1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

National Library
of Canada

Bibliothèque nationale
du Canada

0-315-22196-8

Canadian Theses Division Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Ecrire en lettres mouleées ou dactylographier

Full Name of Author — Nom complet de l'auteur

ALFONSO RICARDO LARREA ZURENA

Date of Birth — Date de naissance

Country of Birth — Lieu de naissance

CHILE

Permanent Address — Residence fixe

321. 1st Ave.
Verdun Quebec

Title of Thesis — Titre de la thèse

Automated Structural Drafting for
Buildings

University — Université

Carleton University

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M.ENG. (Civil)

Year this degree conferred — Année d'obtention de ce grade

1984

Name of Supervisor — Nom du directeur de thèse

Prof. J. L. Human

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film

L'auteur se réserve les autres droits de publication, ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur

Date

May 2/84

Signature

P. Chalenguel/ab

AUTOMATED STRUCTURAL DRAFTING
FOR BUILDINGS

by

© PATRICIA VALENZUELA

A thesis submitted to the Faculty of
Graduate Studies and Research in partial
fulfillment of the requirements for the
degree of

Master of Engineering

Department of Civil Engineering
Carleton University
Ottawa, Ontario
April 2, 1984

The undersigned recommend to the Faculty of Graduate
Studies and Research acceptance of the thesis

" AUTOMATED STRUCTURAL DRAFTING FOR BUILDINGS"

submitted by Patricia Valenzuela, C.E., University of Chile
in partial fulfillment of the requirements for
the degree of Master of Engineering

J. Sumar

THESIS SUPERVISOR

M. L. L. L. L.

CHAIRMAN, DEPARTMENT OF CIVIL ENGINEERING

Carleton University

April 2, 1984

iii

ABSTRACT

This thesis deals with a computer program developed to produce graphic displays of structures. The program is integrated into AMECO-17 system of structural engineering programs.

The graphic displays range from single line drawings to finished engineering drawings. The latter consist of framing plans and elevations for steel and concrete structures- drawn in accordance with manual drafting standards.

The design results of the AMECO-17 system are directly passed on to the graphic program. Thus the drawings are fully derived within the system, requiring no operator input.

The graphic information can be displayed on a variety of CRT terminals, directly plotted, or transmitted to an interactive graphics system for further processing.

The system is presently operational on Control Data Corporation CYBER series computers.

ACKNOWLEDGEMENTS

The author expresses her gratitude to Prof. J. Humar who supervised the work and provided guidance.

Dr. W. Wright suggested this area of research and gave many helpful suggestions during the course of the work.

The author, also wishes to thank Mr. A. A. Palejs, principal author of AMECO-17 system, for guidance and continuing support throughout the project.

Grateful acknowledgement is given to Multiple Access Division of CSG Ltd., specially to Mr. R. Crossan and Mr. J. Kern, for the permission to use their computing facilities in the Montreal offices.

Appreciation is extended to Mrs. M. Voitkus who typed major part of this thesis.

TABLE OF CONTENTS

	Page
TITLE PAGE	i
ACCEPTANCE SHEET	11
ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Objective of this Project	2
CHAPTER 2 OUTLINE OF THE PROJECT	5
2.1 Overview of AMECO-17 System	6
2.2 Overview of the Graphic Module	10
CHAPTER 3 DESCRIPTION OF THE GRAPHIC MODULE	13
3.1 Introduction	13
3.2 Computation of 2-D Coordinates for Vertical Frames	16
3.3 Computation of 3-D coordinates for Orthogonal Structures	19
3.4 Computation of 3-D Coordinates for Non-orthogonal Structures	23
3.5 Generation of Building Grid Lines	24
3.6 Drawing Sizes, Border and Title Block...	25
3.6.1 Border	26
3.6.2 Title Block	27

	Page
3.7 Location of Views on Drawings	29
3.8 Plotting of Views	32
3.8.1 Framing Elevations	32
3.8.2 Floor Framing Plans	44
3.8.3 Floor Plans for Steel Structures ..	44
3.8.4 Floor Plans for Concrete Structures	54
CHAPTER 4 EXECUTION OF THE PROGRAM	59
4.1 Command Description	61
4.1.1 PLOT Commands	65
4.1.2 Labelled Data in PLOT Commands ...	68
4.2 Purposes of Plotting	71
a) Verification of Input Data	71
b) Schematic Display of Design Results	75
c) Production of Final Drawings	78
CHAPTER 5 DISCUSSIONS	81
5.1 General	81
5.2 Interface with AMECO-17 Files	81
5.3 Joint Coordinates	84
5.4 Location of Views on Drawings	88
5.5 Plotting of Secondary Beams	91
5.6 Column Marks	93
5.7 Building Grid Lines	95
5.8 Evaluation of Five Sample Drawings	99

	Page
CHAPTER 6 SUMMARY AND CONCLUSIONS	103
REFERENCES	105
APPENDIX A Five Sample Drawings	
APPENDIX B Graphic Program Description	
APPENDIX C Graphic Program Source Listing	

LIST OF FIGURES

Figure		Page
1.1	Integration of Analysis, Design and Drafting Functions	4
2.1	AMECO Integrated Analysis/Design Process	7
2.2	AMECO Integrated Analysis/Design Process with Graphics	11
3.1	Orthogonal Structure	14
3.2	Non-orthogonal Structure	14
3.3	Joint Numbering	16
3.4	Computation of 2-D Coordinates	18
	a) Plane Frame with Col. 1 as Reference Stack	18
	b) Plane Frame with Col. 2 as Reference Stack	18
3.5	Column Stack Coordinates for an Orthogonal Structure	22
3.7	Standard Border for a Drawing	26
3.8	Title Block for a Drawing	28
3.9	Standard Margins	30
3.10	Multiple Views on a Drawing	31
3.12	Conventions used in Drawing the Beams	35
3.13	Vertical Bracing System and Walls	37
	a) Types of Bracing Members	37
	b) Frame with 'V' Braces and Walls	37
3.14	Column Sizes, Splice Points and Column Marks	39
3.16	Framing Elevations	42
3.17	Steel Building Plan	46

Figure		Page
3.18	Floor Plan for a Steel Building showing Secondary Beams	48
3.20	Floor Plan for a Steel Building showing column marks and Beam Marks.....	50
3.21	Dimension Lines, Grid Lines and Floor Plan Title	53
3.22	Floor Plan of a Concrete Structure	55
3.23	Floor Plan of a Concrete Structure showing the Representation of Slabs	57
4.1	PLOT Command Facsimile	64
4.2	AMECO Integrated Analysis/Design Process with Graphic ...	76
5.1	Member Connectivity Conventions	83
	a) AMECO	83
	b) Conventional Programs	83
5.2	Calculation of Column Stack Coordinates	85
5.3	A Case where Stack Coordinates cannot be Computed	86
5.4	Location of Views on Drawings	90
5.7	Example of Nested Secondary Beams	92
5.8	Example of Identical Beam Lines	92
5.9	Position of Column Marks	93
5.10	Convention used in Drawing Grid Lines	96
5.11	A Non-orthogonal Structure	97
5.12	A Complex Non-orthogonal Structure	98
B.1	Flow Chart of the Program Section for Plotting the Framing Elevations	B-5
B.2	Flow Chart of the Program Section for Plotting the Floor Plans	B-9
B.3	Flow Chart of Subroutine DIMS	B-13
B.4	Example of Internal Grid Line	B-14

CHAPTER 1

INTRODUCTION

1.1 Background

- During the past two decades great advances have been made in the application of computer graphics to many fields of human endeavor. The structural engineering profession is no exception, especially since the final product of a design engineering office are drawings - graphical displays of the structure to be built.

As early as the mid-sixties graphic output capabilities were developed for many structural analysis programs, such as STRESS - 3⁽¹⁾, STARDYNE⁽²⁾, ICES-STRUDL⁽³⁾ and others. A few years later, similar capabilities were incorporated in the well known finite element programs, such as NASTRAN⁽⁴⁾, MASAP⁽⁵⁾, ANSYS⁽⁶⁾ and others. In their basic form, these graphic displays show to scale line diagrams of the structural model with joint and member numbers on the diagrams. Deformed shapes of the model, shear and moment diagrams, and stress contours are typical other displays obtainable from these programs. For the structural design engineer, these graphic developments are invaluable for verifying the accuracy of the structural model and for evaluating the analysis results.

Computer aided preparation of engineering drawings came of age in the early seventies with the advent of interactive graphics. In their simplest form, interactive graphic systems can be looked upon as super-fast drafting machines. Today, the more advanced turn-key systems, such as APPLICON, AUTOTROL, CALMA, COMPUTER-VISION, described by W. Wright⁽⁷⁾, have emerged as very powerful graphic tools for the rapid creation of drawings. These systems are built around a large Data Base, that can for example, store complete tables of structural shape properties, a large variety of drafting symbols and character fonts, standard details and texts, etc., all quickly accessible by referencing a menu. Complete drawings can be stored in the Data Base for future retrieval, revision and reproduction.

These graphic systems are very effective for schematic work, such as circuit diagrams, cable routing layouts, piping isometrics. The efficiency of such systems drops in the preparation of drawings that have to be made to scale, especially when a large amount of design information has to be placed on the drawing. For example, a typical steel floor framing drawing can have over 250 members - beams, columns and bracing members. In such a case the input of member sizes alone can become a very lengthy and error prone operation.

It is obvious that a desirable step would be to eliminate this

manual operation of transferring data from design documents to the graphic system.

1.2 Objective of this Project

A structural engineering computer system, where the results of a design program are directly passed on to graphic program for production of drawings is illustrated in Fig. 1.1. Here the drawings are fully created within the system, requiring practically no operator input.

The objective of this project was to develop a graphic program and to integrate it into AMECO - 17^{(8),(9)} system of structural engineering programs. AMECO - 17 is a system of integrated structural engineering programs for the analysis and automated design of buildings, bridges and similar structures having steel, concrete or composite members. From the design data produced by AMECO - 17, the graphic module should automatically prepare the following types of design engineering drawings:

1. Framing plans and elevations for steel buildings
 2. Framing plans for concrete buildings
-

These drawings should be as complete as possible, meet the manual drafting standards and be of contract drawing quality.

FIGURE 1.1 Integration of Analysis, Design and Drafting Functions

CHAPTER 2

OUTLINE OF THE PROJECT

This chapter presents a brief description of the external characteristics of the AMECO-17 system, followed by an overview of the graphic module.

The "vertical" integration of the graphic module into AMECO-17 is illustrated by flow charts.

2.1 Overview of AMECO-17 System

AMECO-17 is a system of integrated structural engineering computer programs for the analysis and automated design of buildings, bridges and similar structures having steel, concrete or composite members.

The result of 40 men years of effort, the AMECO system permits the computerization and integration of all design activities - from loading and geometry generation through member design to project cost estimates. Therefore, an entire three-dimensional building of any size can be designed with one pass on the computer.

The AMECO multi-cycle analysis/design flowchart is shown in Fig. 2.1.

The user of AMECO is provided with a concise command language to input specifications for geometry, design criteria and loads.

Phase A of the multi-cycle process interprets the input commands, checks them for syntax and other errors, edits geometry and proceeds to Phase B if no fatal errors are encountered (see Fig. 2.1).

Phase B computes all gravity loads, member stiffnesses, and performs linear analysis by floor segments to establish the

FIGURE 2.1 AMECO integrated analysis/design process

initial member sizes of the structure.

Phase C carries out three-dimensional analysis by a method of joint relaxation⁽¹⁰⁾, followed by member proportioning.

Four structural analysis procedures are available:

1. Linear elastic analysis.
2. Nonlinear geometric analysis including the P-Delta effect.
3. Segmental analysis to simulate the erection of the structure.
4. Dynamic response spectrum analysis for seismic loads.

For a given structure any one of these procedures or all four of them can be requested.

Member size selection is a code dependent process. For that purpose the requirement of 12 national standards (codes) for the design of concrete and steel members are incorporated into AMECO, using WSD, USD, LSD, Plastic and Ductile design methods.

Three system of units are available: English, Metric and SI.

Concrete structure may consist of flat plates, flat slabs, voided slabs, joist slabs or beam and slab systems with columns and/or shear walls supported on footings on elastic soil.

Steel structures may have composite or noncomposite beams, pinned or continuous, open web steel joists, trusses, columns with flexible or fully rigid connections. Resistance to

horizontal forces is provided by rigid frame action, bracing members or shear walls. Foundation support is provided by footings on elastic soil.

Phase C is repeated as many times as necessary until either the changes in member properties have become less than the prescribed tolerances or the number of cycles specified by the user is reached.

Phase D retrieves from disk file the computational results produced by Phase C and outputs reports requested by the user.

Fig. 2.1 illustrates the automatic process of the program.

There are numerous commands available to the user to execute specific modules, to stop the process in any one of the phases, review and save intermediate results and then retrieve these at a later date to restart and to continue the design process.

2.2 Overview of the Graphic Module

The graphic module developed in this project has been designed to produce visual displays of a structure of varying complexities - ranging from scaled, single line diagrams to framing plans and elevations of contract drawing quality. The structure can be of steel or concrete.

The module is totally integrated in AMECO-17 system. AMECO supplies the graphic module with all the data necessary to start the graphic computations. Therefore, no additional information has to be supplied by the user to produce graphical displays of the structure. Simple commands have been developed for the user to select the desired views, to control the amount of information to be displayed and to specify at what phase of the design process these views should be displayed.

Fig. 2.2 illustrates the positions in the AMECO analysis/design process where the module can be called upon to produce graphic displays.

Position I is normally used to display scaled single line diagrams for the purpose of verifying the input data.

Position II can be used to monitor the design process by

FIGURE 2.2 AMECO integrated analysis/design process with graphics

displaying diagrams with member sizes on them for each design cycle.

Position III is normally activated to produce framing plans and elevations with the final member sizes.

For a medium to large size building, however, it is more practical to bypass Position II and save all results on disk files. These results can be retrieved at a later date and the drawings produced at Position I.

The graphic module is written in ANSI-77 Fortran. It consists of one main program and 30 subroutines and has a total of 6300 Fortran statements. The graphic module uses the Calcomp subroutines for producing the graphical information.

CHAPTER 3

DESCRIPTION OF THE GRAPHIC MODULE

3.1 Introduction

In AMECO a three dimensional structure is internally represented by a series of intersecting two dimensional vertical frames, designated as the X and Y frames.

In a true orthogonal structure, the X set consists of frames parallel to the X-global axis while the Y set consists of frames parallel to the Y-global axis, as shown in Fig.3.1. In a nonorthogonal structure the AMECO representation is identical, except, that some or all of the quasi X or Y frames are no longer parallel to the global axis. In fact the frames can even be on a curved surface as illustrated by frame X-7 in Fig.3.2.

There are two data tables in AMECO that are used to initiate the geometric computations. One table contains information pertaining to all X frames while the other has information on all Y frames.

These tables store: member lengths, direction cosines, member and frame connectivities and member properties.

FIGURE 3.1 Orthogonal Structure.

FIGURE 3.2 Non-orthogonal Structure
Frame on a curved surface

Joint coordinates are not computed in AMECO, since they are not required in a structural analysis by joint relaxation. Therefore, the first task in obtaining graphical displays of the structure is to compute the joint coordinates. This is done in two steps. The joint coordinates of each two-dimensional frame are computed first, using a local X-Z system of axes for each frame. The global X, Y, Z joint coordinates for all beam-column connections are then obtained in the second step.

Secondary beam joint coordinates (beam to beam connections) are computed at a later stage, while creating the plot files for floor plans.

3.2 Computation of Two Dimensional Joint Coordinates for each Vertical Frame.

Subroutine PLOT5 positions each frame in a matrix, with as many columns as there are column lines in the frame being processed and as many rows as there are levels in the entire structure. Processing starts with the first frame encountered in the AMECO file of frames. All X frames are processed first followed by all Y frames.

The joints in a frame are numbered in sequence along each column stack, starting from the foundation level and proceeding upwards as shown in Fig. 3.3

FIGURE 3.3 Joint numbering.

AMECO assumes that the building will have at least one level to which all columns are attached (reference level). The user should ensure that this is so, if necessary by providing suitable dummy columns.

The following steps are involved in the computation of the joint coordinates of each vertical frame, using a local X-Z system of axes :

- For each frame subroutine PLOT5 selects as a reference stack the first column stack that has no dummy columns between levels and then sets the origin at the intersection of the reference level with the reference stack.

For a reference stack:

- Subroutine STACDN5 computes the coordinates of all joints below the reference level.
- Subroutine STACUP5 computes the coordinates of all joints above the reference level.

Once all the joint coordinates in the reference stack have been computed, subroutine STACRI5 is called, to move to the next column stack to the right.

This procedure is repeated till all the joint coordinates of the last stack of the frame have been computed. Fig. 3.4(a) illustrates this procedure.

The same method is followed when the first column stack is not the reference stack, except, that now the missing joint coordinates to the left of the reference stack are to be computed using subroutine STACLE5 instead of subroutine STACRI5. Fig 3.4(b) illustrates this case.

If the origin selected for the computation of coordinates was not at the intersection of the first column stack with the reference level because of dummy columns in that stack, the origin is now repositioned so that it is at the intersection of the first column stack with the reference level. The coordinates values are modified so that they refer to the repositioned origin. The coordinates information is stored in a disk file.

The above procedure is valid for all X and Y frames.

FIGURE 3.4 Computation of 2-D Coordinates

- Note: The arrows and numbers represent the consecutive steps in computing the 2-D joint coordinates.

3.3 Computation of Three Dimensional Joint Coordinates for Orthogonal Structures.

The computation of three dimensional joint coordinates begins at the reference level, using X, Y, Z axes. The origin is set at the first column stack of the first X frame encountered in the AMECO file (reference frame).

The three dimensional joint coordinates are computed as follows:

- 1) Set the global X coordinates of each column stack of the frame as equal to the corresponding local X coordinates of the joints computed in Section 3.2.
- 2) Since the structure is orthogonal, the global Y coordinates of each column of the reference frame is set equal to 0.
- 3) Search for intersecting Y frames.
- 4) Set the global X coordinate of each column stack of the Y frame equal to the global X coordinate of the intersecting column.
- 5) Set the global Y coordinate of each column of the Y frame equal to the corresponding local X joint coordinate.
The steps 3 to 5 are repeated till all the Y frames that intersect the reference frame have been processed.
- 6) Find Y frames that do not intersect the reference frame.
- 7) Find a X frame that intersects with the Y frame under

consideration.

In this new X frame :

- 8) Find the first column with unknown global Y coordinate and the column closest to it with known global X/Y coordinates.
- 9) Compute the unknown global X coordinate, based on the length of the beam(s) joining the column of known coordinates and the column of unknown coordinates.
- 10) Set the global Y coordinate equal to the global Y coordinate of the column of known coordinates.

This process is repeated till all the column coordinates are determined. Fig.3.5 illustrates the above procedure.

In that figure the coordinates of frames 1-2-3 and intersecting frames 1-6-11-16 , 2-7-12-17 and 3-8-13 are determined first. Frames 4-9-14 and 5-10-15 do not intersect frame 1-2-3. The new X frame which does intersect them is 4-5-6-7-8. The first column with unknown Y coordinates is in the intersecting frame 4-9-14 and is column 4. The column closest to it with known coordinates is column 6. The coordinates of columns 4 and 5 are determined by knowing the coordinates of column 6 and the lengths of beams 4-5 and 5-6. After this coordinates along 4-9-14 and 5-10-15 can be determined.

This process produces X/Y coordinates of all column stacks. These coordinates can be used for a graphic display of any of the floors.

FIGURE 3.5 Column stack coordinates for an orthogonal structure.

Note: The arrows and corresponding numbers indicate the consecutive steps to find the column stack coordinates.

3.4 Computation of Three Dimensional Coordinates for Non-orthogonal Structures.

The procedure for determining X, Y and Z coordinates for non-orthogonal structures is in principle identical to the procedure for orthogonal structures.

In the non-orthogonal structures the beams and thus the vertical frames are no longer parallel to the global X/Y axis. The angle of deviation is input by the user and ranges anywhere from 0° to 90° . As in the other cases the structure is again composed of X/Y frames but these frames are no longer parallel to either axis. Therefore in computing the coordinates, use is made of the directional sines and cosines of the beams. In every other respect, the computational procedure is identical to that for orthogonal structures.

Secondary member (beam) coordinates are computed at the time of plotting, and this is explained in Section 3.8.2.

3.5 Generation of Building Grid Lines

Two types of grid line are defined at every floor of the building:

- a) X-grid lines parallel to the X-axis.
- b) Y-grid lines parallel to the Y-axis.

The X and Y grid line coordinates are generated at the reference level and apply to all levels.

Subroutine GRID5 determines the location of the grid lines. X-grid lines are located by looking up the Y-coordinates of the first column of each X-frame, similarly Y-grid lines are located by looking up the X-coordinate of the last column of each Y-frame.

These values are saved on a disk file for later plotting of frame elevations and floor plans.

3.6 Drawing Size, Border and Title Block

Five drawing sizes can be requested (A, B, C, D, E); the default is "D" corresponding to a 34"x 22" drawing.

3.6.1 The border is plotted by subroutine BORDER and consists of:

- a) Double solid frame.
- b) A cutting dashed line around the frame.

An example of the border lines is shown in Fig. 3.7.

FIGURE 3.7 Standard Border for a Drawing

3.6.2 Title Block

The title block is plotted by subroutine LOG05, and the text written in it is controlled by AMECO "JOB TITLE" statements input by the user. Fig. 3.8 illustrates a typical title block, reduced about 30%.

Information under the headings 'CONSULTING ENGINEER', 'CLIENT', 'STRUCTURE' and 'PROJECT' text is plotted using the largest character size possible but no larger than .2". For long texts the character size is decreased.

The user may split the above texts in two lines by inserting a slash "/" at the desired location during data input phase.

'JOB' number and 'REVISION' number (input by the user), and 'DATE' (determined by AMECO) are also written in the title block using .125" character size.

Framing views title is written using .15" character size. Boxes are drawn for manually entering draftman's, checker's and engineer's names, and dates.

Space is also provided for professional engineer's seal.

CONSULTING ENGINEER ►	CARLETON UNIVERSITY CIVIL ENGINEERING DEPT.		
CLIENT name ►	BRAMALEA LTD.		
PROJECT name ►	MASTER THESIS		
STRUCTURE ►	3 STOREY STEEL BUILDING		
FRAMING view ►	FRAMING ELEVATIONS		
Engineer's seal ►	SCALE	AS NOTED	DATE
	DRAWN		04/08/15
	CHECKED		
	APPROVED		
	PROJECT NO. 19823	DRAWING NO. SS-1	SHEET NO. A1

FIGURE 3.8 Title Block of a Drawing

3.7 Location of views on drawings

Two type of views can be drawn, framing elevations and framing plans.

