUND := TRUE;
        END;
    END;
  inkey := cmd;
END;
```

```
{
  input_field(x, y, limit, color, string, last_key)
    x        - x coordinate of begining of input field
    y        - y coordinate of begining of input field
    limit    - max number of character in the input field
    string   - string variable to hold the input field
    last_key - the last key entered before the field terminated
}


Procedure input_field(x,y,limit,color : integer; Var strg :
work_strg; Var last_key : char);

  Var
    x_pos : integer;
    c     : char;
    done  : boolean;

  begin
    cursor_type(c_normal);
    x_pos := x;
    done := false;
    writexy(x,y,color,strg);
    locate_cursor(x,y);

    while not done do
      begin
        c := inkey;
        case c of
          c_F1        : begin
                          cursor_type(c_off);
                          last_key := c;
                          done := true
                        end;
          c_bktab     : begin
                          cursor_type(c_off);
                          last_key := c;
                          done := true
                        end;
          c_up        : begin
                          cursor_type(c_off);
                          last_key := c;
                          done := true
                        end;
          c_left      : begin
                          if ((x_pos - x) > 0) then
                            begin
                              x_pos := x_pos - 1;
                              locate_cursor(x_pos,y);
                            end;
                        end;
          c_right     : begin
```

```
                        if ((x_pos - x + 1) < limit + 1) then
                          begin
                            x_pos := x_pos + 1;
                            locate_cursor(x_pos,y);
                          end;
                      {  else
                          begin
                            cursor_type(c_off);
                            last_key := c;
                            done := true;
                          end; }
                        end;
        c_tab         : begin
                        cursor_type(c_off);
                        last_key := c;
                        done := true;
                      end;
        c_down        : begin
                        cursor_type(c_off);
                        last_key := c;
                        done := true
                      end;
        c_enter       : begin
                        cursor_type(c_off);
                        last_key := c;
                        done := true;
                      end;
        c_bkspace     : begin
                        if (x_pos > x) then
                          begin
                            x_pos := x_pos - 1;
                            strg[x_pos - x + 1] := ' ';
                            locate_cursor(x_pos,y);
                            writexy(x,y,color,strg);
                            locate_cursor(x_pos,y);
                          end;
                      end;
      else
        if ((x_pos - x) < limit) then
          begin
            locate_cursor(x_pos,y);
            write(c);
            while (length(strg) < (x_pos - x + 1)) do strg :=
strg + ' ';
            strg[x_pos - x + 1] := c;
            x_pos := x_pos + 1;
          end;

        locate_cursor(x_pos,y);

      end;
    end;
```

```
  end;
procedure menu(x,y,color : integer; Var strg :work_strg; Var
last_key : char);

Var
  x_pos                 : integer;
  c                     : char;
  done                  : boolean;

Begin

  x_prev := 0;
  y_prev := 0;

  cursor_type(c_block);
  x_pos := x;
  done  := false;
  writexy(x,y,color,strg);
  locate_cursor(x,y);

  while not done do
    begin
      c := inkey;
      case c of
        c_left         : begin
                           cursor_type(c_off);
                           last_key := c;
                           done := true
                         end;
        c_right        : begin
                           cursor_type(c_off);
                           last_key := c;
                           done := true
                         end;
        'N','O','S','E','P','F':
                         begin
                           cursor_type(c_off);
                           last_key := c;
                           done := true
                         end;
        'Q'            : begin
                           cursor_type(c_off);
                           last_key := c;
                           done := true
                         end;
      end;
  end;
end;
```