Based on manual drafting practice, margins have been preset as shown in Fig. 3.9.

Subroutine FRAME computes the space required for each view by considering the actual size of the framing elevation or floor plans, the extension lines, dimension lines, grid lines with their circles and margins.

Subroutine FRAME attempts to place as many views (elevation/plans) on a single drawing as possible. An example is shown in Fig.3.10. When a drawing is filled and more views are to be plotted, a new drawing is initiated.

If a single view does not fit on the drawing (based on user supplied scale and drawing size) a message is printed and processing is terminated.

Subroutine FRAME also sets the new origin for the next view to be plotted.

FIGURE 3.9 Standard Margins

FIGURE 3.10 Multiple Views on a Drawing

3.8 Plotting of Views

There are two types of views that can be requested:

Framing elevations

Floor plans

All members are plotted using .02 mm thick lines. The building grid and dimension lines are .01 mm thick. For a drawing made in Imperial units, all distances are written in feet, inches and fractions of inches (e.g. 3/8, 1/16). Member sizes are written in inches.

For drawings made in SI or Metric units, all distances and member sizes are written in mm.

3.8.1 Framing Elevations

As explained earlier in Section 3.7, subroutine FRAME locates the elevation to be plotted on the drawing and sets its origin. In the plotting of the elevations, the joint coordinates determined by the procedure described in Section 3.2 are used.

The elevations are plotted in the following sequence:

1) Columns

The column are represented as follows:

Real columns are shown by a solid line.

Dummy columns when plotted are shown by a single dashed line.

Each column stack is drawn from the top level downwards.

Each column is drawn from its top joint downwards using its length supplied in the input.

Each column bottom end is represented by a short horizontal line. If the length of the column has been wrongly defined during the input phase the error will appear on the plot, otherwise the short line will be masked by the connecting beam.

2) Beam and Slab Representation

In general, each beam is drawn from its left joint towards the right, using its length supplied in the input.

If the frame is not parallel to the X or Y axis, or if some beams of the frame are not parallel to the X/Y axis, the length is modified by the directional cosine of the beam.

If the beam is too short, a gap will appear at the right end of the beam indicating the error.

Real beams and slabs are represented by a single solid line.

Dummy beams, if they are to be drawn, are represented by a single dashed line.

Different conventions are used in drawing the beam depending on the type of connection. If the beam is continuous it is drawn to its full length. If the beam is pin connected a gap of 2 mm is left at the ends of the beam.

These conventions are shown in Fig. 3.12.

FRAMING ELEVATION LINE ©

SCALE 1 = 20

FIGURE 3.12 Conventions used in drawing the beams

3) Vertical Bracing System and Walls

Five types of bracing are available in AMECO : right bracing, left bracing, X-bracing, A-bracing and V-bracings. These are illustrated in Fig. 3.13(a).

The bracing end coordinates are computed as follows:

Subroutine PLOT5 finds the joint numbers to which the bracing ends are attached, by using AMECO functions MBRDN52, JTLE52 and JTRI52. Based on this information the joint coordinates are obtained and the bracing members are drawn.

The bracing members are represented by a solid line, without any gap at the ends. Walls are represented as X-bracing and drawn using dashed lines.

Fig. 3.13(b) illustrates the display of steel V-bracing and concrete shear walls, the latter represented by dashed X-braces.

(a)

(b)

FIGURE 3.13 Vertical Bracing System and Walls

4) Beam Connections, Marks and Sizes

(a) Beam connections

Rigid connections.

Steel- A rigid connection is represented by a solid triangle and is drawn by subroutine SYBM5.

Concrete- no symbol is drawn.

Pinned connections.

In both steel and concrete a gap of 2 mm is left at the member ends.

(b) Beam, slab and wall marks.

Subroutine LABEL5 writes the member mark under the beam/slab/wall line using a character size of .125".

The beam, slab and wall marks are supplied by the user while generating the input data for AMECO.

(c) Beam sizes. Subroutine SIZE5 writes the corresponding beam size (profile name) above the beam, also using a character size of .125". It should be noted that the beam sizes on framing elevations are shown only for steel structures.

FIGURE 3.14 Column sizes, splice points and Column marks

5) Column sizes, splice points and column marks.

For writing column marks and sizes a character size of .094" is used.

(a) Column sizes and splice points.

For steel drawings, subroutine SIZE5 draws the splice points as a short horizontal line across the corresponding column at the locations requested by the user. An example is shown in Fig. 3.14.

The column sizes (profile names) are also written by subroutine SIZE5. The size information is placed to the left of the column near a splice point.

It should, however, be noted that column sizes are shown only for steel structures.

(b) Column marks.

Subroutine LABEL5 writes the column marks under the column stack.

A column mark consists of the letter C followed by the column number. The column marks are as supplied by the user while preparing the input data for AMECO.

6) Dimension Lines and Building Grid lines.

(a) Dimension Lines.

Three different types of dimension lines are drawn on the elevations by subroutine DIM5.

A vertical dimensioning line is drawn to the left of the first column stack to indicate the interstorey heights.

In addition, two horizontal lines are shown, one to indicate the overall width of the elevation, and the other, the distances between the grid lines. An example appears in Fig. 3.16.

The dimensions are written using a character size of .125". If the space available for the dimensions is not adequate, the character size is decreased.

(b) Building Grid Lines.

The positioning of the grid lines has already been discussed in Section 3.5. Following the convention described above, subroutine LINE5 draws a grid line towards the top of each column stack, as an extension of the column line.

A grid line is represented by a dashed dotted line that ends in a circle of .5" diameter, with a label inside it, as shown in Fig. 3.16.

FRAMING ELEVATION LINE (D)
SCALE 1" = 20'

FIGURE 3.16 Framing Elevations

Letters A to Z are used to label grid lines parallel to the global Y-axis.

Numbers 1 to 99 are used to label grid lines parallel to the global X-axis.

7) Elevation Titles.

A title is written below each framing elevation, identifying it by the grid line label (placed in a circle), as shown in Fig. 3.16.

The scale is written below the elevation title.

3.8.2 Floor Framing Plans.

The floor plans are plotted in the following sequence :

1. Columns.
2. Beams, walls and slabs.
3. Column marks.
4. Main beam marks and end connections.
5. Secondary beam marks.
6. Main beam/wall/slab sizes.
7. Secondary beam sizes.
8. Floor plan titles.

For clarity, the description of the floor plans are presented in two parts; first for Steel structures and then for Concrete structures.

3.8.3 Floor Plan for Steel Structures.

1) Columns.

The column representation is worked out by subroutine SYMB5.

Columns are drawn as 'I' shapes, 50% larger than their actual dimensions. The web is drawn parallel to either X or Y-global axis or at an oblique angle, depending on the information supplied during input.

2) Main Beams, Secondary Beams and Walls.

A continuous beam is represented by a single solid line drawn to the full length of the beam.

A simply supported beam is represented by a single solid line drawn 2 mm shorter at the pinned ends. All secondary beams are assumed to be simply supported and are drawn accordingly. An example is shown in Fig. 3.17.

Each main or secondary beam can support a maximum of 8 secondary beams.

Subroutine SECBM computes the secondary beam coordinates as follows:

- The subroutine first identifies the secondary beam supported by X main beams using function JLBMS.
- It then uses function FXBM5 to obtain from the AMECO data base the length of the secondary beam and the distance of its left end from either the left end of the supporting beam or from the immediately preceding secondary beam should there be one.

The information generated by SECBM is used for plotting the secondary beams. A similar procedure is followed in plotting secondary beams supported by Y main beams.

Once all the secondary beams supported by the X/Y main beams have been determined a search is made for secondary beams supported by other secondary beams. The steps described in the foregoing paragraphs are then followed to determine the coordinates of such secondary beams.

FIGURE 3.17 Steel Building Plan

In order to avoid interferences between marks/sizes of supporting beams and the secondary beams framing into such beam, the secondary beams may be drawn shorter at one or both ends. An example is shown in Fig. 3.18.

Walls are represented by two solid lines.

3) Column Marks.

In order to avoid interferences between column marks and the marks/sizes of adjacent beams, a search is made for the best position of the column mark. This is done in subroutine POSIT, there are four possible locations for each column mark, as shown in Fig. 3.19.

Position 1 is the one normally selected if there is no interference. Position 2 to 4 are alternative positions, if none of them satisfy the requirement for clarity, the character size is decreased. Finally if no place can be found for the column mark an asterisk is plotted. Some examples of column marks can be seen in Fig. 3.20.

4) Main Beam Marks and End Connections.

Main beam and wall marks are written using a character size of .125". Under special circumstances, i.e. for short spans, the character size is decreased. If marks become unreadable they are replaced by an asterisk.

FIGURE 3.18 Floor Plan for Steel Building showing Secondary Beams

The main beam and wall marks are written out by subroutine LABEL5 and are located under the beam line parallel to it, as shown in Fig. 3.20.

A rigid connection is indicated by a solid triangle at the rigidly connected end. The symbol is drawn by subroutine SYMB5.

A pinned connection is shown by a gap of 2 mm between the connected members.

5) Secondary Beam Marks.

Secondary beam marks are written by subroutine SECBM, they are located under the beam and parallel to it.

To distinguish between a main and a secondary beam, the secondary beam mark is written in a character size of .094", or even smaller for short spans.

6) Main Beam/Wall sizes.

Beam sizes are written by subroutine SIZE5 just above the beam and parallel to it.

A character size of .125" is used, and it may be decreased for short spans or even replaced by an asterisk if the character size becomes too small.

Wall thicknesses are not written.

FIGURE 3.19 Positions for placing Column Marks

FIGURE 3.20 Floor Plan for a Steel Building showing Column marks and Beam Marks

7) Secondary Beam sizes.

Secondary beam sizes are written by subroutine SIZE5 and are located above the beam line and parallel to it.

Character sizes used are similar to those used for secondary beam marks.

8) Dimension and Grid Lines.

(a) Dimension Lines.

Subroutine DIM5 draws the dimension lines on floor plans. As shown in Fig. 3.21, three different types of dimension lines are drawn. These are:

1. Dimension line to indicate the overall floor width in the Y-direction. This line is drawn parallel to the global Y-axis and at 1.25" from the first Y-frame.
2. Dimension line to indicate the distances between X grid lines. This line is drawn parallel to the Y axis at .875" from the first Y-frame.
3. Dimension line to indicate the distances between the secondary beams. This line is drawn at .5" from the first Y-frame.

At this time the program draws dimension lines for only those secondary beams that frame into perimeter beams which are close to the dimension lines referred

to in 1 and 2 above.

A similar set of lines are drawn in the X direction.

The dimensions are written using a character size of .125"; the character size is decreased if the available space for the dimension is too small.

(b) Grid Lines.

The position of the grid lines have been discussed in Section 3.5. Following the convention described there, subroutine LINE5 draws grid lines on the plans.

Each grid line connects a series of columns and is represented by a dashed dotted line that ends in a circle of .5" diam. with a label inside it.

A letter from A to Z identifies the Y grid lines and a number from 1 to 99 identifies the X grid line- (Fig. 3.21).

9) Floor Plan Titles.

A title is written below each floor plan using a character size of .2". The floor plan is identified by its level number. The scale is written below the floor plan title- (Fig. 3.21).

FRAMING PLAN LEVEL 2
SCALE 1" = 20'

FIGURE 3.21 Dimension Lines, Grid lines and Floor Plan Titles

3.8.4 Floor Plans for Concrete Structures.

The plotting of the floor plan is executed in a sequence similar to that used in plotting the floor plan for steel structures and in a very similar manner.

1) Column Representation.

The columns on the floor plan of a concrete structure are drawn by subroutine SYMB5. Columns are represented by their real shape (square, rectangle or round). The column faces can be oriented parallel to either X or Y axis or at an oblique angle.

2) Beams, Walls and Slabs.

In general, beams are represented by their real width and length and are drawn using double dashed lines.

Secondary beam coordinates are computed in a manner similar to that explained in Section 3.8.2(2). On the drawings the secondary beams are represented in the same manner as the main beams.

If interference occurs between marks/sizes of supporting beams and secondary beams, the latter are drawn shorter than their real length. Example can be seen in Fig. 3.22.

Walls are represented by double solid lines

Slabs are represented by solid lines with half arrow head at each end, specific representation of the slabs is

FRAMING PLAN LEVEL 2
SCALE 1/20

FIGURE 3.22 Floor Plan of a Concrete Structure

however shown on the drawings only if the marks of the beams/slabs/walls have been requested by the user during the input phase. Examples are shown in Fig. 3.23.

3) Column Marks.

Subroutine POSIT locates the column marks as described in Section 3.8.2(3).

4) Main Beam Marks and End Connections.

The main beam marks are written in the same manner as for steel beams. Details are described in Section 3.8.2(4).

A rigid beam-column connection is represented by joining the column and the beam together.

A gap of 2 mm is left at a pinned end of the beam.

5) Secondary Beam Marks.

The details are similar to those described in Section 3.8.2(5) for steel structures.

6) Main Beam/Wall/Slab Sizes.

Main beam sizes are written by subroutine SIZE5; they are located above the beam parallel to it. A beam size shows the width and depth of the beam - in inches for drawings made in Imperial units, and in mm for drawings made in SI or Metric units.

FRAMING PLAN LEVEL 3

FIGURE 3.23 Floor Plan of a Concrete Structure showing the Representation of Slabs

A character size of .125 " is used.

Wall and slab sizes are omitted.

7) Secondary Beam Sizes.

The secondary beam sizes are written in the same manner as main beam sizes, except that a character size of .094" is used.

8) Dimensions and Building Grid Lines.

These are drawn in a manner similar to that for steel structures in Section 3.8.2(8). Examples can be seen in Fig. 3.22.

9) Floor Plan Titles.

The details are similar to those described for a steel structure in Section 3.8.2(9).

CHAPTER 4

EXECUTION OF THE GRAPHIC PROGRAM

The graphic module, integrated in the AMECO - 17 system, is operational on Control Data Corp. CYBER Series computers, under the NOS operating system.

AMECO-17 is executed in batch mode. Therefore, as a first step, an input file is created (or an existing file retrieved). This file consists of AMECO commands describing the structure, the loads acting on it, the design criteria, etc. - all in accordance with the AMECO User's Manuals(8),(9). The file is then supplemented with the desired PLOT commands. The sentence structure and the application of the PLOT commands are described in the following sections of this chapter.

Together with appropriate NOS Control Statements, that load AMECO, the input file is then submitted to the CYBER machine for processing.

Besides the files that the design modules of AMECO may create, the PLOT commands will generate permanent "Plot Files". These files can be displayed on a variety of graphic CRT terminals or plotted directly on a hard copy plotter. In addition, "Plot Files" can be transmitted to an Inter-

active Graphic System for re-formatting and further graphics processing.

This chapter presents the rules for the PLOT commands and describes the various optional procedures available for obtaining graphic displays.

4.1 Command Description

General.

- The commands are formed according to the AMECO language conventions. These conventions include the following definitions.

Statements are " All instructions and data entered into the computer for AMECO processing in the form of sentences and in free format."

Command is " a statement starting with a verb".

Labelled data is " data entered in any order, in a command with appropriate labels".

Plot commands direct the program to produce a plot file of a specific view. Labelled data defines the specific characteristic required in the view.

The Plot commands have the same structure as the other AMECO commands, therefore they follow the same rules. These are :

- Default values have been preset for most of the labelled data; they are indicated by arrows in the facsimile of the command.
- All words can be abbreviated to the first four characters.
- When shorter abbreviations are allowed, the required

characters are underlined in the facsimile.(Fig. 4.1).

- Words with horizontal bars above can be omitted without changing the meaning of the statement.
- Prepositions : 'AT', 'FROM', 'AND', 'WITH', 'ON', 'FOR', are ignored by the lexical analyser, they can be used to construct more legible statements.
- A maximum of 80 characters can be entered per line.
- A hyphen may be used at the end of a line to continue a long Plot Command on the next line. A blank must precede the hyphen.

The command structure is shown by the command facsimile in Fig. 4.1. The command begins with a verb and is followed by one or more modifiers and then by a series of labelled data. More than one command modifier can be included with one command. Labelled data should follow the command modifier. A new command modifier thus terminates the list of labelled data. All components of the command are separated from each other by either one or more spaces or a comma. Only one of the items enclosed in a pair of braces, on the facsimile of the command should be selected at one time.

The Plot Commands are analysed by subroutine C1PLOT. Subroutine C1PLOT reads the plot commands, checks for illegal entries, changes default values to the entered

values and stores flags to execute the plotting according to the user requirements. If an illegal entry is found a message is printed indicating the error and processing is terminated in the input phase. ;

PLOT	ALL	(PL1)
	FRAME	(PL2)
	FLOOR ($l_1, l_2, \dots, l_5$ to l_8)	(PL3) (labelled data)
	XFRAME ($fx_1, fx_2, \dots, fx_5$ to fx_8)	(PL4)
	YFRAME ($fy_1, fy_2, \dots, fy_5$ to fy_8)	(PL5)

PLOT LABELLED DATA.

DIMENSION	$\begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD1)
DUMMY	$\begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD2)
SECONDARY BEAM	$\begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD3)
SCALE	1/n	(PLD4)
MARKS	$\left(\begin{Bmatrix} \text{COLUMNS} \\ \text{BEAMS} \end{Bmatrix} \begin{Bmatrix} \text{MAIN} \\ \text{SECOND} \end{Bmatrix} \right) \begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD5)
SIZES	$\left(\begin{Bmatrix} \text{COLUMNS} \\ \text{BEAMS} \end{Bmatrix} \begin{Bmatrix} \text{MAIN} \\ \text{SECOND} \end{Bmatrix} \right) \begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD6)
DRAWING SIZE	$\begin{Bmatrix} A \\ B \\ C \\ D \\ E \end{Bmatrix} \leftarrow$	(PLD7)
CONNECTIONS BEAMS	$\begin{Bmatrix} \text{NO} \\ \text{YES} \end{Bmatrix} \leftarrow$	(PLD8)

FIGURE 4.1 PLOT COMMAND facsimile.

NOTE: Labels PL1 to PL5 and PLD1 to PLD8 are not part of the command, they are included here only for the purpose of references.

4.1.1 PLOT COMMANDS.

PLOT	ALL	(PL1)
	FRAME	(PL2)
	FLOOR ($l_1, l_2, \dots, l_5$ to l_8)	(PL3) (labelled data)
	XFRAME ($fx_1, fx_2, \dots, fx_5$ to fx_8)	(PL4)
	YFRAME ($fy_1, fy_2, \dots, fy_5$ to fy_8)	(PL5)

(PL1) Enter "ALL" to plot all vertical frames and all floor plans of the structure.

Labelled data - enter data as described in Section 4.1.2

e.g. PLOT ALL

e.g. PLOT ALL, MARKS NO

This command will produce drawings without member marks.

(PL2) Enter "FRAME" to plot all the frames in X and Y directions.

Labelled data - enter data as described in Section 4.1.2

e.g. PLOT FRAME

e.g. PLOT FRAME, SIZE OF BEAM YES

This command will produce drawings of X and Y frames with all beam sizes shown on the drawing.

(PL3) Enter "FLOOR" to plot all the floors of the structure.

Enter level numbers $l_1, l_2, \dots, l_3$ to l_8 to plot selected floor plans.

Labelled data - enter data as described in Section 4.1.2

e.g. PLOT FLOORS

This command will produce a plot file containing all the floor plans of the structure.

e.g. PLOT FLOOR 1

e.g. PLOT FLOOR 2 TO 4 6 9

The last command will produce a plot file for plotting floor plans of levels 2, 3, 4, 6 and 9.

e.g. PLOT FLOOR 1 MARKS FOR MAIN BEAM NO

This command produces a plot of level 1 without main beam marks.

(PL4)

Enter "XFRAME" to plot all the X-direction frames of the structure.

Enter frame ID's fx_1 , fx_2 , ... fx_5 to fx_8 to plot selected frames.

Labelled data - enter data as described in Section 4.1.2.

CAUTION: Enter XFRAME as one word, otherwise a syntax error will occur.

e.g. PLOT XFRAME

This command produces a plot file containing all the X direction frames of the structure.

e.g. PLOT XFRAMES 1 6

This command produces a plot file for drawing X-direction frames 1 and 6.

e.g. PLOT XFRAME 1 6, DUMMY NO, SCALE 1/10

This statement produces a plot file for drawing X-direction frames 1 and 6 to scale of 1:10 and with the dummy members omitted.

(PL5) Enter "YFRAME" to ~~plot~~ all the Y-direction frames of the structure.

Enter frame ID's $fy_1, fy_2, \dots, fy_5$ to fy_8 to plot selected Y-frames

Labelled data - enter data as described in Section 4.1.2.

CAUTION : Enter YFRAME as one word, otherwise a syntax error will occur.

Same as command (PL4), but applied to Y-direction frames.

In general, each command (PL1 to PL5) will initiate a new drawing.

4.1.2 Labelled data in Plot Commands

Labelled data are words followed by modifier words or numeric data and can be entered in a random order in the Plot Command. The data items need to be entered only where they differ from the default values.

(PLD1) Enter "NO" when dimensions and grid lines are to be omitted.

e.g. PLOT FLOOR DIME NO, XFRAME 1 TO 4

This statement produces a plot file for all the floor plans of the structure without dimensions and grid lines and also will create another drawing(s) containing X-direction frames 1, 2, 3, 4 with dimensions and grid lines shown.

(PLD2) Enter "NO" when dummy members are to be omitted.

e.g. PLOT XFRAME 3 DUMMY NO

(PLD3) Enter "NO" when secondary beams are not to be plotted in any or all the floor plans.

e.g. PLOT FLOOR SECO BEAMS NO

This statement produces a plot file for drawing all the floor plans without secondary beams.

(PLD4) Enter scale "1/n" to change the default scale.

In the Imperial units scale designation 1/n stands for

1 inch equal to n feet. The default value is 1/20
implying 1 inch equal to 20 feet.

For SI or Metric units the scale designation 1/n
stands for 1 mm equal to n mm, the default value is
1/200.

The slash "/" in the scale designation is optional
and can be replaced by a blank, so then 1 n is the
same as 1/n.

For Imperial units:

e.g. PLOT FLOOR 5, SCALE 1/15

For SI or Metric units:

e.g. PLOT FLOOR 5, SCALE 1 100

The last command produces a plot file for drawing the
floor plan at level 5 at a scale which implies 1 mm on
the drawing represents 100 mm of real dimension.

(PLD5) Enter "NO" to omit all member marks (columns, main and
secondary beams)

e.g. PLOT YFRAME 3, MARKS NO

This command will produce a plot file for drawing the
Y-direction frame 3 without any member marks.

e.g. PLOT YFRAME 3, MARKS FOR BEAM NO

This command will produce a plot file for drawing Y
frame 3 with column marks but with no beam marks.

e.g. PLOT FLOOR 2, MARKS OF BEAM NO

This command will produce a plot file for drawing the floor plan at level 2 without main and secondary beam marks.

(PLD6) Enter "YES" to write the sizes of either all the members or of only the selected type of members.

e.g. PLOT FLOOR 12, SIZE BEAM YES

This command will produce a plot file for floor level 12 on which the beam sizes (main and secondary) will be shown.

(PLD7) Enter drawing size to modify the default size "D" that corresponds to a 34"x 22" drawing.

e.g. PLOT FLOOR 1, DRAW SIZE A

This will produce a plan of floor level 1 on a 11"x 8 1/2" drawing size.

Drawing sizes available are:

	X	Y
A	11	8 1/2
B	17	11
C	22	17
D	34	22
E	44	34

All the sizes are given in inches.

(PLD8) Enter "NO" to omit rigid connection representation for steel drawings.

This instruction applies to steel structures only.

e.g. PLOT FLOOR 6, CONN BEAM NO

This will produce a plot file for the floor plan of level 6 in which the rigid connection symbols (solid triangles) will be omitted from the rigidly connected beam-column joints.

4.2 Purposes of Plotting

Graphic displays can serve three different objectives:

- a) Verification of input data.
- b) Schematic displays of design results.
- c) Production of final drawings.

a) Verification of input data

AMECO-17 is a system that can handle a wide range of structures.

For a successful design the first requirement is that the geometry of the model entered into the computer represent the real structure in every detail.

To achieve this objective it is necessary to carry out a very careful check of the geometry (member dimensions, connectivities, etc.)

depending on whether the grid is an X or Y grid.

- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

b) Concrete structures

- Framing elevation/section drawn to specified scale and units of measurement.
- Column marks.
- Beam marks.
- Level numbers.
- Dimension showing overall width of the elevation/section.
- Dimension indicating interstorey heights.
- Building grid lines identified by letters or numbers depending on whether the grid is an X or a Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

All framing plan and elevation drawings have a border and a title block.

Appendix A illustrates five drawings produced by the graphic module.

execute phase A only and come to a stop thereafter as
illustrated in Fig. 4.2.

The following example illustrates the use of these two
Commands.

```
00100 CARLETON
00110 PROJECT CITY HALL, STE. JOWITE
00120 STRUCTURE 7 STOREY CONCRETE STRUCTURE
00130 CLIENT MARATHON REALTY
00140 INPUT F.V
00150 FILE CITYHALL
00160 $
00170 AWE 0 CONCRETE SI
00180 $
00190 USE CODE NBC-90 CONC 30 COL CONC 35 STEEL 400
00200 PLOT FLOOR 1 TO 3 SCALE 1:150 DIMENSION NO
00210 PLOT XFRAME 1 SCALE 1:250 DIMENSION NO
00220 PLOT YFRAME 1 SCALE 1:250 DIMENSION NO
00230 $
00240 LIST ALL
00250 LIST FRAME FORCES EQ
00260 $
00270 LEVEL 7 TO 3
00280 DESIGN TWO WAY SYSTEM
00290 USE SPAN 9 SLAB 100 WEB 450 400 FDL 1.454 FLL 3.875
00300 S301 TO 404
00310 USE DL W 3.69
00320 B201 TO 204
00330 B501 TO 504
00340 USE SPAN 8
00350 B701 B702 7.5 B703
00360 B1101 B1102 7.5 B1103
00370 USE DL W 0 WALL THICK 300 WEB 700 1200
00380 S801 WALL802 7.5 S803
00390 S901 S902 7.5 S903
00400 S1001 WALL1002 7.5 S1003
00410 DESIGN COLS L 3.65 RECT 900 500
00420 C1 TO 5 201 701
00430 C10 TO 14 301 702
00440 C19 TO 23 401 703
00450 C28 TO 32 501 703R
00460 C11 RECT 300 500
00470 C13 DO
00480 C20 DO
00490 C22 DO
00500 C12 RECT 900 1100
00510 C21 DO
00520 $
00530 LEVEL 2
00540 DESIGN TWO WAY SYSTEM
00550 USE SPAN 9 WEB 450 500 SLAB 100 FDL 1.454 FLL 3.875
00560 B101 TO 104 REQ 3
00570 S301 TO 304
00580 B305 TO 408
00590 USE DL W 3.69
00600 B201 TO 208
00610 B501 TO 508
00620 USE SPAN 8
00630 B701 B702 7.5 B703
00640 USE FDL 4.15 FLL 15.2
00650 B1501 4.104R B1502 3.75 104R 7.5 B1503 4 104R
```

Over

[illegible][illegible]

```

01000 $
01010 DESIGN COLS L 3.65 RECT 400 500
01020 $
01030 01 TO 9 201 701
01040 010 TO 18 301 701
01050 019 TO 78 27 401 703
01060 028 TO 36 501 7036
01070 011 RECT 300 100
01080 013 DO
01090 010 DO
01100 012 DO
01110 011 RECT 400 1100
01120 021 DO
01130 014 TO 17 DO
01140 023 TO 26 DO

```

```

0170 FRAME 1 1 1 1 1 1
0170 EQ FACTOR 1.3
0180 FRAME 1 1 1 1 1 1
0170 EQ FACTOR 1.3
0180 EXECUTE FLOT

```

Schematic Display of Design Results

Generally, for the design of frame structures more than one design cycle is required. Referring to Fig. 4.2, the cycling process in Phase C will be repeated as many times as has been requested by the user or till the changes in member sizes is less than the allowed tolerance.

Phase C provides the user the opportunity to obtain a graphic display of the member sizes for each design cycle.

Two sets of commands are required to obtain a graphic display of the design results:

- 1). One or more Plot Command to specify which are to be displayed. Framing elevations provide the user the column and beam sizes for all or specified frames, while the floor plans provide him the main and secondary beam sizes for all or specified levels.
- 2) The execution command to carry out the plotting cycle by cycle is "EXECUTE ALL, PLOT EACH CYCLE" and like other execution commands it must be entered last.

The command "EXECUTE ALL, PLOT EACH CYCLE", instructs the program to execute Phases "A" to "D", producing a plot file for each design cycle in phase "C".

FIGURE 4.2 AMECO integrated analysis/design process with graphics

Sometimes, the user may desire to review the results of one cycle before proceeding to the next cycle. A set of commands permit the user to do the following.

- 1) Execute one design cycle.
- 2) Save the results.
- 3) Produce the plot file.

The last of these commands has been added as a result of the work described in this thesis. To be able to produce a plot file after a design cycle, the user must input:

- a) Plot commands that will select the views to be plotted, displaying the member sizes most recently selected.
- b) The Execution Command "EXECUTE" which will carry out a design and produce a plot file after completion of a design cycle.

If plotting was not requested during a run but the user desires to have a graphic display of the member sizes, he can retrieve the results, enter the appropriate Plot command and insert "EXECUTE PLOT" command as the last instruction.

c) Production of Final Drawings

At this stage, the objective of plotting is to produce a set of engineering drawings which include framing plans, elevations and sections. Appendix A contains five example drawings.

Each engineering drawing will contain the following information:

(1) Framing-Plans

a) Steel structure floor plans

- Floor plans drawn to specified scale and units of measurement.
- Column marks.
- Main and secondary beam sizes.
- Dimensions showing the overall widths of the floor plan, distances between grid-lines and distances between secondary and main/secondary beams.
- Building grid lines identified by letters in the case of Y-direction grid lines, and by numbers in the case of X direction grid lines.
- Floor plan title and floor level number.
- Scale at which the floor plan was drawn.

b) Concrete structure floor plans.

The concrete floor plans show the same information as the steel plans described above. The member representation is obviously different. Thus :

- beams are drawn with two dashed lines.
- columns are drawn as rectangles, squares or circles.
- flat plate/slab members are not shown.

Also the edge of the floor slab is indicated by drawing a line at the exterior faces of beams/columns on the perimeter of the floor.

(2) Framing Elevations and Sections

a) Steel structures.

- Framing elevations and or sections drawn to the specified scale and units of measurement.
- Column marks.
- Column splice points.
- Column sizes.
- Beam sizes.
- Level numbers
- Dimension showing the overall width of the elevation.
- Dimensions indicating the interstorey heights.
- Building grid lines identified by letters or numbers

depending on whether the grid is an X or Y grid.

- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

b) Concrete structures

- Framing elevation/section drawn to specified scale and units of measurement.
- Column marks.
- Beam marks.
- Level numbers.
- Dimension showing overall width of the elevation/section.
- Dimension indicating interstorey heights.
- Building grid lines identified by letters or numbers depending on whether the grid is an X or a Y grid.
- Elevation/section title identified by grid line letter or number.
- Scale at which the elevation/section was drawn.

All framing plan and elevation drawings have a border and a title block.

Appendix A illustrates five drawings produced by the graphic module.

CHAPTER 5

DISCUSSIONS

5.1 General

The principal objective of this project is to implement a graphic module for AMECO-17 system, that would produce automatically structural framing drawings.

This Chapter deals with some of the programming problems encountered and identifies the deficiencies and limitations of the graphic module at its present state of development.

This is followed by a description and a qualitative evaluation of five sample drawings presented in Appendix A.

5.2 Interface with AMECO-17 Files

The graphic module has been programmed to produce drawings totally within the system - all input data to the graphic must come from the files of the "parent" AMECO-17 System.

The structure of AMECO files was set up years ago to accomodate its joint relaxation process, for structural analysis, with not much thought given to future plotting needs. One striking example of this is the fact that

AMECO has no joint coordinate tables - it never computes the joint coordinates.

Further more, its member connectivity tables are quite different from those in conventional structural analysis programs. For example, an AMECO file for a typical beam states that at given distances along the span, certain tributary beams frame into it.

As an example of the problem in plotting that arise because of the conventions followed by AMECO in defining connectivity, consider the frames shown in Fig. 5.1. To draw B301 of Fig. 5.1(a), a search has to be made through all the beam files to find which beam "supports" the left end of B301. Once, the supporting beam B101 has been found then knowing its end coordinates and the direction cosine and the distance to B301 from the end of B101, the left end coordinate of B301 can be established.

In a similar manner, the right end coordinates of B301 must be found. It is only after these coordinates have been established that the beam can be drawn.

For conventional programs, the plotting procedure is far more simpler. Referring to Fig. 5.1(b), the member incidence table should state that member 7 goes from joint 4 to 9; joint coordinate table would provide the location of joints 4 and 9. Based on this information member 7 can be easily drawn from joints 4 to 9, without any searches or computations.

FIGURE 5.1 Member Connectivity Convention

The efficiency of AMECO is largely attributed to the fact that structural analysis is performed totally in core (e.g. 2,700 joints solved in 130,000 word memory). To achieve this, in many AMECO files, more than one datum is stored in each word. For example, a 3 bit field is allocated for storing member cross-section index, e.g. 1 for T-beam, 2 for slab, etc. Thus indices for 20 members can be stored in one 60 bit word.

Because of the presence of these specialized storage algorithms, considerable effort had to be spent to become familiar with the structure of data files. Further, to access the data contained in the files, extensive use had to be made of unpacking routines.

5.3 Joint Coordinates

The computation of the joint coordinates forms the starting point in the graphic module.

The computation of the two-dimensional and three-dimensional joint coordinates has been described in detail in Sections 3.2 and 3.3. These computations were based on three assumptions:

- a) Columns are vertical elements joined together by horizontal members.
- b) All columns are defined on at least one level.
- c) Any three-dimensional structure is composed of a series of intersecting frames.

The example shown in Fig. 5.2 illustrates step by step the method used in computing the column stack coordinates. It is assumed that local frame coordinates have been computed earlier.

The example shown in Fig. 5.3 explains in detail a case where all the two dimensional coordinates have been computed, but some of the three-dimensional joint coordinates cannot be computed and therefore the floor plan cannot be plotted.

FIGURE 5.2 Calculation of Column Stack Coordinates.

- 1) Set frame 1-2-3 as reference frame.
- 2) Set X-coordinates of column stacks 1, 2 and 3 equal to the corresponding X joint coordinates computed in the local X-Z axes of the reference frame.
Set Y-coordinates of column stacks 1, 2 and 3 equal to 0.
This is valid because the structure is orthogonal.
- 3) Compute coordinates of intersecting frame 1-5-9.
For column stacks 5 and 9 :
Set the Y-coordinates as equal to the local X-coordinates of joints 5 and 9 respectively.
Set the X-coordinates as equal to the X-coordinate of column stack 1.
- 4) By a similar procedure obtain the coordinates of column stacks 6, 10, 7 and 11.

- 5) Find frame 4-8 that does not intersect frame 1-2-3.
- 6) Find frame 4-5-6-7 that does intersect frame 4-8.
- 7) Repeat step 3 to compute the coordinates of column stacks 4 and 8.

FIGURE 5.3 A Case where Stack Coordinates cannot be Computed
 In this case, the floor plan is formed by X-frames 1-2-3 and 4-5-6-7, 8-9-10-11 and Y-frames 4-8, 5-9, 6-10 and 7-11. According to AMECO convention there is no Y-frame connecting column 1 and points A and B. Similarly there is no frame connecting column 2 and points C and D; also no frame connects column 3 and points E and F.

Even though, all the AMECO frames have been input and the two-dimensional coordinates computed, the structure is indetermined from the point of view of plotting, because

2

there are no connections between frame 1-2-3 and either of frames 4-5-6-7 and 8-9-10-11.

The elevations can be plotted, but not the floor plan.

This situation can be remedied by entering dummy columns at points A to F.

5.4 Location of Views on Drawings

A drawing may display one or more views - plans or elevations. This is a function of the size of the drawing, scale and size of the individual views.

Referring to Fig. 5.4, the following steps are involved in positioning the views on the drawing :

- 1) Set origin at 'A' and draw drawing border and title block.
- 2) Determine the size of view I.
- 3) Check if the view fits on the drawing.
- 4) Position the view on the drawing and reset origin at 'B'.
- 5) Draw view I.
- 6) Determine the size of view II, considering that X-direction grid lines must match up in views I and II.
- 7) Repeat steps 3 to 5.
- 8) Determine size of view III.
- 9) Check if view III fits next to view II, otherwise attempt to place it below view I.
- 10) Reset origin at 'D' and draw view III.
- 11) Repeat procedure until the drawing is full, then start a new drawing.

The determination of the size of the view is itself a rather complex operation. In this determination one has to consider:

- a) the actual size of the framing; b) the length of the grid lines with their circles; c) the length of the extension lines;
- d) the requirement that X-direction grid lines must line up from one adjacent view to the next; the width of margins; f) the title of the view.

FIGURE 5.4 Location of Views on Drawings.

5.5 Plotting Secondary Members

The end coordinates of the secondary beams are computed at the time a plot file is being created.

The following steps are involved in plotting:

- 1) Determine the main/secondary beam that supports the left end of the secondary beam.
- 2) Determine the length of the supported secondary beam.
- 3) If supporting beam marks have been requested :
 - a) Check if interferences will occur between the secondary beam and the lettering for the supporting beam mark.
 - b) If interferences occurs, it is necessary to shorten the length of the supported beam at its left end.
- 4) Determine the main/secondary beam that supports the right end of the secondary beam.
- 5) If supporting beam sizes have been requested :
 - a) Check if interferences will occur between the secondary beam and the lettering for the supporting beam size.
 - b) If interferences occurs, it is necessary to shorten the length of the secondary beam at its right end.

Example of secondary beams are presented in Fig. 5.7

An AMECO floor framing plan may have a number of beams with the same mark, as shown in Fig. 5.8. Consequently it becomes a rather involved logistics problem to plot the beams with

identical marks and to make sure that indeed all of them have been plotted.

FIGURE 5.7 Example of Nested Secondary Beams

FIGURE 5.8 Example of Identical Beam lines

5.6 Column Marks

One of the most important requirement in a drawing is clarity. Thus interference of lettering and lines should be avoided as far as possible.

Sometime the space available to plot the column mark and the beam mark/size of adjacent member is not large enough and interferences may occur.

A very involved procedure was developed to avoid this interferences. The procedure works as follows :

Any column mark in a floor plan can have 4 different positions, as shown in Fig. 5.9.

FIGURE 5.9 Position of Column Marks

The principles involved in the calculation of the coordinates of the lettering block containing the column mark are the same irrespective of the position selected. This description of the method is for position 1, and is based on the assumption that the user has instructed the program to plot the main beam marks/sizes as well as the column marks.

Near position 1, the column has attached to it beams A and B.

The coordinates of the column mark are calculated taking into account:

- a) Angle of deviation of the column from the X-axis.
- b) Angle of deviation of beam A from the X-axis

Once the position of the column mark has been defined, it is checked against the position of the mark of beams A and B.

If interferences are encountered, the mark is moved to position 2, 3, 4 and a similar procedure is followed for determining the coordinates of the mark in its new position.

5.7 Building Grid Lines

There are two types of grid lines:

- a) X grid lines, they are parallel to the X-direction and are drawn near the first column stack of each X frame.
- b) Y grid lines, they are parallel to the Y-direction and are drawn near the last column stack of each Y frame.

The position of each X/Y grid line is set by the position of the first/last column stack of the corresponding frame.

According to this definition, any column out of line with the first/last column stack is drawn eccentric to the grid line.

Fig. 5.10 illustrates this point - the columns along grid lines 1,2 and 3 are not in line.

For non-orthogonal structures, not all the grid lines will be properly defined. Therefore it is advisable, in the plot command to suppress the grid line generation.

Fig. 5.11 illustrates a case where the grid lines cannot be drawn properly.

28

FRAMING PLAN LEVEL 3
SCALE 1" = 10'

FIGURE 5.10 Convention used in Drawing Grid Lines

FRAMING PLAN LEVEL 1
SCALE 1"=20'

FIGURE 5.12 A Complex Non-orthogonal Structure

5.8 Evaluation of Sample Drawings

Appendix 'A' contains five sample drawings, all of size 'D' photo-reduced to 17"x 11" sheets. These were plotted using ink pens on Calcomp model 1039 drum plotter.

Drawing 1 and 2 are typical of 12 structural steel drawings for a 14 storey steel structure built in Culver City, California. The drawings were created by retrieving the actual design result files created in 1982.

Drawing 1 is representative of a steel framing of average complexity encountered in practice. This drawing shows :

- a) all structural members and their sizes.
- b) dimensions locating 95% of all the members.
- c) building grid lines.
- d) beam/column moment connections, identified by solid triangles.
- e) hinged connection represented by 2 mm gap at member end.
- f) column cross-section, drawn to 50% larger scale.

These drawings do not show:

- a) dimensional location of secondary beam in mid-part of the plan for level 4.
- b) location of floor openings.

- c) floor projection beyond column lines.
- d) number of studs for composite beams.

Overall, the drawing is about 95% complete.

Drawing 2 is representative of steel building elevation drawings, showing moment resistant frames.

The drawing shows:

- a) all beam and column sizes.
- b) column splice points identified by horizontal bars.
- c) simple (web) connections identified by 2 mm gap at beam ends.
- d) moment connections identified by solid triangles.
- e) dimensions between grid lines and floor levels.
- f) beam and column marks for cross-referencing to design documents.

It could be advisable to have the column web orientation shown at the bottom of the column stacks, although this is already shown on floor plans.

The drawing is 100% complete.

Drawings 3 and 4 show a 7 storey concrete building composed of one-way slabs, flat slabs, beams, columns and shear walls.

Drawing 3 is representative of a typical, regular layout of floor framing.

The drawing shows :

- a) all beams (drawn to scale) with their sizes indicated.
- b) columns (drawn to scale), but with sizes not indicated.
- c) walls (drawn to scale), but with thicknesses not indicated.
- d) slab extended on the perimeter to the face of the columns, but with projection dimensions not indicated.
- e) building grid lines and distances between grid lines.

If desired the beam/slab marks could have been displayed on the drawing for cross-referencing with reinforcing schedules prepared by AMECO.

The drawing lacks the following information:

- a) floor openings.
- b) wall thicknesses.
- c) slab thicknesses.
- d) floor projection dimensions.

The drawing is about 85% complete.

Drawing 4 shows a longitudinal and a transverse framing elevation of the building.

In engineering practice framing elevation drawings for concrete buildings are normally not made. Floor plans, reinforcing schedules with member sizes and detail drawings are normally sufficient to fully describe the structure. However, such a drawing could be used to supplement the design calculations.

Drawing 5 illustrates a typical flat plate floor plan having the interior columns offset from the building grid lines. The drawing shows:

- a) all beams (drawn to scale) with their sizes indicated.
- b) columns (drawn to scale), but with sizes not indicated.
- c) slab extended on the perimeter to the face of the columns, but with projection dimensions not indicated.
- d) walls (drawn to scale), but thicknesses not indicated.
- e) building grid lines and distances between grid lines.

The drawing lacks the following information:

- a) floor openings.
- b) wall sizes.
- c) offsets of interior columns from grid lines.
- d) reinforcing steel band lines (column/middle strips).

SUMMARY AND CONCLUSIONS

A graphic module has been developed and integrated into an existing system of structural engineering programs, known by the acronym AMECO-17. As programmed, the graphic module can produce structural steel and concrete framing plans and elevations to serve three different objectives:

1.- Verification of input data.

The graphic displays are powerful tools for debugging the input data, specially for large structures. These schematic drawings can be consider to be fully finished for both, steel and concrete structures.

2.- Schematic displays of design results.

The user can obtain a graphic display of the member sizes for each design cycle.

For steel structures the drawings can be considered as almost complete. In the case of concrete structures the completeness is less since no member sizes are shown in the framing elevations.

3.- Production of final drawings.

The drawings are completely generated within the system, requiring no operator input. The degree of completion varies and it is higher for steel structures than for concrete structures.

The graphic module provides with the opportunity for direct transfer of data from the program to the drawing sheet. Even though the degree of completion varies, they provide with an important amount of information that avoid the need to do a lot of tedious work. Furthermore, the drawings can be completed by means of an interactive graphic system.

This graphic module is by no means a finished product and more effort could be invested to implement additional capabilities, such as: cantiliver projections, user selected column marks, floor openings, user specified grid lines, dimensioning of interior secondary beams. For concrete structures, column capitals and drop panels should be shown and reinforcing schedules plotted on the drawings. Also standard concrete details could be shown.

The development of this graphic module was made possible by the existence of the AMECO-17 system. Based on research on literature, AMECO-17 is one of the few analysis/design systems that can produce, for an entire building, the complete design information needed for automated structural drafting.

000000

REFERENCES

1. " STRESS-3.0" User Manual, Multiple Access Division, Canada Systems Group, Don Mills, Ontario, 1976.
2. "STARDYNE" User's Manual, System Development Corp., El Segundo, California, 1979.
3. "ICES STRUDL-II", Engineering User's Manual, Vol. 1, 2, 3, MIT Press, Cambridge, Mass., 1970.
4. "NASTRAN" User's Manual, Vol. II, MacNeal Schwendler Corp., Los Angeles, California, 1982.
5. "MASAP" User Manual, WMJ Engineering Inc., Ottawa, Ont., 1983.
6. "ANSYS" User's Manual, Swanson Analysis Systems Inc., Houston, Pennsylvania, 1982.
7. Wright, W. "Application of Computer Graphics in Civil Engineering", Canadian Society for Civil Engineering, Ottawa, Ontario, 1981.
8. "AMECO-17 Design in Concrete", User Manual, Ameco Computing Systems Company, Toronto, Ontario, 1980.
9. "AMECO-17 Design in Steel", User Manual, Ameco Computing Systems Company, Toronto, Ontario, 1982.
10. Palejs, Andre A. "Practical Approach to Nonlinear Analysis and Construction Simulation for Tall Buildings", Preprints ACI Canadian Chapter, Ottawa, 1976.

APPENDIX A Five Sam

APPENDIX A Five Sample Drawings

FRAMING PLAN LEVEL 14
SCALE 1" = 20'

FRAMING PLAN LEVEL 14

FRAMING PLAN LEVEL 14
SCALE 1/8" = 1'-0"

FRAMING PLAN LEVEL 4
SCALE 1/8" = 1'-0"

NOTES

E.C. ELLIS AND BAMELA LTD.

CORPORATE POINT CULVER CITY

PHASE 1

FRAMING PLANS

DWG. 1

SCALE	AS NOTED	DATE
1/8" = 1'-0"		04/04/00
1/4" = 1'-0"		
1/2" = 1'-0"		
3/4" = 1'-0"		
1" = 1'-0"		
1 1/2" = 1'-0"		
2" = 1'-0"		
3" = 1'-0"		
4" = 1'-0"		
6" = 1'-0"		
8" = 1'-0"		
12" = 1'-0"

FRAMING ELEVATION LINE 7

NOTES

E.C. ELLIS AND BRAMEL, LTD.

CORPORATE POINT CULVER CITY

PHASE 1

FRAMING ELEVATIONS

SCALE	BY	DATE
1/8" = 1'-0"	MS	04/02/13
CHECKED		
APPROVED		

DWG. 2

1182 55-1 2055 4

FRAMING PLAN LEVEL 3
SCALE 1 = 250

FRAMING PLAN LEVEL 1
SCALE 1 = 250

FRAMING PLAN LEVEL 3
SCALE 1 : 250

FRAMING PLAN LEVEL 2
SCALE 1 : 250

FRAMING PLAN LEVEL 1
SCALE 1 : 250

NOTES

CARLETON UNIVERSITY
CIVIL ENGINEERING DEPT.

MARATHON REALTY

CITY HALL, STE. JOVITE

7 STOREY CONCRETE STRUCTURE

FRAMING PLANS

SCALE	AS NOTED	DATE
DRAWN		04/10/00
CHECKED		
APPROVED		
PROJECT NO.	SC-1	

DWG. 3

	A	B	C	D	E	F	G	H	I
LEVELS	7200								
7	201	202	203	204					
6	201	202	203	204					
5	201	202	203	204					
4	201	202	203	204					
3	201	202	203	204					
2	201	202	203	204					
1	201	202	203	204	205	206	207	208	
	201	202	203	204	205	206	207	208	
	C1	C2	C3	C4	C5	C6	C7	C8	C9

FRAMING ELEVATION LINE ④
SCALE 1 = 250

FRAMING ELEVATION LINE ①
SCALE 1 = 250

FRAMING ELEVATION LINE ②
SCALE 1 = 250

NOTES

CARLETON UNIVERSITY
CIVIL ENGINEERING DEPT.

MARATHON REALTY

CITY HALL, STE. JOVITE

7 STORY CONCRETE STRUCTURE

FRAMING ELEVATIONS

DATE	BY	DATE
06/01/14	CHW	
	CHW	
	CHW	
	CHW	

DWG. 4

FRAMING PLAN LEVEL 3
SCALE 1/8" = 1'-0"

NOTES

CARLETON UNIVERSITY
CIVIL ENGINEERING DEPT.

OLYMPIA & YORK

BENOIT ESTATES

8 STOREY FLAT SLAB APART BUILDING

FRAMING PLANS

SCALE	NO. SHEETS	DATE
AS SHOWN	10	04/05/14
CAD		
APP		

DWG. 5

PROJECT NO.	702111	DESIGN NO.	SC-1	DATE	PL-1
-------------	--------	------------	------	------	------

APPENDIX B

Program and Subroutines Description

This Section presents a brief description and flowcharts of the main program and the 30 subroutines of the graphic module. The program and the subroutines together have approximately 6300 Fortran statements. The complete Fortran listing is presented in Appendix C.

The coding is done in ANSI-77 Fortran. Advantage is taken of the CDC Fortran 5 extensions, e.g. direct access I/O, Decode/Encode routines for alphanumeric conversions, etc.

No description is presented for the bit manipulating routines that already existed in AMECO prior to the commencement of this project.

(1) Subroutine C1PLOT

C1PLOT decodes all the Plot commands and saves the requested information on a disk file. It is called by AMECO overlay OVL11 in Phase "A", whenever the word PLOT is encountered in the input stream of commands.

(2) Subroutine PLOT5

PLOT5 computes the two-dimensional joint coordinates, using subroutines : STACDN5, STACUP5, STACLE5, STACRI5.

Also, it finds the joint number at which the bracing ends are attached using AMECO functions : MBRDN52, JTLE52 and JTRI52. PLOT5 is called by AMECO overlay OVL52 in Phase "A" or "C", (Fig. 2.2).

(3) Subroutine STACDN5

STACDN5 finds the coordinates of all joints below the reference level. It uses Function MCOLDN52.

(4) Function MCOLDN52

This AMECO function finds the joint number at the bottom of a column.

(5) Subroutine STACUP5

STACUP5 finds the coordinates in local X-Z axis of all joints above the reference level. It uses Function MCOLU52.

(6) Function MCOLU52

This AMECO function finds the joint number at the top of a column.

(7) Subroutine STACLE5

STACLE5 computes the two-dimensional joint coordinates in local X-Z axis of the joint at the reference stack and at the reference level.

(8) Subroutine STACRI5

Similar to subroutine STACLE5 except that it looks up the joint at the right of the reference stack.

(9) Function MBRDN52

MBRDN52 finds the joint number at the bottom end of the brace.

(10) Function JTLE52

Function JTLE52 finds the joint number at the left end of the beam.

(11) Function JTRI52

Function JTRI52 finds the joint number at the right end of the beam.

(12) Program OVL53

This is the main program of the graphic module its ultimate function is to compute the three-dimensional joint coordinates to produce the plot files for the requested views/drawings.

OVL53 is an overlay and is loaded in memory immediately after OVL52 is executed, either in Phase "A" or Phase "C".

Fig. B.1 shows the flowchart for the plotting of framing elevations.

Fig. B.2 shows the flowchart for the plotting of floor plans.

OVL53 calls subroutines 13 through 31. The latter are described in the remaining part of this appendix.

(13) Subroutine ARROW

ARROW draws the arrows for the dimensioning lines.

(14) Subroutine BORDER

BORDER plots the double border for a drawing.

(15) Subroutine DIM5

DIM5 does the dimensioning of framing elevations and floor plans. Fig. B.3 shows the flowchart for this subroutine.

FIGURE B.1 Flow Chart of the Program Section for Plotting the Framing Elevations.

FIGURE B.1 (cont.)

FIGURE B.1 (cont.)

FIGURE B.1 (cont.)

FIGURE B.2 Flow Chart of the Program Section for Plotting the Floor Plans.

FIGURE B.2 (cont.)

FIGURE B.2 (cont.)

FIGURE B.2 (cont.)

FIGURE B.3 Flow Chart for Subroutine DIM5

(16) Subroutine ENCO

The objective of this subroutine is to encode the beam/column profile names for steel structures and the 'packed' sizes for concrete structures.

(17) Subroutine FRAME

Subroutine FRAME locates each view into the drawing, checks if the drawing size is large enough, starts a new drawing when necessary or sets a flag to stop the process.

(18) Subroutine GRIDL

GRIDL sets the origin of an internal grid line, such as A in Fig. B.4, so as to avoid intersecting the beam on which the grid line terminates (beam B703 in Fig. B.4).

FIGURE B.4 Example of Internal Grid Lines

(19) Subroutine GRID5

GRID5 sets the origin of the X/Y building grid lines.

(20) Subroutine IDIM5

IDIM5 encodes the dimensions in feet, inches and fractions of inches if the drawing is done in Imperial units of measurements. The subroutine also plots the dimensions in Imperial or in SI or Metric.

(21) Function FXBM5

This function computes the distances between secondary beams.

(22) Function JLBM5

JLBM5 looks up the mark of the secondary beam.

(23) Subroutine LABEL5

LABEL5 plots :

- 1) beam and column marks in elevations and in floor plans.
- 2) view titles.

(24) Subroutine LINE5

LINE5 draws the grid lines with the corresponding circles and labels.

(25) Subroutine LOG05

LOG05 draws the complete title block including the corresponding text.

(26) Subroutine POSIT

POSIT positiones the column marks to avoid interferences with the lettering for beam marks/sizes.

(27) Subroutine R53

R53 reads all the floor data ~~from~~ the AMECO files.

(28) Subroutine SECBM

SECBM looks up all the secondary beams, draws them and positions the secondary beam marks.

(29) Subroutine SIZE5

SIZE5 plots all the main/secondary beam sizes and column sizes for steel or concrete structures.

(30) Subroutine UNP53

UNP53 computes the direction sine and cosine of each main beam.

(31) Subroutine SYMB5

SYMB5 draws :

- 1) End connection symbols for steel structures.
- 2) Column representation for steel and concrete structural plans.

APPENDIX C Graphic Program Source Listing

SUBROUTINE CIPLOT(NR)

$$\begin{array}{r} c \\ c \\ \hline c \end{array}$$

```
COMMON MAT, MEM, LEV, ISOLV, LVFIN, LNFIN, LVPAX, LNMAX, NL, IFLEV, IFFRM
COMMON ZZZZ, NME, DAT(3), IDOC(6), IDOS(6), IDAT, ICRIS(4), ICRIC(6)
COMMON JSTOP, NSTOP, IPRINT, ILINK, EC1, EC5, MAXFR, ITEST, IRUN, LVU, LNU
COMMON KAK, LIST(10), IO9, CODE(2), MD(20), UF(20)
COMMON ITC1, NEZ(90)
```

```
COMMON JSF(10),BGN(20),LON(20)
COMMON TAKI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/PACK1/CJ(80),CO(80),KEY(80)
COMMON/PLOTIN/JPLOT(25),IPL0T(25)
COMMON/DYNIN/LYN(10),DYN(20)
```

```
COMMON/EMLN7/IDUM(35)
DIMENSION PL(30),IRIT(140)
DATA PL/3HALL,4HFRAM,4HPEPS,4HTRUS,4HFLOO,4HXFRA,4HYFRA,
4HDI ME,4HDUMM,4HLEVE,3HLEV,4HSECO,4HCONN,4HMARK,
4HSIZE,4HJOIN,4HSCAL,3HDMF,4HDEVI,4HMAIN,4HTEK,
2HMP,4HCALC,4HCOLU,2HBM,3HYES,2HNO,3*0/
```

C
C
C
C
C
C

```

SET IBUG EQ 53 IF DEBUG IS NECESSARY
WORD PLOT WAS READ BY CARD1

```

```
IBUG=LET(11)
```

C MAIN COMMANDS AFTER PLOT

C	COMMAND	SET	TO	AFTER
C	1 ALL	IPLOT(1)	6	SECOND. COMMAND ONLY
C	2 FRAM		1	(3) TO (5), SECON. COMM
C	3 PERS		5	(4) TO (7)
C	4 TRUS		7	(3), (5) TO 7 SECON COMM
C	5 FLOOR		4	3,4,6,7, SECON COMM, LABEL DATA
C	6 XFRAM		2	3 TO 5, 7 SEC., LABEL DATA
C	7 YFRAM		3	3 TO 6 SECON. LABEL DATA

C SECONDARY COMMANDS

C	COMMAND	FOLLOWED BY	DEFAULT	STORED ON
C	*****			
C	DIMENSION		YES=1	IPL0T(8)=0
C	DUMMY		YES=1	IPL0T(9)=0
C	LEVEL OR LEV		YES=1	IPL0T(12)=0
C	SECOND BEAMS		YES=1	IPL0T(16)=0
C	CONNECTION BEAM	MAIN	YES=1	IPL0T(5)=0
C		SECOND	YES=1	IPL0T(6)=0
C	MARK	COLUMN	YES=1	IPL0T(13)=0
C		BEAM MAIN	YES=1	IPL0T(14)=0
C		SECON	YES=1	IPL0T(15)=0
C	SIZE	COLUMN	NO= 0	IPL0T(17)=1
C		BEAM MAIN	NO= 0	IPL0T(18)=1
C		SECOND	NO= 0	IPL0T(19)=1
C	JOINT		NO= 0	IPL0T(11)=1
C	SCALE		1/20	IPL0T(20)=SC
C	DMF(MODIF. FACTOR)		-	IPL0T(10)=F
C	DEVICE	TEKTRONIK	TEK=1	IPL0T(7)=1

C		HF	TEK	IPLCT(7)=2
C		CALCOM	TEK	IPLCT(7)=3
C	DRAWING SIZE A		4	IPLCT(21)=9
C		B	4	7
C		C	4	5
C		D	4	3
C		E	4	1
C	RAND LOGG		-	IPLCT(22)=7

C SET IPLCT TO DEFAULT VALUES JPLCT

C

I=2
LET(12)=LET(12)+1
IF(CO(I).EQ.4)TEMP)RETURN

C

C LET(12) COUNTS NUMBER OF PLOT COMMANDS ENTERED(MAIN)

C

CALL LAST(I,NB)
IF(I.GT.NB)GO TO 4
IF(KEY(I).EQ.1)GO TO 1003

4 CONTINUE

DO 2 II=1,25
IPLCT(II)=JPLCT(II)

2 CONTINUE

IF(IDUM(1).EQ.7.OR.IDUM(1).EQ.8)IPLCT(22)=IDUM(1)
IF(IPUG.EQ.53.OR.IBU6.EQ.63)

WRITE(6,(IDUM=,2I5))IDUM(1),IPLCT(22)

IF(I.GT.NB)GO TO 10

DO 5 J=1,4

IF(CO(I).NE.PL(J))GO TO 5
GO TO(10,20,30,40)J

CALL 60TOER

5 CONTINUE

GO TO 50

C

C "ALL" OR "ALL" FOLLOWED BY SECOND COMMAND

C SET IPLCT(1)=6

C

10 CONTINUE

I=I+1

IPLCT(1)=6

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 195

IF(KEY(I).EQ.1)GO TO 1003

GO TO 120

C

C FRAME ALL OR "FRAME" FOLLOWED BY S.C./M.C.

C SET IPLCT(1)=1

C

20 CONTINUE

IPLCT(1)=1

21 CONTINUE

I=I+1

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 195

IF(KEY(I).EQ.1)GO TO 1003

```
IF(CO(I).EQ.PL(1))GO TO 21
DO 22 J=3,5
IF(CO(I).EQ.PL(J))GO TO 190
```

```
22 CONTINUE
GO TO 120
```

```
C
C PERSPECTIVE FOLLOWED BY P.P. COMM. 4 TO 7
C SET IPLOT(1)=5
30 CONTINUE
```

```
IPLOT(1)=5
I=I+1
CALL LAST(I,NB)
IF(I.GT.NB)GO TO 195
IF(KEY(I).EQ.1)GO TO 1003
DO 32 J=4,7
IF(CO(I).EQ.PL(J))GO TO 190
32 CONTINUE
GO TO 1004
```

```
C
C TRUSSES OR TR. FOLLOWED BY 3 5 TO 7, S.C
C SET IPLOT(1)=7
```

```
40 CONTINUE
IPLOT(1)=7
I=I+1
CALL LAST(I,NB)
IF(I.GT.NB)GO TO 195
IF(KEY(I).EQ.1)GO TO 1003
DO 42 J=3,7
IF(CO(I).EQ.PL(J))GO TO 190
42 CONTINUE
GO TO 120
```

```
C
C PP. COMMANDS 5 TO 7
C
```

```
50 CONTINUE
DO 52 J=5,7
IF(CO(I).NE.PL(J))GO TO 52
JJ=J-4
GO TO(60,70,80)JJ
CALL 60TOER
52 CONTINUE
GO TO 120
```

```
C
C FLOOR/FLOOR ALL/FLOOR+PP.COMM/FLOOR+LABEL DATA/
C FLOOR+SECOND COMM/
C SET IPLOT(1)=4
```

```
60 CONTINUE
IPLOT(1)=4
GO TO 90
```

```
C
C XFRAM/XFRAM+PP. COMM/XFRAM+LABEL DATA/
C XFRAM+SECOND COMM/
C SET IPLOT(1)=2
```

```
70 CONTINUE
```

```
      IPLOT(1)=2
      GO TO 50
C
C SAME AS XFRAM
C YFRAM
C SET IPLOT(1)=3
C
      80 CONTINUE
      IPLOT(1)=3
      90 CONTINUE
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 195
      IF(KEY(I).EQ.1)GO TO 100
      DO 92 J=1,7
      IF(CO(I).EQ.PL(J))GO TO 190
      92 CONTINUE
      GO TO 120
C
C LAPE DATA L1 TO L2/L1 L3..... L5 TO L6
C PACK LABEL DATA IN PLOT(2),IPLOT(3),IPLOT(4)
C
      100 CONTINUE
      ITEM=I
      ITEM1=CO(I)
      IF(IBUG.EQ.53)PRINT 999,I,ITEM1
      999 FORMAT(' 102',2I6)
      CALL SQZBITS(1,IPLOT(2),ITEM1,1,IER)
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 195
      IF(KEY(I).EQ.1)GO TO 100
      DO 102 J=3,7
      IF(CO(I).EQ.PL(J))GO TO 190
      102 CONTINUE
      IF(CO(I).NE.4NTHRU)GO TO 120
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 1001
      IF(KEY(I).EQ.0)GO TO 1002
      L2=CO(I)
      IF(CO(ITEM).GT.L2)L2=CO(ITEM)-1
      IF(CO(ITEM).EQ.(L2+1))GO TO 104
      L1=CO(ITEM)+1
      GO TO 106
      104 L1=CO(I)
      106 CONTINUE
      DO 108 J=L1,L2
      CALL SQZBITS(1,IPLOT(2),J,1,IER)
      108 CONTINUE
      GO TO 90
C
C SECONDARY COMMANDS
C
      120 CONTINUE
      IQ=1
      IF(IBUG.EQ.53)PRINT 993,IQ,I,CO(I)
```

993 FORMAT(' IQ,2I4,1X,A10')

DO 122 J=8,12

IF(CO(I).NE.PL(J))GO TO 122

121 I=I+1

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 1001

IF(KEY(I).EQ.1)GO TO 1003

IF(CO(I).EQ.PL(26))GO TO 1005

IF(CO(I).EQ.PL(27))GO TO 1004

IF(J.LT.12)GO TO 1004

IF(J.EQ.12.AND.CO(I).EQ.PL(25))GO TO 121

GO TO 1004

122 CONTINUE

DO 124 J=13,16

IF(CO(I).NE.PL(J))GO TO 124

I=I+1

IQ=2

IF(1BUG.EQ.53)PRINT 993,IQ,I,CO(I)

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 1001

IF(KEY(I).EQ.1)GO TO 1003

IF(CO(I).EQ.PL(26))GO TO 1005

IF(CO(I).EQ.PL(27))GO TO 1004

IF(J.EQ.16)GO TO 1004

IF(CO(I).EQ.PL(25))GO TO 1005

IF(J.EQ.13)GO TO 1004

IF(CO(I).EQ.PL(24))GO TO 1005

GO TO 1004

124 CONTINUE

C

C SCALE AND DMF

C

DO 128 J=17,18

IF(CO(I).NE.PL(J))GO TO 128

I=I+1

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 1001

IF(KEY(I).EQ.0)GO TO 1002

IF(J.EQ.18)GO TO 125

IF(CO(I).NE.1)GO TO 1005

I=I+1

IF(J.EQ.17)IPLOT(20)=CO(I)

125 IF(J.EQ.18)IPLOT(10)=CO(I)

I=I+1

CALL LAST(I,NB)

IF(I.GT.NB)GO TO 1001

IF(KEY(I).EQ.1)GO TO 1003

DO 126 JJ=2,7

IF(CO(I).EQ.PL(JJ))GO TO 1005

126 CONTINUE

GO TO 128

128 CONTINUE

IF(CO(I).EQ.PL(19))GO TO 130

IF(CO(I).EQ.4NDRAW)GO TO 200

GO TO 1004

C

C DEVICE

C

```
130 CONTINUE
  I=I+1
  CALL LAST(I,NR)
  IF(I.GT.NR)GO TO 1001
  IF(KEY(I).EQ.1)GO TO 1003
  DO 132 JJ=21,23
  IF(CO(I).NE.PL(JJ))GO TO 132
  IF(JJ.EQ.21)GO TO 134
  IF(JJ.EQ.22)IPLOT(7)=2
  IF(JJ.EQ.23)IPLOT(7)=3
134 CONTINUE
  I=I+1
  CALL LAST(I,NR)
  IF(I.GT.NR)GO TO 195
  IF(KEY(I).EQ.1)GO TO 1003
  DO 136 J1=2,7
  IF(CO(I).EQ.PL(J1))GO TO 190
136 CONTINUE
  GO TO 120
138 CONTINUE
```

C

C "YES"

C

```
140 CONTINUE
  IF(J.LT.15)GO TO 180
  IF(J.EQ.16)GO TO 142
  IF(J1.EQ.25)GO TO 141
  IPLOT(17)=1
141 IPLOT(18)=1
  IPLOT(19)=1
  GO TO 190
142 CONTINUE
  IPLOT(11)=1
  GO TO 190
```

C

C "NO" 8 TO 16

C

```
144 CONTINUE
  J2=J-7
  GO TO(150,152,154,154,156,158,160,162,164)J2
  CALL GOTOER
150 CONTINUE
  IPLOT(8)=0
  GO TO 190
152 CONTINUE
  IPLOT(9)=0
  GO TO 190
154 CONTINUE
  IPLOT(12)=0
  GO TO 180
156 CONTINUE
  IPLOT(16)=0
  GO TO 180
158 CONTINUE
  IPLOT(5)=0
  IPLOT(6)=0
```



```
      GO TO 140
160 CONTINUE
      IPLOT(14)=0
      IPLOT(15)=0
      GO TO 140
162 CONTINUE
      IF(J1.EQ.25)GO TO 163
      IPLOT(17)=1
163 IPLOT(18)=0
      IPLOT(19)=0
      GO TO 140
164 CONTINUE
      IPLOT(11)=1
      GO TO 140
```

C

C WORD "BEAM" FOR CONNECTION, MARK, AND SIZE

C

```
166 CONTINUE
      J1=0
      IF(J.EQ.14.OR.J.EQ.15)J1=25
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 1001
      IF(KEY(I).EQ.1)GO TO 1003
      IF(IBUG.EQ.53)PRINT 994,I,CO(I)
994 FORMAT(' IQ3',I6,1X,A10)
      IF(CO(I).EQ.PL(20))GO TO 168
      IF(CO(I).EQ.PL(12))GO TO 172
      IF(CO(I).EQ.PL(26))GO TO 140
      IF(CO(I).EQ.PL(27))GO TO 144
      GO TO 1004
```

C

C "MAIN"

C

```
168 CONTINUE
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 1001
      IF(KEY(I).EQ.1)GO TO 1003
      IF(CO(I).EQ.PL(27))GO TO 170
      IF(J.LT.15)GO TO 180
      IPLOT(18)=1
      GO TO 140
170 CONTINUE
      IF(J.EQ.15)GO TO 180
      IF(J.EQ.13)IPLOT(5)=0
      IF(J.EQ.14)IPLOT(14)=0
      GO TO 180
```

C

C SECONDARY

C

```
172 CONTINUE
      I=I+1
      CALL LAST(I,NB)
      IF(I.GT.NB)GO TO 1001
      IF(KEY(I).EQ.1)GO TO 1003
      IF(CO(I).EQ.PL(27))GO TO 174
```

IF(J.LT.15)GO TO 180
 IPLOT(19)=1
 GO TO 180

174 CONTINUE
 IF(J.EQ.15)GO TO 180
 IF(J.EQ.13)IPLOT(6)=0
 IF(J.EQ.14)IPLOT(15)=0
 GO TO 180

C
 C "COLUMN"
 C

176 CONTINUE
 I=I+1
 CALL LAST(I,NB)
 IF(I.GT.NB)GO TO 1001
 IF(KEY(I).EQ.1)GO TO 1003
 IF(CO(I).EQ.PL(27))GO TO 178
 IF(J.EQ.14)GO TO 180
 IF(J.EQ.15)IPLOT(17)=1
 GO TO 180

178 CONTINUE
 IF(J.EQ.15)GO TO 180
 IF(J.EQ.14)IPLOT(13)=0
 GO TO 180

C
 C DRAWING SIZE
 C DEFAULT SIZE D IPLOT(21)=3
 C

200 CONTINUE
 I=I+1
 CALL LAST(I,NB)
 IF(I.GT.NB)GO TO 1001
 IF(KEY(I).EQ.1)GO TO 1003
 IF(CO(I).EQ.4*SIZE)GO TO 200
 ITEMP2=0
 IF(CO(I).EQ.1HA)ITEMP2=9
 IF(CO(I).EQ.2HBM)ITEMP2=7
 IF(CO(I).EQ.1HC)ITEMP2=5
 IF(CO(I).EQ.1MD)ITEMP2=3
 IF(CO(I).EQ.1HE)ITEMP2=1
 IF(ITEMP2.EQ.0)GO TO 1004
 IPLOT(21)=ITEMP2

C
 C
 C WRITE OUT AND GO BACK

C*****

C

180 CONTINUE
 I=I+1
 CALL LAST(I,NB)
 IF(I.GT.NB)GO TO 195
 IF(KEY(I).EQ.1)GO TO 1003
 DO 182 JJ=2,7
 IF(CO(I).EQ.PL(JJ))GO TO 190
 182 CONTINUE
 GO TO 120
 190 CONTINUE

```
      LET(12)=LET(12)+1
195  CONTINUE
      WRITE(P1,750)IPLOT
750  FORMAT(3I20)
      IF(IPUG.EQ.53)PRINT 995,IPLOT(1),LET(12)
995  FORMAT(*I0*,2I4)
      DO 900 IPOS=1,140
      CALL UNSQZB(1,IPLOT(2),IPOS,IRI,IER)
      IRIT(IPOS)=IRI
900  CONTINUE
      DO 950 II=1,140
      IF(IPUG.EQ.53)PRINT 996,II,IRIT(II)
996  FORMAT(* I01*,20I3)
950  CONTINUE
      IF(IPUG.EQ.53)PRINT 997,(IPLOT(J),J=2,4)
997  FORMAT(* I0*,3I20)
      IF(IPUG.EQ.53)PRINT 998,(IPLOT(J),J=5,20)
998  FORMAT(* I0*,5I4)
C      REWIND P1
C      DO 902 IRE=1,2
C      READ(P1,750)IPLOT
C      DO 901 IPOS=1,140
C      CALL UNSQZB(1,IPLOT(2),IPOS,IRI,IER)
C      IRIT(IPOS)=IRI
C 901  CONTINUE
C      DO 952 II=1,140
C      IF(IPUG.EQ.53)PRINT 996,II,IRIT(II)
C 952  CONTINUE
C 902  CONTINUE
      IF(I.GT.NB)GO TO 1010
      GO TO 4
C ----- MESSAGES
C INCOMPLETE
1001 CONTINUE
      IF(KEY(I).NE.0) CO(I)=4H =
      CALL SYNTAX(1,I,CO(I),I)
      GO TO 1010
C NUMBER EXPECTED
1002 CALL SYNTAX(2,I,CO(I),I)
      GO TO 1010
C WORD EXPECTED
1003 CALL SYNTAX(3,I,CO(I),I)
      GO TO 1010
C ILLEGAL WORD
1004 CALL SYNTAX(4,I,CO(I),I)
      GO TO 1010
C ILLEGAL NUMBER
1005 CALL SYNTAX(5,I,CO(I),I)
      GO TO 1010
C ILLEGAL WORD, WORD ASSUMED ---
1006 CALL SYNTAX(6,I,CO(I),DUM)
      GO TO 1010
C ILLEGAL NUMBER, NUMBER ASSUMED ---
1007 CALL SYNTAX(7,I,CO(I),NS)
      GO TO 1010
C MAX EXPECTED VALUE EXCEEDED, VALUE ACCEPTED
1009 CALL SYNTAX(9,I,CO(I),7.)
```

GO TO 1010

C
1010 CONTINUE

C

RETURN
END

LE MAP--(LC=4)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
642B	//		REAL	20	JET	724B	//
OP	/PACK1/		REAL	80	JJ	1675B	
120B	/PACK1/		REAL	80	JFLOT	OP	/PLOTIN/
423B	//		REAL	2	JSTOP	111B	//
ESP	//		REAL	3	J1	1705B	
1712B		*S*	REAL		J2	1706B	
12B	/DYNIN/		REAL	20	KAK	124B	//
115B	//		REAL		KFT	736B	//
116B	//		REAL		KEY	240B	/PACK1/
1672B			INTEGER		LFT	762B	//
1711B			INTEGER		LFV	2B	//
1455B			INTEGER	140	LIST	410B	//
1671B			INTEGER		LNFIN	5B	//
103B	//		INTEGER	6	LNMAX	7B	//
75B	//		INTEGER	6	LNJ	123B	//
74B	//		INTEGER		LCN	666B	//
60B	//		INTEGER	6	LVFIN	4B	//
66B	//		INTEGER	6	LVMAX	6B	//
OP	/BMLN7/		INTEGER	35	LVU	122B	//
1700B			INTEGER		LYN	OB	/DYNIN/
12B	//		INTEGER		L1	1702B	
11B	//		INTEGER		L2	1701B	
1673B			INTEGER		MAT	OB	//
114B	//		INTEGER		MAXFR	117B	//
422B	//		INTEGER		MD	425B	//
31B	/PLOTIN/		INTEGER	25	MEM	1B	//
1710B			INTEGER		NE	1	DUMMY-AL
113B	//		INTEGER		NEZ	476B	//
1704B			INTEGER		NL	10B	//
121B	//		INTEGER		NME	14B	//
3B	//		INTEGER		NSTOP	112B	//
475B	//		INTEGER		PL	1417B	
1676B			INTEGER		TAKI	712B	//
1707B			INTEGER		UF	451B	//
1677B			INTEGER		USF	630B	//
120B	//		INTEGER		WS	1713B	
1674B			INTEGER		ZZZZZ	13B	//

SUPROUTINE PLOTS(KPLOT,TWNY)

COMMON MAT,MEM,LEV,ISOLV,LVFIN,LAFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON 72222,NME,DAT(3),IDCC(6),IDOS(6),MAXFL,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LAN
COMMON KAK,LIST(10),IO9,CODE(2),MD(20),UF(20)
COMMON TTC1,NE7(90),USF(10)
COMMON PON(20),LON(20)
COMMON TAKI(10),JFT(10)
COMMON KET(20),LET(20)

COMMON/RATS/BATP(10)
COMMON/MASSES/MASS(24)
DIMENSION NME(11,3)
INTEGER KAK(90,2)
COMMON /CONSTS/ CONSTS(10),IECF,ACC(3)
COMMON/TUES/TUF(7)

COMMON/FMLN7/ DUM(35),LHEAD
COMMON/DAIND/ MINDEX(92)
COMMON/DADAT/LINDEX(171),HTZ(22),PINDEX(21)
..LDI(22),CBM(2,2,28),CAF(28),VIZ(28),ROZ(2,28)
DIMENSION LIND2(171),CJP(18),CLDAT(10,6)
COMMON /DADAT2/LIND2,CJP,CLDAT

C EQUIVALENCE (LINDEX,LIND2),(CJP,HTZ),(CLDAT,HTZ(19))
COMMON/DAIND2/ MSOP2,MTNC2(=1)

C FRAME DATA

DIMENSION CGA(90),CGA1(90),DIST(30,2)
1,MKC(140,2),COL1(30,2),INWAY(2),INFR(2),INMB(2),INLEV(2)
2,HX(2),LDTY(28),PERC(90,2)
3,LDOLD(28),LDWAY(28),ULTVAL(28),LATIN(28)
INTEGER COL1,BYPASS,BAYMX

C COMMON TO A AND B

DIMENSION BSTIF(980),PSTIF(980),BLEN(980),BLEN1(980)
1,MKB(980),CSTIF(992),CSTIF1(992),CLEN1(992),CAX(992),PR(992)
2,PC(992),PHAS(540),PHAS1(540),PHAST(90,2)
3,IBR(980),IBR1(980),LOYES(10),PD(992)
EQUIVALENCE (PD(1),PR(1))

C FRAME B ONLY

DIMENSION DR(992),DR1(992),DV(992),PA(992),PHIC(992)
A,PHIC1(992),TA(992),TD(91),P(91),P1(91),SELI(91),TG(140),TH(91)
B,TRANS(91),TRANS1(91),NC1(140,2)

C

COMMON/FRAMB/CGA,CGA1,LDSW,LDSW1,DIST,MKC,COL1,INWAY,INFR,INMB
A,INLEV,HX,LDTY,PERC,MBEX,BYPASS,IVAY,IFR,MP,MB1,LEVNX
B,BSTIF,BSTIF1,BLEN,BLEN1,MKB,CSTIF,CSTIF1,CLEN1,CAX,PB,PC,PHAS
C,PHAS1,PHAST,DR,DR1,DV,PA,PHIC,PHIC1,TA,TD,P,P1,SELI,TG,TH,TRANS
D,TRANS1,NC1,KMON,KMON1,KSHR,KSHR1,KDV,KDV1

E,LD,N90,NIF,JIF,DELTA,LVMA,E,BAYMX,EST,IV,IV1,
FIBR,IBR1,KST(980),KST1(980),MSW,IDFR(140,2),HRJ(992),HRJ1(992)
G,STB
H,LDOLD,LDWAY,HRFL(91),HRFL1(91),LOYES,BUF(12)

EQUIVALENCE(NIF,L2SAVE)
COMMON/LEN/TBLEN,NRM,LBLAST

C

DIMENSION CX(992),CZ(992)
EQUIVALENCE (CX(1),PA(1)),(CZ(1),PR(1))
COMMON/PLOT/IFLA,KCC,KC,L,LEVEL
DIMENSION JBR(980)
DIMENSION ISTAC(140)

EQUIVALENCE (JER(1),PHIC(1)),(ISTAC(1),CR1(1))
 CALL ZERO(ISTAC,140)

C

C START TO COMPUTE JOINT COORDINATES

C

CALL ZERO(CX,992)
 CALL ZERO(CZ,992)
 CALL ZERO(IFLA,4)
 CALL ZERO(JBR,990)

IPUG=LE*(11)
 LV1=LV=LVNC=IDOS(3)+1
 NCP=1+(LVNC-LV1)/LVH4

C

C COMPUTE FIRST STACK AND LAST STACK NO. ON FRAME NFR

C KA=FIRST STACK

C KP=LAST STACK

C

C

DO 500 IFR=1,IFR
 KA=COL1(NFR,IW)
 IF(NFR.LT,IFR)KP=COL1(NFR+1,IW)-1
 IF(NFR.EQ,IFR)KP=MB

C

C SEARCH FOR ONE STACK WITHOUT SHORT DUMMY COLUMN

C THIS STACK WILL BE REFERENCE STACK

C SET FLAG IFLA=0 TO INDICATE STACK HAD BEEN FOUND

C SET ISTAC(K)=0,ISTAC(K)=1 IF SHORT OR DUMMY THPOUGHT COLUMN

C HAS BEEN FOUND.

IFLA=1

LEVEL=LVH4

DO 3 LL=KA,KP

KK=LL

2 LVN=(KK-1)*LVH4+LEVEL

IF(CSTIF(LVN).LE,1.E-20)GO TO 4

GO TO 9

4 CONTINUE

IF(KK.EQ,KB)LEVEL=LEVEL-1

IF(KK.EQ,KB)GO TO 3

KK=KK+1

GO TO 2

3 CONTINUE

9 CONTINUE

DO 100 K=KA,KB

ISTAC(K)=0

IF(K.NE,KA)GO TO 6

NCP=KA

LV=LV1

LVNC=(NCP-1)*LVH4+LV

6 CONTINUE

C

C SEARCH FOR JOINT BELOW REFERENCE LEVEL

C

IQ=4

IF(IPUG.EQ,53)

1PRINT 996,IQ,K,LV,LVNC,NCP,KB,IDOS(3),KC,KCC,CX(LVNC),CZ(LVNC)

996 FORMAT('IQ=',I4,2F18.2)

CALL STACDN5(LV,LVNC,K)

```
      IO=1
      IF(IFLA.EQ.1)GO TO 66
      IF(IPUG.EQ.53)
1PRINT 994,IO,K,LV1,LV,LVNC,L,CX(L),CZ(L)
994 FORMAT(' IO',6I4,2F7.2)
      66 CONTINUE
      IF(IFLA.EQ.0)GO TO 67
      IF(KC.EQ.KB.AND.ISTAC(K).EQ.1)GO TO 7
      IF(ISTAC(K).EQ.1)GO TO 8
      67 LVNC1=LVNC
      C SET POINTER TO REFERENCE LEVEL ,REFERENCE STACK
      LV=LV1
      LVNC=(NCP-1)*LVN4+LV
      C SET ORIGIN CZ(1)=0. IN STACK 1 AND LEVEL 1
      C
      IF(IFLA.EQ.1)GO TO 5
      IF(K.NE.KC)GO TO 15
      CZ1=CZ(LVNC1)
      DO 10 KK=LVNC1,LVNC
      IO=3
      IF(IPUG.EQ.53)
1PRINT 994,IO,K,KK,LVNC1,LVNC,LV,CZ(KK)
10 CONTINUE
      C
      C SINCE LOWER JOINT COORDINATES HAVE BEEN SET,
      C COMPUTE UPPER JOINT COORDINATE IN REFERENCE LEVEL
      C
      15 CONTINUE
      IO=33
      IF(IPUG.EQ.53)
1PRINT 991,IO,K,LV,LVNC
991 FORMAT(' IO33',4I4)
      IF(IPUG.EQ.53)
1PRINT 993,(KK,CZ(KK),KK=LVNC1,LVNC)
993 FORMAT(' IO34',I4,F10.2)
      5 CALL STACUP5(LV,LVNC,K)
      IO=2
      IF(IFLA.EQ.1)GO TO 68
      IF(IPUG.EQ.53)
1PRINT 994,IO,K,LV1,LV,LVNC,L,CX(L),CZ(L)
68 CONTINUE
      IF(ISTAC(K).EQ.1)GO TO 8
      IF(L.EQ.0.AND.IFLA.EQ.0)GO TO 8
      IF(L.EQ.0.AND.IFLA.EQ.1)GO TO 11
      IF(LV.EQ.LEVEL.AND.IFLA.EQ.0)GO TO 8
      IF(IFLA.EQ.0)GO TO 7
      11 KC=K
      KCC=K
      7 CONTINUE
      IFLA=0
      LV=LV1
      LVNC=(NCP-1)*LVN4+LV
      CX(LVNC)=0.
      CZ(LVNC)=0.
      GO TO 6
      8 CONTINUE
      IO=0
```

```

      IF (IPUG.EQ.53)
      1PRINT 997,I0,K,NCP,KA,KB,KC,IFLA,IFR,NFR
997 FORMAT(' I0=*,10I4)
      IF (IFLA.EQ.1) KC=KCC=K
      IF (IFLA.EQ.1) GO TO 84
      IF (NCP.EQ.KB.AND.KC.NE.KA) GO TO 90
      IF (NCP.EQ.KB) GO TO 100
      GO TO 84
84 CONTINUE
      LV=LV1
      LVNC=(KC-1)*LVN4+LV
      GO TO 84
85 CONTINUE
      LV=LV1
      LVNC=(NCP-1)*LVN4+LV
C ADJUST X COORDINATES
C
C LOOK UP FOR NEXT STACK TO THE RIGHT
C
86 CALL STACR15(LV,NCP,LVNC,K)
      IF (KC.EQ.KB.AND.ISTAC(K).EQ.1) GO TO 87
      GO TO 100
87 CONTINUE
      IFLA=0
      LV=LV1
      NCP=KB
      LVNC=(NCP-1)*LVN4+LV
      GO TO 84
90 CONTINUE
      IF (KCC.EQ.KA) GO TO 100
      IF (ISTAC(KCC-1).EQ.0.AND.IFLA.EQ.0) GO TO 100
      LV=LV1
      LVNC=(KCC-1)*LVN4+LV
      IFLA=0
      CALL STACLE5(LV,NCP,LVNC,K)
      CALL STACDN5(LV,LVNC,K)
      IQ=5
      IF (IPUG.EQ.53)
      1PRINT 998,I0,K,LV1,LV,LVNC,L,KC,KCC,CX(L),CZ(L)
998 FORMAT(' I0=,8I4,2F7.2)
      LV=LV1
      LVNC=(NCP-1)*LVN4+LV
      CALL STACUP5(LV,LVNC,K)
      IQ=6
      IF (IPUG.EQ.53)
      1PRINT 998,I0,K,LV1,LV,LVNC,L,KC,KCC,CX(L),CZ(L)
      IF (NCP.EQ.KA) GO TO 95
      IF (KCC.EQ.(KA-1)) GO TO 100
      GO TO 90
95 CONTINUE
C RESET X COORDINATES
      IF (KC.EQ.KA) GO TO 100
      LVN1=(KA-1)*LVN4+1
      LVN2=KB+LVN4
      CX1=CX(LVN1)
      DO 96 KK=LVN1,LVN2
      IF (CX(KK).EQ.999.) GO TO 96

```



```

      CX(KK)=APS(CX1-CX(KK))
      96 CONTINUE
      100 CONTINUE
C      DETERMINE JOINT FOR BRACINGS
      IF(MSW.EQ.1)GO TO 500
      KB1=KB-1
      DO 400 I=KA,KB1
      DO 300 LV=2,LVM4
      LN2=(I-1)*LVM4+LV
      IQ=1
      IF(IEUG.EQ.53)
      1PRINT 999,IQ,NFR,I,LV,LN2,KA,KB1
      IF(IPR(LN2).LT.10)GO TO 300
      NRM=I
      LVNC1=LN2
      300 NCP=NRP
      LVNC1=JTRI52(NCP,LVNC1)
      IF(LVNC1.EQ.0)GO TO 380
      IF(CLENI(LVNC1).LT.1.)GO TO 320
      GO TO 300
      320 CONTINUE
      NCP=NRM
      JL=MERD52(LV,I,LN2,DUM(1),1,LVT)
      JR=MERD52(LV,NCP,LVNC1,DUM(1),2,LVT)
      IQ=11
      IF(IEUG.EQ.53)
      1PRINT 999,IQ,I,LN2,NCP,LV,JL,JR
      999 FORMAT(' IQ',8I4)
      JBR(LN2)=JL+JR*10**5
      380 CONTINUE
      400 CONTINUE
      500 CONTINUE
      NC=MB+LVM4
      NB=NC-LVM4
      IQ=9
      IF(IEUG.EQ.53)
      1PRINT 997,IQ,IFR,NB,NC,LVM4
      WRITE(81,800) IW,IW*X,NB,NC,IFR,INWAY(IW),LVM4,MB,LD,MSW
      800 FORMAT(6I10)
      WRITE(81,800)(MK8(K),K=1,NB),(MKC(K,IW),K=1,MB)
      1,(COL1(K,IW),K=1,IFR)
      WRITE(81,805)(KST(K),K=1,NB)
      805 FORMAT(4I15)
      WRITE(81,810)(BSTIF(K),BLEN(K),K=1,NB),
      1(CLENI(K),CSTIF(K),CX(K),CZ(K),PHIC(K),DV(K),K=1,NC),
      2(TRANS(K),K=1,LVM4)
      810 FORMAT(6E12.5)
      WRITE(81,820) (JBR(K),JBR(K),K=1,NB)
      820 FORMAT(4I15)
      IFRAM1=1HX
      IF(INWAY(IW).EQ.2)IFRAM1=1HY
      IF(IEUG.EQ.53)
      1PRINT 995,IFRAM1
      995 FORMAT(' **** FRAME  *A2,* DIRECTION  ****')
      IF(IEUG.EQ.53)
      1PRINT 998
      998 FORMAT(' IQ *,*JOINT NO.  LEV  CX  CZ  CLEN  BLEN  *)

```

```

DO 200 NC=1,MP
DO 200 LV=1,LVM4
LVNC=(NC-1)*LVM4+LV
  IF (IBUG.EQ.53)
    1PRINT 9=2, LVNC, LV, CX(LVNC), CZ(LVNC), CLEN1(LVNC), BLFN(LVNC)
  92 FORMAT(' ', I4, 2X, I3, 2X, 4F7.2)
200 CONTINUE
RETURN
END

```

LE MAP--(LC=A)

-ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE -NAME---ADDRESS--BLOCK--

13P	/CONSTS/		REAL	3	HRJ1	63022B	/FRAMB/
0P	/BATS/		REAL	10	HX	1116B	/FRAMB/
50705P	/FRAMB/		INTEGER		I	1362B	
307B	/DADAT/		REAL	21	IER	50711B	/FRAMB/
5317P	/FRAMB/		REAL	980	IPR1	52635B	/FRAMB/
7243B	/FRAMB/		REAL	980	IBUG	1333B	
642B	//		REAL	20	ICRIC	183B	//
1447B	/FRAMB/		REAL	980	ICRIS	75B	//
3373B	/FRAMB/		REAL	980	IDFR	60432B	/FRAMB/
65353B	/FRAMB/		REAL	12	IDOC	60B	//
1441B	/FRAMB/		INTEGER		IDOS	66B	//
550B	/DADAT/		REAL	28	IEOF	12B	/CONSTS
20753B	/FRAMB/		REAL	992	IFFRM	12B	//
370B	/DADAT/		REAL	112	IFLA	0B	/PLOT/
0B	/FRAMB/		REAL	90	IFLEV	11B	//
132B	/FRAMB/		REAL	90	IFR	1443B	/FRAMB/
253B	/DADAT2/		REAL	18	IFRAM1	1376B	
275B	/DADAT2/		REAL	60	ILINK	114B	//
17013B	/FRAMB/		REAL	992	INFR	1110B	/FRAMB/
423B	//		REAL	2	INLEV	1114B	/FRAMB/
1012B	/FRAMB/		INTEGER	60	INMB	1112B	/FRAMB/
0B	/CONSTS/		REAL	10	INWAY	1106B	/FRAMB/
13113B	/FRAMB/		REAL	992	IO9	422B	//
15053B	/FRAMB/		REAL	992	IPRINT	113B	//
37027B	/FRAMB/	EQV	REAL	992	IQ	1351B	
1360B			REAL		IRUN	121B	//
31167B	/FRAMB/	EQV	REAL	992	ISOLV	3B	//
1353B		*S*	REAL		ISTAC	33127B	/FRAMB/
55B	//		REAL	3	ITC1	475B	//
50702P	/FRAMB/		REAL		ITEST	120B	//
266B	/FRAMB/		REAL	60	IW	50787B	/FRAMB/
31167B	/FRAMB/	EQV	REAL	992	IWAY	1442B	/FRAMB/
33127B	/FRAMB/	EQV	REAL	992	IWMX	2	DUNNY-A
0B	/BMLN7/		REAL	35	IWI	50710B	/FRAMB/
35067B	/FRAMB/		REAL	992	JBR	40767B	/FRAMB/
50704B	/FRAMB/		REAL		JET	724B	//
115B	//		REAL		JIF	50701B	/FRAMB/
116B	//		REAL		JL	1367B	
50706B	/FRAMB/		REAL		JR	1370B	
253B	/DADAT/		REAL	28	JSTOP	111B	//
65053B	/FRAMB/		REAL	91	K	1347B	
65206B	/FRAMB/		REAL	91	KA	1342B	
61062B	/FRAMB/		REAL	992	KAK	124B	//

```

SUBROUTINE STACDN5(LV,LVAC,M)
COMMON MAT,MEW,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFR
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDCS(6),MAXFL,ICRIS(6),ICRIC(4)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IO9,CODE(2),MO(20),UF(20)
COMMON ITC1,NEZ(90),USF(10)
COMMON BON(20),LON(20)
COMMON TAKI(10),JET(10)
COMMON KET(20),LET(20)

COMMON/BATS/BATP(10)
COMMON/MASSES/MASS(24)
DIMENSION NME(11,3)
INTEGER KAK(90,2)
COMMON /CONSTS/ CONSTS(10),IEOF,ACC(3)
COMMON/TUFS/TUF(7)

COMMON/RMLN7/ DUM(35),LHEAD
COMMON/DAIND/ MINDEX(92)
COMMON/DADAT/LINDEX(171),HIZ(29),PINDEX(21)
* ,LDI(28),CBM(2,2,28),CAF(28),VIZ(28),ROZ(2,28)
DIMENSION LIND2(171),CJP(18),CLDAT(10,6)
COMMON /DADAT2/LIND2,CJP,CLDAT

C EQUIVALENCE (LINDEX,LIND2),(CJP,HIZ),(CLDAT,HIZ(19))
COMMON/DAIND2/ MSOP2,MIND2(91)

C FRAME DATA
DIMENSION CGA(90),CGA1(90),DIST(30,2)
1,MKC(140,2),COL1(30,2),INWAY(2),INFR(2),INPB(2),INLEV(2)
2,HX(2),LDTY(28),PERC(90,2)
3,LDOLD(28),LDWAY(28),ULTVAL(28),LATIN(28)
INTEGER COL1,BYPASS,BAYMX
C COMMON TO A AND B
DIMENSION BSTIF( 980),RSTIF1( 980),BLEN( 980),BLEN1( 980)
1,MKB( 980),CSTIF( 992),CSTIF1( 992),CLEN1( 992),CAX( 992),PB( 992)
2,PC( 992),PHAS(540),PHAS1(540),PHAST(90,2)
3,IBR(960),IBR1(980),LDYES(10),PD(992)
EQUIVALENCE (PD(1),PB(1))

C FRAME B ONLY
DIMENSION DR( 992),DR1( 992),DV( 992),PA( 992),PHIC( 992)
A,PHIC1( 992),TA( 992),TD(91),P(91),P1(91),SELI(91),TG(140),TH(91)
B,TRANS(91),TRANS1(91),NC1(140,2)

C
COMMON/FRAMB/CGA,CGA1,LDSW,LDSW1,DIST,MKC,COL1,INWAY,INFR,INPB
A,INLEV,HX,LDTY,PERC,MKEX,BYPASS,INWAY,IFR,MB,MB1,LEVMB
B,BSTIF,BSTIF1,BLEN,BLEN1,MKB,CSTIF,CSTIF1,CLEN1,CAX,PB,PC,PHAS
C,PHAS1,PHAST,DR,DR1,DV,PA,PHIC,PHIC1,TA,TD,P,P1,SELI,TG,TH,TRANS
D,TRANS1,NC1,KHOM,KHOM1,KSHR,KSHR1,KDV,KDV1
E,LD,NPB,NIF,JIF,DELTA,LVM4,E,BAYMX,EST,IW,IW1,
FIBR,IBR1,KST(980),KST1(980),MSW,IDFR(140,2),HRJ(992),HRJ1(992)
G,STB
H,LDOLD,LDWAY,HRFL(91),HRFL1(91),LDYES,BUF(12)
EQUIVALENCE(NIF,L2SAVE)
COMMON/LEN/TBLEN,NRM,LBLAST

C
DIMENSION CX(992),CZ(992)
EQUIVALENCE (CX(1),PA(1)),(CZ(1),DR(1))
COMMON/PLOT/IFLA,KCC,KC,L,LEVEL
DIMENSION ISTAC(140)
EQUIVALENCE (JBR,PHIC(1)),(ISTAC(1),DR1(1))

```

```

      IRUG=LE*(11)
      IF(IFLA.EQ.0.AND.IRUG.EQ.53)PRINT 990
990 FORMAT('IQD ',JOINT NO. LEV CX CZ CLEN BLEN ')
10 L=MCOLD*2(LV,LVNC)
   IF(CLENI(LVNC).LE.0.)GO TO 40
   IF(L.EQ.0)GO TO 40
   IF(IFLA.EQ.0)GO TO 15
   IF(CLENI(LVNC).GT.1)ISTAC(K)=1
   IF((LVNC-L).GT.1)ISTAC(K)=1
   IF(CSTIF(LVNC).EQ.100.)ISTAC(K)=1
   IF(CSTIF(LVNC).LE.1E-20.AND.(LVNC-L).GT.1)ISTAC(K)=1
   IF(ISTAC(K).EQ.1)GO TO 30
   IF(IFLA.EQ.1)GO TO 25
15 CONTINUE
   CX(L)=CX(LVNC)
   IF(CLENI(LVNC).LE.1.)GO TO 20
   LVNC1=(K-1)*LVNC+LV+L-LVNC
   CZ(L)=CZ(LVNC1)
   GO TO 25
20 CZ(L)=CZ(LVNC)-1./CLENI(LVNC)
25 LEVD=LV+L-LVNC
   IF(IFLA.EQ.1)GO TO 28
   IF((LVNC-L).EQ.1)GO TO 29
   L1=L+1
   L2=LVNC-1
   DO 26 I=L1,L2
   CX(I)=99.
   CZ(I)=99.
26 CONTINUE
28 CONTINUE
   IF(IFLA.EQ.0.AND.IRUG.EQ.53)PRINT 992,L,LEVD,CX(L),CZ(L),
   *CLENI(LVNC),BLEN(LVNC)
992 FORMAT(' ',I4,8X,I3,2X,4F7.2)
   LV=LEVD
   LVNC=L
   IF(LV.EQ.1)GO TO 30
   GO TO 10
30 CONTINUE
   RETURN
40 CONTINUE
C
C ERROR MESSAGE (COL.DOES NOT EXIST)
C
   IF(IRUG.EQ.53)
1PRINT 994,LV,LVNC,L,CLENI(LVNC)
994 FORMAT(' ERROR FROM STADN LV LVNC L CLENI',3I4,F12.2)
   STOP
   RETURN
NO PATH TO THIS STATEMENT
END

```

SUBROUTINE STACUPF(LV,LVAC,K)

COMMON MAT,MFM,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM

COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),MAXFL,ICRIS(6),ICRIC(6)

COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,ECF,MAXFR,ITEST,IRUN,LVU,LNU

COMMON KAK,LIST(10),IO9,CODE(2),MD(20),UF(20)

COMMON ITC1,NEZ(90),USF(10)

COMMON FON(20),LON(20)

COMMON TAKI(10),JET(10)

COMMON NET(20),LET(20)

COMMON/RATS/RATP(10)

COMMON/MASSES/MASS(24)

DIMENSION NME(11,3)

INTEGER KAK(90,2)

COMMON /CONSTS/ CONSTS(10),IEOF,ACC(3)

COMMON/TUFS/TUF(7)

COMMON/FMLN7/ DUM(35),LHEAD

COMMON/DAIND/ MINDEX(92)

COMMON/DADAT/LINDEX(171),HIZ(29),BINDEX(21)

,LDI(28),CBM(2,2,28),CAF(28),VIZ(28),RCZ(2,28)

DIMENSION LIND2(171),CJP(18),CLDAT(10,6)

COMMON /DADAT2/LIND2,CJP,CLDAT

C EQUIVALENCE (LINDEX,LIND2),(CJP,HIZ),(CLDAT,HIZ(19))

COMMON/DAIND2/ MSOP2,MINDEX(91)

C FRAME DATA

DIMENSION CGA(90),CGA1(90),DIST(30,2)

1,MKC(140,2),COL1(30,2),INWAY(2),INFR(2),INMR(2),INLEV(2)

2,HX(2),LDTY(28),PERC(90,2)

3,LDOLD(28),LDWAY(28),ULTVAL(28),LATIN(28)

INTEGER COL1,BYPASS,BAYMX

C COMMON TO A AND B

DIMENSION BSTIF(980),BSTIF1(980),BLEN(960),PLEN1(980)

1,MKB(920),CSTIF(992),CSTIF1(992),CLEN1(992),CAY(992),PB(992)

2,PC(992),PHAS(540),PHAS1(540),PHAST(90,2)

3,IBR(980),IBR1(980),LDYES(10),PD(992)

EQUIVALENCE (PD(1),PB(1))

C FRAME B ONLY

DIMENSION DR(992),DR1(992),DV(992),PA(992),PHIC(992)

A,PHIC1(992),TA(992),TD(91),P(91),P1(91),SELI(91),TG(140),TH(91)

B,TRANS(91),TRANS1(91),NC1(140,2)

C

COMMON/FRAMB/CGA,CGA1,LDW,LDW1,DIST,MKC,COL1,INWAY,INFR,INMR

A,INLEV,HX,LDTY,PERC,MBEX,BYPASS,INWAY,IFR,MB,MB1,LEVMX

B,BSTIF,BSTIF1,BLEN,BLEN1,MKB,CSTIF,CSTIF1,CLEN1,CAY,PB,PC,PHAS

C,PHAS1,PHAST,DR,DR1,DV,PA,PHIC,PHIC1,TA,TD,P,P1,SELI,TG,TH,TRANS

D,TRANS1,NC1,KMOM,KMOM1,KSHR,KSHR1,KDV,KDV1

E,LD,N90,NIF,JIF,DELTA,LVM4,E,BAYMX,EST,IW,IW1,

FIBR,IBR1,KST(980),KST1(980),MSW,IOFR(140,2),HRJ(992),HRJ1(992)

G,STB

H,LDOLD,LDWAY,HRFL(91),HRFL1(91),LDYES,BUF(12)

EQUIVALENCE(NIF,L2SAVE)

COMMON/LEN/TBLEN,NRM,LBLAST

C

DIMENSION CX(992),CZ(992)

EQUIVALENCE (CX(1),PA(1)),(CZ(1),DR(1))

COMMON/PLOT/IFLA,KCC,KC,L,LEVEL

DIMENSION ISTAC(140)

EQUIVALENCE (JBR,PHIC(1)),(ISTAC(1),DR1(1))

```

      IPUG=LE*(11)
      IF(IFLA.EQ.0.AND.IBUG.EQ.53)PRINT 990
990 FORMAT('IQU ',JOINT NO.  LEV  CX      CZ      CLEI  BLEN  ')
      L1=LVNC
10  L=MCOLU*2(LV,LVNC)
      IF(L.GT.0)GO TO 12
      IF(IFLA.EQ.1.AND.LV.NE.LEVEL)ISTAC(K)=1
      IF(IFLA.EQ.1.AND.ISTAC(K).EQ.1)RETURN
      IF(LV.EQ.LVM4)RETURN
      L2=LVNC+1
      CX(L2)=99.
      CZ(L2)=99.
      IF(LV.EQ.LEVEL)GO TO 30
      LV=LV+L2-LVNC
      LVNC=L2
      GO TO 1
12  IF(CLENI(L).LE.0.)GO TO 40
      IF(IFLA.EQ.0)GO TO 15
      IF(CLENI(L).GT.1)ISTAC(K)=1
      IF((L-LVNC).GT.1)ISTAC(K)=1
      IF(CSTIF(L).EQ.100.)ISTAC(K)=1
      IF(CSTIF(L).LE.1E-20.AND.(L-LVNC).GT.1)ISTAC(K)=1
      IF(ISTAC(K).EQ.1)RETURN
      IF(IFLA.EQ.1)GO TO 25
15  CONTINUE
      CX(L)=CX(LVNC)
      IF(CLENI(L).LE.1.)GO TO 20
      LVNC1=(KC-1)*LVM4+LV+L-LVNC
      CZ(L)=CZ(LVNC1)
      GO TO 25
20  CZ(L)=CZ(LVNC)+1./CLENI(L)
25  LEVU=LV+L-LVNC
      IF(IFLA.EQ.1)GO TO 28
      IF((L-LVNC).EQ.1)GO TO 20
      L1=LVNC+1
      L2=L-1
      DO 26 II=L1,L2
      CX(II)=99.
      CZ(II)=99.
26  CONTINUE
28  CONTINUE
      IF(IFLA.EQ.0.AND.IBUG.EQ.53)PRINT 992,L,LEVU,CX(L),CZ(L),
      *CLENI(L),BLEN(L)
992 FORMAT(' ',I4,8X,I3,2X,4F7.2)
      LV=LEVU
      LVNC=L
      IF(LV.EQ.LVM4)GO TO 30
      GO TO 10
30  CONTINUE
      RETURN
40  CONTINUE

C
C  ERROR EXIT (COL.DOES NOT EXIST)
      IF(BUG.EQ.53)
      1PRINT 994,LV,LVNC,L,CLENI(L)
994 FORMAT(' ERROR FROM STADN LV LVNC L CLENI',3I4,F12.2)
      STOP

```

SUBROUTINE STACLES(LV,NCP,LVNC,K)

COMMON MAT, MEM, LEV, ISOLV, LVFIN, LNFIN, LVMAX, LOMAX, NL, IFLEV, IFFRM

COMMON ZZZZZ, NME, DAT(3), IDOC(6), IDOS(6), MAXFL, ICRIS(6), ICRIC(6)

COMMON JSTOP, NSTOP, IPRINT, ILINK, EC1, EC5, MAXFR, ITEST, IRUN, LVU, LNU

COMMON KAK, LIST(10), IO9, CODE(2), MD(20), UF(20)

COMMON TTC1, NEZ(90), USF(10)

COMMON PON(28), LON(20)

COMMON TAKI(10), JET(10)

COMMON KET(20), LET(20)

COMMON/BATS/BATP(10)

COMMON/MASSES/MASS(24)

DIMENSION NME(11,3)

INTEGER KAK(90,2)

COMMON /CONSTS/ CONSTS(10), IEOF, ACC(3)

COMMON/TUFS/TUF(7)

COMMON/BMLN7/ DUM(35), LHEAD

COMMON/DAIND/ MINDEX(92)

COMMON/DADAT/LINDEX(171), HI7(28), BINDEX(21)

LDI(28), CBM(2,2,28), CAF(28), VIZ(28), RGZ(2,28)

DIMENSION LIND2(171), CJP(18), CLDAT(10,6)

COMMON /DADAT2/LIND2, CJP, CLDAT

C EQUIVALENCE (LINDEX, LIND2), (CJP, HI7), (CLDAT, HI7(19))

C COMMON/DAIND2/ MSOP2, MINDEX(91)

C FRAME DATA

DIMENSION CGA(90), CGA1(90), DIST(30,2)

1, MKC(140,2), COL1(30,2), INWAY(2), INFR(2), INMP(2), INLEV(2)

2, HX(2), LDY(28), PERC(90,2)

3, LDOLD(28), LDWAY(28), ULTVAL(28), LATIN(28)

INTEGER COL1, BYPASS, BAYMX

C COMMON TO A AND B

DIMENSION BSTIF(980), BSTIF1(980), BLEN(980), BLEN1(980)

1, MKB(980), CSTIF(992), CSTIF1(992), CLEN1(992), CAX(992), PB(992)

2, PC(992), PHAS(540), PHAS1(540), PHAST(90,2)

3, IBR(980), IBR1(980), LDYES(10), PD(992)

EQUIVALENCE (PD(1), PB(1))

C FRAME B ONLY

DIMENSION DR(992), DR1(992), DV(992), PA(992), PHIC(992)

A, PHIC1(992), TA(992), TD(91), P(91), P1(91), SELI(91), TG(140), TH(91)

B, TRANS(91), TRANS1(91), NC1(140,2)

C

COMMON/FRAMB/CGA, CGA1, LDSW, LDSW1, DIST, MKC, COL1, INWAY, INFR, INMB

A, INLEV, HX, LDY, PERC, MBEX, BYPASS, INWAY, IFR, MB, MR1, LEVMX

B, BSTIF, BSTIF1, BLEN, BLEN1, MKB, CSTIF, CSTIF1, CLEN1, CAX, PB, PC, PHAS

C, PHAS1, PHAST, DR, DR1, DV, PA, PHIC, PHIC1, TA, TD, P, P1, SELI, TG, TH, TRANS

D, TRANS1, NC1, KPMO, KPMO1, KSHR, KSHR1, KDV, KDV1

E, LD, N90, NIF, JIF, DELTA, LVH4, E, BAYMX, EST, LV, IW1,

FIBR, IBR1, KST(980), KST1(980), MSW, IDFR(140,2), HRJ(992), HRJ1(992)

G, STR

H, LDOLD, LDWAY, HRFL(91), HRFL1(91), LDYES, BUF(12)

EQUIVALENCE(NIF, L2SAVE)

COMMON/LEN/TBLEN, NRH, LBLAST

C

DIMENSION CX(992), CZ(992)

EQUIVALENCE (CX(1), PA(1)), (CZ(1), DR(1))

COMMON/PLOT/IFLA, KCC, KC, L, LEVEL

COMMON/ALFA/SINA, COSA, ALFA, ID3

DIMENSION ISTAC(140)

EQUIVALENCE (JPP,PHIC(1)),(ISTAT(1),DR1(1))

IPUG=LEFT(11)

IF(IPUG.EQ.53)

IPRINT 990,NCP,LVNC,LV,KC,KCC

990 FORMAT(' IQLE1 NCP LVNC LV L',6I4)

L=JITLEP(NCP,LVNC)

IF(L.EQ.0)GO TO 10

IF(ESTIF(L).LT.0.)GO TO 10

IF(ELEN(L).EQ.0.)GO TO 10

CALL UNF2(KST(L),A1,A2,F)

PM=COSA/BLN(L)

CY(L)=CY(LVNC)-PM

CZ(L)=CZ(LVNC)

NCP=1+(L-LV)/LVMA

LVNC=L

KCC=KCC-1

IF(IPUG.EQ.53)

IPRINT 991,NCP,LVNC,LV,L,KC,KCC

991 FORMAT(' IQLE2 NCP LVNC LV L',6I4)

RETURN

10 CONTINUE

C

C ERROR MESSAGE 1./BLN DOES NOT EXIST

IF(IPUG.EQ.53)

IPRINT 992,NCP,LVNC,LV,L,KC,KCC

992 FORMAT(' IQLE3 NCP LVNC LV L',6I4)

STOP

RETURN

NO PATH TO THIS STATEMENT

END

FILE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
13B	/CONSTS/		REAL	3	CLDAT	275B	/DADAT2
2B	/ALFA/		REAL		CLEN1	17013B	/FRAMB/
136B		*S*	REAL		CODE	423B	//
137B		*S*	REAL		COL1	1012B	/FRAMB/
140B		*S*	REAL		CONSTS	0B	/CONSTS
0B	/RATS/		REAL	10	COSA	1B	/ALFA/
50705B	/FRAMB/		INTEGER		CSTIF	13113B	/FRAMB/
307B	/DADAT/		REAL	21	CSTIF1	15053B	/FRAMB/
5317B	/FRAMB/		REAL	980	CY	37027B	/FRAMB/
7243B	/FRAMB/		REAL	980	CZ	31167B	/FRAMB/
141B			REAL		CAT	55B	//
642B	//		REAL	20	DELTA	50702B	/FRAMB/
1447B	/FRAMB/		REAL	980	DIST	266B	/FRAMB/
3373B	/FRAMB/		REAL	980	DR	31167B	/FRAMB/
65353B	/FRAMB/		REAL	12	DR1	33127B	/FRAMB/
1441B	/FRAMB/		INTEGER		DUM	0B	/BMLN7/
550B	/DADAT/		REAL	28	DV	35067B	/FRAMB/
20753B	/FRAMB/		REAL	992	E	50704B	/FRAMB/
370B	/DADAT/		REAL	112	EC1	115B	//
0B	/FRAMB/		REAL	90	EC5	116B	//
132B	/FRAMB/		REAL	90	EST	50706B	/FRAMB/
253B	/DADAT2/		REAL	18	HIZ	253B	/DADAT/

SUPROUTINE STACRIS(LV,ACF,LVNC,K)

COMMON MAT,MEP,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM

COMMON 77227,NME,DAT(3),IDCC(6),IDCS(6),MAXFL,ICRIS(6),ICRTC(6)

COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFP,ITEST,IRUN,LVU,LNU

COMMON KAK,LIST(10),IO9,CODE(12),MD(20),UF(20)

COMMON ITC1,NEZ(90),USF(10)

COMMON HON(20),LON(20)

COMMON TAKI(10),JET(10)

COMMON MET(20),LET(20)

COMMON/RATS/RATP(10)

COMMON/PASSES/MASS(24)

DIMENSION NME(11,3)

INTEGER KAK(90,2)

COMMON /CONSTS/ CONSTS(17),IEOF,ACC(3)

COMMON/TUFS/TUF(7)

COMMON/RMLN7/ DUM(35),LHEAD

COMMON/PAIND/ MINDEX(92)

COMMON/DADAT/LINDEX(171),HI2(28),BINDEX(21)

,LDI(28),CPM(2,2,28),CAF(28),VIZ(28),ROZ(2,28)

DIMENSION LIND2(171),CJP(18),CLDAT(10,6)

COMMON /DADAT2/LIND2,CJP,CLDAT

C EQUIVALENCE (LINDEX,LIND2),(CJP,HI2),(CLDAT,HI2(19))

COMMON/PAIND2/ MSOP2,MIND2(91)

C FRAME DATA

DIMENSION CGA(90),CGA1(90),DIST(30,2)

1,MKC(140,2),COL1(30,2),INWAY(2),INFR(2),INPP(2),INLEV(2)

2,HX(2),LDTY(28),PERC(90,2)

3,LDOLD(28),LDWAY(28),ULTVAL(28),LATIN(28)

INTEGER COLI,BYPASS,BAYMX

C COMMON TO A AND B

DIMENSION BSTIF(980),BSTIF1(980),BLEN(980),PLEN1(980)

1,MKB(980),CSTIF(992),CSTIF1(992),CLEN1(992),CAY(992),PB(992)

2,PC(992),PMAS(540),PMAS1(540),PMAS1(90,2)

3,IBR(980),IPR1(980),LDYES(10),PD(992)

EQUIVALENCE (PD(1),PB(1))

C FRAME B ONLY

DIMENSION DR(992),DR1(992),DV(992),PA(992),PHIC(992)

A,PHIC1(992),TA(992),TD(91),P(91),P1(91),SELI(91),TG(140),TH(91)

B,TRANS(91),TRANS1(91),NC1(140,2)

C

COMMON/FRAMB/CGA,CGA1,LDOW,LDOW1,DIST,MKC,COL1,INWAY,INFR,INPP

A,INLEV,HX,LDTY,PERC,MBEX,BYPASS,INWAY,IFR,MB,MB1,LEVMBX

B,BSTIF,BSTIF1,BLEN,BLEN1,MKB,CSTIF,CSTIF1,CLEN1,CAY,PB,PC,PMAS

C,PMAS1,PMAS1,DR,DR1,DV,PA,PHIC,PHIC1,TA,TD,P,P1,SELI,TG,TH,TRANS

D,TRANS1,NC1,KHOM,KHOM1,KSHR,KSHR1,KOV,KOV1

E,LD,N90,NIF,JIF,DELTA,LVMA,E,BAYMX,EST,IV,IV1,

FIBR,IBR1,KST(980),KST1(980),MSW,IDFR(140,2),HRJ(992),HRJ1(992)

6,STB

H,LDOLD,LDWAY,HRFL(91),HRFL1(91),LDYES,BUF(12)

EQUIVALENCE(NIF,L2SAVE)

COMMON/LEN/TBLEN,NRP,LBLAST

C

DIMENSION CX(992),CZ(992)

EQUIVALENCE (CX(1),PA(1)),(CZ(1),DR(1))

COMMON/PLOT/IFLA,KCC,KC,L,LEVEL

COMMON/ALFA/SINA,COSA,ALFA,IO3

DIMENSION ISTAC(140)

```

EQUIVALENCE (JFR,PHYC(1)),(ISTAC(1),CP1(1))
IPUG=LET(11)
L=JTRIS(NCP,LVNC)
IF(ISTAC(K).EQ.1.AND.IFLA.EQ.1)GO TO 8
IF(IPSTIF(LVNC).LT.0.)GO TO 10
IF(FLFA(LVNC).LE.0.)GO TO 12
CALL UPR2(KST(LVNC),A1,A2,P)
PM=CCSA/PLEN(LVNC)
CX(L)=CX(LVNC)+PM
IF(IPUG.EQ.53)PRINT 992,L,LVNC,ALFA
992 FORMAT(' **IGR1** L LVNC ALFA*,2I4,F10.2)
CZ(L)=CZ(LVNC)
IQ=1
IF(IPUG.EQ.53)
IFRINT 990,IG,LVNC,L,LV,NCP,A,CX(LVNC),CZ(LVNC),CX(L),CZ(L)
990 FORMAT(' IGR*,6I4,4F10.2)
A NCP=1+(L-LV)/LVNA*
LVNC=L
RETURN
10 CONTINUE

```

```

C
C ERROR EXIT BEAM DOES NOT EXIST
RETURN
END

```

LE MAP--(LDEF)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
13P	/CONSTS/		REAL	3	CSTIF	13113P	/FRAMB/
2B	/ALFA/		REAL		CSTIF1	15053P	/FRAMB/
134P		*S*	REAL		CX	37027P	/FRAMB/
135P		*S*	REAL		CZ	31167P	/FRAMB/
136P		*S*	REAL		DAY	55B	//
0B	/BATS/		REAL	10	DELTA	50702B	/FRAMB/
50705B	/FRAMB/		INTEGER		DIST	266P	/FRAMB/
307P	/DADAT/		REAL	21	DR	31167B	/FRAMB/
5317P	/FRAMB/		REAL	990	DR1	33127P	/FRAMB/
7243B	/FRAMB/		REAL	980	DUM	0B	/BMLN7/
137B			REAL		DV	35067B	/FRAMB/
642B	//		REAL	20	E	50704B	/FRAMB/
1447P	/FRAMB/		REAL	980	EC1	115P	//
3373P	/FRAMB/		REAL	980	EC5	116P	//
65353B	/FRAMB/		REAL	12	EST	50706P	/FRAMB/
1441B	/FRAMB/		INTEGER		MIZ	253B	/DADAT/
550P	/DADAT/		REAL	28	HRFL	65053B	/FRAMB/
20753B	/FRAMB/		REAL	992	HRFL1	65206B	/FRAMB/
370B	/DADAT/		REAL	112	HRJ	61062B	/FRAMB/
0B	/FRAMB/		REAL	90	HRJ1	63022B	/FRAMB/
132P	/FRAMB/		REAL	90	HX	1116P	/FRAMB/
253B	/DADAT2/		REAL	18	IBR	50711B	/FRAMB/
275P	/DADAT2/		REAL	60	IBR1	52635B	/FRAMB/
17013P	/FRAMB/		REAL	992	IBUG	133B	
423B	//		REAL	2	ICRIC	103B	//
1012B	/FRAMB/		INTEGER	60	ICRIS	75P	//
0P	/CONSTS/		REAL	10	IDFR	60432B	/FRAMB/
1B	/ALFA/		REAL		IDOC	60B	//

PROGRAM CVL53.

```

COMMON MAT,MFM,LFV,ISOLV,LVF IN,LNF IN,LVMAX,LMAX,ML,IFLFV,IFFRM
COMMON 7ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),MAXFL,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(18),IO9,CODE(2),MD(25),UF(20)
COMMON ITC1,NEZ(98),USF(10)
COMMON PON(20),LON(28)
COMMON TAKI(10),JET(10)
COMMON KET(20),LET(20)
DIMENSION NME(11,3)
INTEGER KAK(90,2)
COMMON/PLOT1/RSTIF(980,2),PLEN(980,2),MKR(980,2),KST(980,2),
1IBR(980,2),CSTIF(992,2),CLENI(992,2),CX(992,2),CZ(992,2),
2INMAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JBR(980,2)
COMMON/BATS/BATP(18)
COMMON/LOTO/LOTO(18)
COMMON/ALFA/ALFA,COSA,ID3
COMMON /BMFL1/WIDTH(30,20),DEPTH(30,20)
COMMON /BMFL2/SPAN(30,20),XPM(30,20,4),LBM(30,20,4),JMX(30)
COMMON /BMFL3/EYE(30,20),ILTNE(30)
COMMON/BMLN1/FC1,FY1
COMMON /RIGI/ IGI(30,20)
COMMON /BRACE/IBRA(30,20),LAR(5)
COMMON/CLFL1/FC5,FY5,CVR5,FKC(9)
COMMON/CLFL2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)
COMMON/CLFL3/X(140),Y(140),Z(140),XMAX(140),YMAX(140),KSC(140)
*,KDC(140)
COMMON/CLFL6/NOX,NEQ(140),MAK(140)
COMMON/CLFL5/ KBAR(140,2)
COMMON /CLFL7/ MCD(140,3),XY(9),STUB(9),IFO(9),XK(9),YK(9),INX(9)
COMMON/CLFL8/TOR(140,2),TORM(21)
COMMON/RIBS/IRIB,RIB(8)
COMMON/MOVE/MOVNO,XF(21,10),PM(21,10),MSTART(10,10),
.MEND(10,10),IREVE(18),NMH(10),PACT(18),MIT(10)
COMMON/ALS/ALFC(4),ALFY(4)
COMMON/BRCS/IBRC(18),EBRC(18)
COMMON/GEN1/ESTI(12,4),JYPE(30,2),KANT(30,2)
COMMON/CPROS/CPRO(140)
COMMON/CONSTS/CONSTS(18),IEOF,ACC(3)
COMMON/DYNIN/LYN(18),DYN(28)
COMMON/ALFAC/ALFAC(140)
COMMON/BMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFRM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(98),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140),
5,SIDE(18),DARG,IPLOT(25,15),JPL,IGRID1(140),
6,IGRID(140,2),DUMMY(240)
INTEGER COL1(140,2)
DIMENSION MB2(2),NC2(2),XARR(5),YARR(5)
COMMON/BAIND2/NSOP2,MIND2(91)
COMMON /BADATA2/LIND2(171),CJP(18),CLDAT(18,6)
COMMON/SPLICE/NSPL(140)
COMMON/REIZ1/KUZ,UZ1,UZ2,UZ3,UZ4,UZ5,UZ6,UZ7,UZ8,UZ9,UZ10,UZ11,
.UZ12,UZ13,UZ14,UZ15,UZ16,UZ17,UZ18
COMMON/THETFL/DUM(1898)
DIMENSION IFRB(258),YB1(258),MKB1(258),BFSH(21)
1,BFSH(30,20),OB2(258),BIN(240)

```

```

EQUIVALENCE (DUM(1),IFRB(1)),(DUM(251),XB1(1)),(DUM(501),YB1(1))
EQUIVALENCE (DUM(751),MK91(1))

```

```

EQUIVALENCE (DUM(1001),RMSH(1)),(DUM(1022),RIMSH(1,1))
EQUIVALENCE (DUM(1622),OP2(1)),(DUM(1872),INUMX)
EQUIVALENCE (DUM(1),RIM(1))

```

```

REAL MARG(10)

```

```

DATA MARG/5*2.,1.75,1.45,2.,125,2*2.5/

```

```

DATA SIDE/34.,44.,22.,34.,17.,22.,11.,17.,2.,5.,11./

```

```

C

```

```

C ***** INITIALIZE PLOT
IF (LET(12).EQ.0) GO TO 9998
IF (LET(12).EQ.-1) GO TO 6

```

```

C

```

```

CALL PLOTID(0,0,0,0,0,98)

```

```

CALL NEWPEN(1)

```

```

C

```

```

C ***** INITIALIZE VARIABLES

```

```

C

```

```

6 CONTINUE

```

```

IRUG=LET(11)

```

```

CALL ZERO(C11,4)

```

```

CALL ZERO(CLEV,90)

```

```

CALL ZERO(IFBB,1000)

```

```

ENDFILE 81

```

```

REWIND 81

```

```

KPL0T=LET(12)

```

```

IF (LET(12).EQ.-1) GO TO 8

```

```

DO 7 J=1,KPL0T

```

```

READ(81,750,END= 7)(IFLOT(I,J),I=1,25)

```

```

750 FORMAT(BZ,3I20)

```

```

7 CONTINUE

```

```

8 CONTINUE

```

```

DO 1 II=1,2

```

```

READ(81,800,END=10000) IW,IWMX,NB2(IW),NC2(IW),IFR2(IW),INWAY(IW),
1 LVM4, MB1(IW),LD,MSW

```

```

10000 NB=NB2(IW)

```

```

NC=NC2(IW)

```

```

IFR=IFR2(IW)

```

```

MB=MB1(IW)

```

```

800 FORMAT(BZ,6I10)

```

```

READ(81,800,END=10001) IWK(K,IW),K=1,NB), (MKC(K,IW),K=1,NB)
1 , (COL1(K,IW),K=1,IFR)

```

```

10001 READ(81,805,END=10002) (KST(K,IW),K=1,NB)

```

```

805 FORMAT(BZ,4I15)

```

```

10002 READ(81,810,END=10003) (BSTIF(K,IW),BLEN(K,IW),K=1,NB),
1 (CLENI(K,IW),CSTIF(K,IW),CX(K,IW),CZ(K,IW),PHIC(K,IW),DV

```

```

2(K,IW), K=1,NC), (TRANS(K,IW),K=1,LVM4)

```

```

810 FORMAT(BZ,6E12,5)

```

```

10003 READ(81,820,END=10004) (TBR(K,IW),JBR(K,IW),K=1,NB)

```

```

820 FORMAT(BZ,4I15)

```

```

10004 IF (IWMX.EQ.1) GO TO 2

```

```

1 CONTINUE

```

```

2 CONTINUE

```

```

IF (EOF(81).NE.0) GO TO 9999

```

```

DO 5 II=1,IWMX

```

```

MB=MB1(IW)

```

```

DO 4 LV=2,LVM4

```

```

      DO 3 I=1,MB
      LV1=(I-1)*LVM4+LV
      IF(IPUG.EQ.53)
      1PRINT 898,II,LV,I,IWMX,LVM4,NC,LV1,LEVEL,
      1CLEV(LV),CZ(LV1,II)
      898 FORMAT(* ** DEBUG1 ***,8I4,2F12.2)
      IF(CX(LV1,II).EQ.999..AND.CZ(LV1,II).EQ.999.)GO TO 3
      IF(CX(LV1,II).EQ.0..AND.CZ(LV1,II).EQ.0.)GO TO 3
      IF(CLEN1(LV1,II).EQ.0.)GO TO 3
      CLEV(LV)=CZ(LV1,II)
      LEVEL=LV
      3 CONTINUE
      4 CONTINUE
      5 CONTINUE

C
C ***** DETERMINE SCALE
C
      IF(IPUG.EQ.53)
      1PRINT 896
      896 FORMAT(* **DEBUG3 ** JOINT NO. (KA) LEVEL CX CZ *.
      1* CSTIF BSTIF*)
      STX=0.
      IF(IBUG.NE.53)GO TO 1020
      DO 1010 IW=1,IWMX
      MB=MB1(IW)
      DO 1010 KKK=1,MB
      DO 1010 LV=1,LVM4
      LLN=(KKK-1)*LVM4+LV
      CALL UNP53(KST(LLN,IW),A1,A2,B)
      IF(IBUG.EQ.53)
      1PRINT 895,LLN,KKK,LV,CX(LLN,IW),CZ(LLN,IW),CSTIF(LLN,IW),
      1BSTIF(LLN,IW)
      895 FORMAT(* ** DEBUG3 ***,3I6,2X,2F8.2,2E8.2)
      IF(IBUG.EQ.53)
      1PRINT 881,BLEN(LLN,IW),ALFA
      881 FORMAT(* ** DEBUG3 ** BLEN ALFA *,3F18.2)
      1010 CONTINUE
      1020 CONTINUE
      DO 1030 JJ=1,KPLOT
      IF(IPLOT(1,JJ).EQ.6)GO TO 1040
      IF(IPLOT(1,JJ).LE.3)GO TO 1040
      1030 CONTINUE
      SCALE=20.
      GO TO 1045
      1040 CONTINUE
      SCALE=IPLOT(20,JJ)
      IF(SCALE.EQ.0.)SCALE=20.
      1045 CONTINUE
      STZZ=0.
      STXX=STZZ*STXMM=STXMX=STZMM=STZMX=0.
      DO 20 IWI=1,IWMX
      IFR1=IFR2(IWI)
      MB=MB1(IWI)
      DO 20 NN=1,IFR1
      KA=COL1(NN,IWI)
      IF(NN.LT.IFR1)KB=COL1(NN+1,IWI)-1
      IF(NN.EQ.IFR1)KB=MB

```

```

      DO 20 ISI=KA,KP
      L=ISI+LVM4
      LL2=(ISI-1)+LVM4+1
      LL1=(ISI-1)+LVM4+(IDOS(3)+1)
      IF(CX(LL1,IWI).GT.STXX)STXX=CX(LL1,IWI)
      IF(CX(LL,IWI).EQ.999..AND.CZ(L,IWI).EQ.999.)GO TO 20
      IF(CZ(L,IWI).GT.STZZ)STZZ=CZ(L,IWI)
      IF(CZ(LL2,IWI).LE.STZZ1)STZZ1=CZ(LL2,IWI)
20  CONTINUE
      STXMA=-MARG(6)
      STZMN=STZZ1/SCALE-MARG(7)
      STYMX=STYX/SCALE
      STZMX=STZZ/SCALE+MARG(8)
      IF(IPUG.EQ.53)
      *PRINT 875,STXX,STZZ,STZZ1,STXMN,STZMN,STYMX,STZMX,STMX
875  FORMAT(' ** DEBUG LOCATION ** ',8F12.2)
      MBX=MB1(1)
      MBY=MB1(2)
      MAX=MAX0(MBX,MBY)
      IF(IPUG.EQ.53)
      1PRINT 888,MAX,MBX,MBY,MB1
888  FORMAT(' *** MB1 *** ',5I20)
C
C SET ALL STACK COORD TO -9999.
C
      DO 3000 IW=1,2
      I11=INVER(IW)
      MAXI=MB1(IW)
      DO 3000 N=1,MAXI
      MK=MKC(N,IW)
      IF(MK.LE.0)GO TO 3000
      CORD(MK,IW)=-9999.
      CORD(MK,I11)=-9999.
3000  CONTINUE
      MBX=MB1(1)
      IFRX=IFR2(1)
      KAX=COL1(1,1)
      IF(IFRX.GT.1)KBX=COL1(2,1)-1
      IF(IFRX.EQ.1)KBX=MBX
      BM=0.
      IW=1
      DO 3500 K=KAX,KBX
      LVNC=(K-1)+LVM4+(IDOS(3)+1)
      MK=MKC(K,1)
      CORD(MK,1)=CX(LVNC,1)
      CORD(MK,2)=BM
      IF(BLEN(LVNC,IW).LE.0.)GO TO 3500
      CALL UNP53(KST(LVNC,IW),A1,A2,B)
      BM=CORD(MK,2)+ALFA/BLEN(LVNC,IW)
      IF(IPUG.EQ.53)
      1PRINT 887,K,MK,LVNC,CORD(MK,1),CORD(MK,2),ALFA,BM
887  FORMAT(' ** DEBUG PL33 ** ',3I4,4F8.2)
3500  CONTINUE
      MBY=MB1(2)
      IFRY=IFR2(2)
      DO 3610 I=1,IFRY
      KA=COL1(I,2)

```

```

      IF(I.LT.IFRY)KB=COL1(I+1,2)-1
      IF(I.EQ.IFRY)KB=MBY
      CORDY=0.
      CORDY=0.
      DO 3600 K=KA,KB
      MK=MKC(K,2)
      DO 3510 KK=KAX,KBX
      LVNC=(KK-1)*LVM4+(IDOS(3)+X)
      MK1=MKC(KK,1)
      IF(MK.EQ.MK1)GO TO 3520
3510 CONTINUE
      GO TO 3600
3520 CONTINUE
      LVNC1=(K-1)*LVM4+(IDOS(3)+1)
      DO 3580 K1=KA,KB
      LVNC2=(K1-1)*LVM4+(IDOS(3)+1)
      MK=MKC(K1,2)
      IF(K.EQ.K1)GO TO 3570
      BM=0.
      IF(BLEN(LVNC1,2).LE.0.)GO TO 3540
      CALL UNP53(KST(LVNC1,2),A1,42,B)
      BM=ALFA/BLEN(LVNC1,2)
3540 CONTINUE
      CORD(MK,2)=CX(LVNC2,2)+CORDY-CX(LVNC1,2)
      CORD(MK,1)=CORDX-BM
      LVNC1=LVNC2
3570 CONTINUE
      CORDX=CORD(MK,1)
      CORDY=CORD(MK,2)
3580 CONTINUE
      GO TO 3610
3600 CONTINUE
3610 CONTINUE
      DO 3640 IW=1,2
      IW1=INVER(IW)
      IFR=IFR2(IW)
      DO 3630 IF=1,IFR
      KA=COL1(IF,IW)
      IF(IF.LT.IFR)KB=COL1(IF+1,IW)-1
      IF(IF.EQ.IFR)KB=MB1(IW)
      DO 3620 IK=KA,KB
      MK=MKC(IK,IW)
      IF(CORD(MK,IW1).NE.-9999.)GO TO 3630
3620 CONTINUE
3630 CONTINUE
3640 CONTINUE
      DO 3670 IW=1,2
      IW1=INVER(IW)
      IFR=IFR2(IW)
      DO 3670 IF=1,IFR
      KA=COL1(IF,IW)
      IF(IF.LT.IFR)KB=COL1(IF+1,IW)-1
      IF(IF.EQ.IFR.AND.IW.EQ.1)KB=MBX
      IF(IF.EQ.IFR.AND.IW.EQ.2)KB=MBY
      N1=0
      CORD1=0.
      NN2=0

```

```

DO 3662 IK=KA,KB
N=KB
MK=MKC(IK,IV)
3642 CONTINUE
IF(BUG.NE.53)GO TO 3643
PRINT 882,IV,IV1,IF,IFR,MP1(IV),IK,KA,KB,MK,N1,CORC(MK,IV1)
R82 FORMAT(' ** DEBUG PL1 **',10I4,F10.2)
3643 CONTINUE
IF(CORD(MK,IV1).NE.-9999.)GO TO 3645
IF(N1.NE.0.AND.IK.EQ.KB)GO TO 3650
IF(N1.NE.0)GO TO 3660
N1=IK
IF(IK.EQ.KB)GO TO 3642
GO TO 3660
3645 CONTINUE
IF(N1.NE.0)GO TO 3652
GO TO 3660
3650 CONTINUE
IF((N1-1).LT.KA)GO TO 3654
N=IK
N1=N1-1
MK1=MKC(KA,IV)
GO TO 3654
3652 CONTINUE
N=IK
LVNC=(N-1)*LVM4+(IDOS(3)+1)
CORD1=CX(LVNC,IV)
3654 CONTINUE
DO 3657 NN=N1,N
IF(NN2.EQ.KB)GO TO 3657
NN1=N+N1-NN
NN2=NN1-1
LVNC1=(NN2-1)*LVM4+(IDOS(3)+1)
ICORDI=1
IF(N.NE.KB)GO TO 3653
NN1=NN
NN2=NN1+1
LVNC1=(NN1-1)*LVM4+(IDOS(3)+1)
ICORDI=1
3653 CONTINUE
MK1=MKC(NN1,IV)
MK=MKC(NN2,IV)
IF(CORD(MK,IV).EQ.-9999..OR.CORD(MK,IV1).EQ.-9999.)GO TO 3667
IF(CORD(MK1,IV1).EQ.-9999..OR.CORD(MK1,IV1).EQ.-9999.)GO TO 3663
GO TO 3667
3663 CONTINUE
NN1=N+N1-NN
NN2=NN1-1
LVNC1=(NN2-1)*LVM4+(IDOS(3)+1)
MK1=MKC(NN1,IV)
MK=MKC(NN2,IV)
ICORDI=1
3667 CONTINUE
IF(CORD(MK1,IV).EQ.-9999..OR.CORD(MK1,IV1).EQ.-9999.)GO TO 3657
IO=1
IF(IV.NE.53)
1PRINT 884,IO,IV,IV1,IF,IK,NN,N1,N,KA,KB,MK,MK1,

```



```

1LVNC1,CORD1,CORD(MK,IV),CX(LVNC1,IV),CORD(MK1,IV)
  RM=0.
  IF(BLEN(LVNC1,IV).LE.0.)GO TO 3655
  CALL UNP53(KST(LVNC1,IV),A1,A2,P)
  RM=COSA/BLEN(LVNC1,IV)
3655 CONTINUE
  IF(CORD(MK,IV).NE.-9999.)GO TO 3656
  IF(ICORDI.EQ.0)
1CORD(MK,IV)=CORD(MK1,IV)+RM
  IF(ICORDI.EQ.1)
1CORD(MK,IV)=CORD(MK1,IV)-RM
  MK2=MKC(KA,IV)
  IF(CORD(MK2,IV).EQ.0..AND.N.NE.KB)
1CORD(MK,IV)=CX(LVNC1,IV)
3656 CONTINUE
  IF(CORD(MK,IV1).NE.-9999.)GO TO 3657
  IQ=2
  IF(IPUG.EQ.53)
1PRINT 884,IQ,IV,IV1,IF,IK,NN,N1,N,KA,KB,MK,MK1,
1LVNC1,CORD1,CORD(MK,IV1),CX(LVNC1,IV1),CORD(MK1,IV1)
  BM=0.
  IF(BLEN(LVNC1,IV).LE.0.)GO TO 3658
  CALL UNP53(KST(LVNC1,IV),A1,A2,P)
  BM=ALFA/BLEN(LVNC1,IV)
3658 CONTINUE
  IF(CORD(MK,IV1).NE.-9999.)GO TO 3659
  IF(ICORDI.EQ.0..AND.IV.EQ.2)
1CORD(MK,IV1)=CORD(MK1,IV1)-BM
  IF(ICORDI.EQ.0..AND.IV.EQ.1)
1CORD(MK,IV1)=CORD(MK1,IV1)+BM
  IF(ICORDI.EQ.1..AND.IV.EQ.1)
1CORD(MK,IV1)=CORD(MK1,IV1)-BM
  IF(ICORDI.EQ.1..AND.IV.EQ.2)
1CORD(MK,IV1)=CORD(MK1,IV1)+BM
  MK2=MKC(KA,IV)
  IF(CORD(MK2,IV1).EQ.0..AND.N.NE.KB)
1CORD(MK,IV1)=CX(LVNC1,IV1)
3659 CONTINUE
  IQ=3
  IF(IBUG.EQ.53)
1PRINT 884,IQ,IV,IV1,IF,IK,NN,N1,N,KA,KB,MK,MK1,
1LVNC1,BM,CORD(MK,IV1),CX(LVNC1,IV1),CORD(MK1,IV1),ALFA
884 FORMAT(' ** DEBUG PL2 ** ',13I4,3F10.2)
3657 CONTINUE
  N1=0
3660 CONTINUE
3662 CONTINUE
  IF(IBUG.NE.53)GO TO 3670
  DO 3665 IKK=KA,KB
  MK=MKC(IKK,IV)
  PRINT 880,IF,IV,MK,CORD(MK,IV),CORD(MK,IV1)
880 FORMAT(' ** DEBUG PL ** ',3I6,2F8.2)
3665 CONTINUE
3670 CONTINUE
3680 CONTINUE
  WRITE(16)CORD

```

```

      IF (LOTO(1).EQ.1.OR.LOTO(1).EQ.3) GO TO 9998
3685 CONTINUE
      DO 3690 JJ=1,KPLOT
      IF (IPLOT(1,JJ).EQ.6) GO TO 3694
      IF (IPLOT(1,JJ).LE.3) GO TO 3694.
3690 CONTINUE
      SCALE=20.
      GO TO 3696
3694 CONTINUE
      SCALE=IPLOT(20,JJ)
      IF (SCALE.EQ.0.) SCALE=20.
3696 CONTINUE
      STPX=STPX1=STPY=STPY1=0.
      STPXMN=STPYMN=STPXMX=STPYMX=0.
      DO 4000 IV=1,2
      MAX=MB1(IV)
      DO 4000 ISI=1,MAX
      MK=MKG(ISI,IV)
      IF (CORD(MK,1).LT.STPX) STPX=CORD(MK,1)
      IF (CORD(MK,1).GT.STPX1) STPX1=CORD(MK,1)
      IF (CORD(MK,2).LT.STPY) STPY=CORD(MK,2)
      IF (CORD(MK,2).GT.STPY1) STPY1=CORD(MK,2)
4000 CONTINUE
      STPYMN=STPY/SCALE
      STPXMN=STPX/SCALE-MARG(9)
      STPXMX=STPX1/SCALE
      STPYMX=STPY1/SCALE+MARG(9)
      STPIX=STPXMX-STPYMN
      STPIY=STPYMX-STPYMN
      STMX=AMAX1(STPIX,STPIY)*SCALE
      IFPL3=0
      WALL=BATP(8)
      CALL GRID5(LVM4)
      DO 5200 J=1,KPLOT
      JPL=J
      IFPL=0
      IFPL1=0
      IFPL2=0
      IFPL4=0
C
C IFPL NO. OF FRAMES/FLOORS TO BE PLOTTED
C IFPL1 0 WRITES ERROR MESSAGES FOR FRAMES
C IFPL2 0 WRITES ERROR MESSAGES FOR FLOORS
C IFPL3 COUNTS NO. OF PLOTS COMMANDS
C
      IF (IPLOT(1,J).LT.2) GO TO 5110
      IF (IPLOT(1,J).EQ.5.OR.IPLOT(1,J).EQ.7) GO TO 5200
      IF (IPLOT(1,J).EQ.6) GO TO 5110
      DO 5100 II=1,140
      CALL MMS070(1,IPLOT(2,J),II,IRI,IER)
      KKPLLOT(II)=IRI
      IF (IRI.EQ.1) IFPL=1+IFPL
5100 CONTINUE
      IF (IFPL.GT.0) GO TO 5115
      DO 5112 II=1,140
      KKPLLOT(II)=1
5112 CONTINUE

```

```

5115 CONTINUE
      IF(IPL0T(1,J).EQ.4)GO TO 5160
5110 CONTINUE
      DO 5000 IW=1,IWMX
        IFR=IFR2(IW)
        MB=MP1(IW)
        DO 10 N=1,IFR
          KA=COL1(N,IW)
          IF(N.LT.IFR)KB=COL1(N+1,IW)-1
          IF(N.EQ.IFR)KB=MB
          DO 10 I=KA,KB
            L=I+LVN4
            IF(CX(L,IW).EQ.999..AND.CZ(L,IW).EQ.999.)GO TO 10
            IF(CX(L,IW).GT.STX)STX=CX(L,IW)
            IF(CZ(L,IW).GT.STX)STX=CZ(L,IW)
10      CONTINUE
          SCALE=IPL0T(20,J)
          IF(SCALE.EQ.0.)SCALE=20.
          IDIR=1HX
          IF(INWAY(IW).EQ.2)IDIR=1HY
          IF(INWAY(IW).EQ.2.AND.IPL0T(1,J).EQ.2)GO TO 5000
          IF(INWAY(IW).EQ.1.AND.IPL0T(1,J).EQ.3)GO TO 5000
          IF(IBUG.EQ.53)
            1PRINT 899,IDIR
          899 FORMAT(' ** DEBUG FRAME IN ',A2,' DIRECTION ****')
          ORGX=ORGY=0.
          IFPL3=IFPL3+1
          ICOUNT=0
          MCOUNT=0
          IF(IFPL.EQ.0)IFPL=IFR
          IFPL4=IFPL
          DO 2000 K=1,IFR
            KA=COL1(K,IW)
            MK=MKC(KA,IW)
            IQ=1
            IF(IBUG.EQ.53)
              1PRINT 873,IQ,IW,INWAY(IW),J,KPLOT,IFPL,MK,IPL0T(1,J),(KKPLOT(K),
              2(K=1,20),IFPL3,ICOUNT,MCOUNT,IFPL4
          873 FORMAT(' *** PLOT-1 ****,32I3)
            IF(K.LT.IFR)KB=COL1(K+1,IW)-1
            IF(K.EQ.IFR)KB=MB
            IF(IPL0T(1,J).EQ.1.OR.IPL0T(1,J).EQ.6)GO TO 2015
            IF(KKPLOT(MK).EQ.0)GO TO 2000
2015      CONTINUE
            IFPL1=1
            ICOUNT=ICOUNT+1
            IPLA=0
            ORGX3=ORGX1
            IF(IFPL3.EQ.2)ORGY3=ORGY1
            IF(IPS.EQ.0)IPS=IPL0T(21,J)
            IF(IPS.EQ.0)IPS=3
            CALL FRAME(IPLA,ICOUNT,MCOUNT,KA,KB,IW,LVN4,STZMW,CRGX,ORGY,
            *SCALE,ORGX1,ORGY1,IPS)
            IF(IFPL3.EQ.1)ORGY3=ORGY1
            IQ=3
            IF(IBUG.EQ.53)PRINT 873,IQ,J,MK,IW,IPL0T(1,J)
            IF(IBUG.EQ.53)PRINT 806,ORGX3,ORGY3

```

```

      IF (MCOUNT.NE.999) GO TO 2020
      PRINT 877
877 FORMAT(' ** MESSAGE PLOT-1 *** FRAMING ELEV. NEEDS*,
1* LARGER PAGE ****)
      CALL REMARK(' DRAWING SIZE*)
      CALL REMARK(' TOO SMALL*)
      STOP
2020 CONTINUE
      IF (MCOUNT.EQ.0.AND.IFPL3.EQ.1) GO TO 2023
      IF (MCOUNT.EQ.0) GO TO 2022
      ORGX3=ORGX1
      ORGY3=ORGY1
2022 CONTINUE
      IF (IFPL3.GE.2.AND.MCOUNT.EQ.0) ORGX3=0.
      IF (IFPL3.GE.2.AND.MCOUNT.EQ.0) ORGY3=0.
      IF (IBUG.EQ.53) PRINT 806, ORGX3, ORGY3, ORGY1
806 FORMAT(' LOCATION 3*, 3F8.2)
      IF (ICOUNT.GT.1.AND.MCOUNT.EQ.0) GO TO 2021
      CALL PLOT(ORGX3, ORGY3, -3)
      CALL PLOT(1., 0., -3)
2023 CONTINUE
      IF (MCOUNT.EQ.0.AND.ICOUNT.GT.1) GO TO 2021
      SIDEY=SIDE(IPS+1)
      SIDEY=SIDE(IPS)
      CALL NEWPEN(11)
      CALL BORDER(0., 0., SIDEY, SIDEY, -3, -3, 2., 03)
      CALL NEWPEN(1)
      CALL PLOT(1.5, -.5, 3)
      XARR(1)=-.5
      XARR(2)=-.5
      YARR(1)=-.5
      YARR(2)=SIDEY+.5
      XARR(3)=SIDEY+.5
      YARR(3)=YARR(2)
      XARR(4)=XARR(3)
      YARR(4)=YARR(1)
      XARR(5)=XARR(1)
      YARR(5)=YARR(1)
      CALL DASH(XARR, YARR, 5, 1., .5)
      IFPL4=IFPL4-1
      IFPL5=IFPL-IFPL4
      IS=1
      IM=2
      IF (IPLT(22, J).EQ.8) IM=3
      IF (IPLT(22, J).EQ.7) IM=1
      CALL LOGO5(IM, IS, MK, IM, IFPL5, IPS, ORGX, ORGY, SCALE)
2021 CONTINUE
      IF (IBUG.EQ.53) PRINT 887, ICOUNT, MCOUNT, K, ORGX1, ORGY1
887 FORMAT(' ** LOCATION 2 *** 3I4, 4F8.2)
      CALL PLOT(ORGX, ORGY, -3)
      IF (MCOUNT.EQ.998)
      *CALL FRAME(IPLA, ICOUNT, MCOUNT, KA, KB, IM, LVH4, STZHN, ORGX, ORGY,
      *SCALE, ORGX1, ORGY1, IPS)
      CALL PLOT(STXHN, STZHN, 3)
      CALL SYMBOL(STXHN, STZHN, .01, 3, 0., -1)
      CALL PLOT(STXHX, STZHX, 3)
      CALL SYMBOL(STXHX, STZHX, .01, 3, 0., -1)

```

```

      IF(IPUG.EQ.53)
        1PRINT 990
      990 FORMAT(* ** DEBUG ** TOTAL NO. FR FRAME NO. ST(KA)*,
        1* ST(KB) ST(MB) MAXLEV MAXLEVP*)
      IF(1BUG.EQ.53)
        1PRINT 991,IFR,K,KA,KB,MB,LVM4,LEVEL
      991 FORMAT(* ** DEBUG***,I5,X,I4,7X,I3,5X,I3,4X,I6,7X,I4,6X,I3)
      CALL NEVPEN(11)
      DO 1000 I=KA,KB
        LN1=(I-1)*LVM4+1
        LN2=I*LVM4
        IF(IPUG.EQ.53)
          1PRINT 992
      992 FORMAT(* ** DEBUG ** STAC N2. LAST JOINT FIRST JOINT COOR.,
        1* JOINT NO. CX CZ CLENI *)
      IF(IPUG.EQ.53)
        1PRINT 993,I,LN2,LN1,LN2,CX(LN2,IW),CZ(LN2,IW),CLENI(LN2,IW)
      993 FORMAT(* ** DEBUG ***,I5,5X,I5,8X,I6,7X,I7,F12.2,3F8.2)
      50 IF(LN2.LE.LN1)GO TO 900
      IITYPE=4H
      C
      C
      C
      IF(CX(LN2,IW).EQ.999..AND.CZ(LN2,IW).EQ.999.)GO TO 160
      IF(CLENI(LN2,IW).EQ.0.)GO TO 160
      BM=0.
      COL=0.
      CX1=CX(LN2,IW)
      CZ1=CZ(LN2,IW)
      C11(1)=CX1/SCALE
      C22(1)=CZ1/SCALE
      IF(I.EQ.KB)GO TO 70
      IF(BLEN(LN2,IW).EQ.0.)GO TO 70
      CALL UNP53(KST(LN2,IW),A1,A2,B)
      BM=(1.-ALFA*ALFA)**.5/BLEN(LN2,IW)
      70 COL=1./CLENI(LN2,IW)
      CALL PLOT(CX1/SCALE,CZ1/SCALE,3)
      CX2=CX1*BM
      CZ2=CZ1-COL
      C11(2)=CX1/SCALE
      C22(2)=CZ2/SCALE
      IF(CSTIF(LN2,IW).LE.1.E-10.OR.COL.EQ..9)GO TO 100
      CALL PLOT(CX1/SCALE,CZ2/SCALE,2)
      IITYPE=4HLINE
      IF(1BUG.EQ.53)
        1PRINT 897,LN2,I,IW,CX1,CZ1,CX2,CZ2
      897 FORMAT(* ** DEBUG2 ***,3I4,4F12.2)
      GO TO 120
      100 CONTINUE
      IF(1PLOT(9,J).EQ.0)GO TO 120
      CALL DASH(C11,C22,2,.1,.2)
      CALL PLOT(CX1/SCALE,CZ1/SCALE,3)
      IITYPE=4HDASH
      120 CONTINUE
      IF(COL.EQ..9)GO TO 900
      GO TO 160
      160 CONTINUE

```

```

      IF(I.EQ.KA.AND.IBUG.EQ.53)PRINT 994
994  FORMAT(' ** DEBUG COLS ** FROM JOINT  CX      CZ      IC JOINT=
1)  CX      CZ      CLEN1      COLTYP  ')
      IF(IBUG.EQ.53)
1PRINT 995, LN2, CX(LN2, IW), CZ(LN2, IW), (LN2-1), CX1, CZ2,
1CLEN1(LN2, IW), IITYPE
995  FORMAT(' ** DEBUG COLS **', I4, F9.2, F8.2, 8X, I4, F9.2, F8.2, F9.2,
1(4Y, A4))
      LN2=LN2-1
      GO TO 50
900  CONTINUE
      LN2=LN2+1
      IF(IPL0T(9, J).EQ.0.AND.CSTIF(LN2, IW).LE.1.E-10)GO TO 1000
      IF(LN2.LT.LN1)GO TO 1000
      C1=(CX(LN2, IW)-.5)/SCALE
      C2=CZ(LN2, IW)/SCALE
      C3=C1+1./SCALE
      CALL PLOT(C1, C2, 3)
      CALL PLOT(C3, C2, 2)
      IF(COL.F0..9)LN2=LN2-1
      IF(COL.F0..9.AND.LN2.GE.LN1)GO TO 50
1000 CONTINUE
      DO 1200 LV2=2, LEVEL
      LN2=(KA-1)*LVM4+LV2
      LN3=(KB-1)*LVM4+LV2
      IF(IBUG.EQ.53)
1PRINT 890
890  FORMAT(' ** DEBUG ** LEVE NO. LAST JOINT  FIRST JOINT  COOR.**
1*JOINT NO.  CX      CZ      BLEN  ')
      IF(IBUG.EQ.53)
1PRINT 891, LV2, LN3, LN2, LN2, CX(LN2, IW), CZ(LN2, IW), BLEN(LN2, IW)
891  FORMAT(' ** DEBUG **', I5, 5X, I5, 8X, I6, 7X, I7, F12.2, 3F8.2)
1050 IF(LN2.GE.LN3)GO TO 1200
      IITYPE=4H
      LN4=LN2+LVM4
      IF(CX(LN2, IW).EQ.999..AND.CZ(LN2, IW).EQ.999.)GO TO 1160
      IF(BLEN(LN2, IW).GT.0.)GO TO 1060
      IF(CX(LN4, IW).EQ.999..AND.CZ(LN4, IW).EQ.999.)GO TO 1160
1060 CONTINUE
      CX1=CX(LN2, IW)/SCALE
      CZ1=CZ(LN2, IW)/SCALE
      C11(1)=CX1
      C22(1)=CZ1
      BM=0.
      GAP1=0.
      GAP2=0.
      IF(BLEN(LN2, IW).EQ.0.)GO TO 1070
      CALL UNP53(KST(LN2, IW), A1, A2, B)
      BM=(1.-ALFA*ALFA)**.5/BLEN(LN2, IW)
      K1=0
      K2=0
      IF(IPL0T(5, J).EQ.0)GO TO 1070
      IF(A1.GT..05)K1=1
      IF(A2.GT..05)K2=1
      IF(K1.EQ.0)GAP1=.083
      IF(K2.EQ.0)GAP2=.083
      CX1=CX1+GAP1

```

```

1070 CALL PLOT(CX1,CZ1,3)
      IF(IRUG.EQ.53)
        PRINT 874,J,LN2,K1,K2,IPLT(5,J),GAP1,GAP2,CX1,CX2
874  FORMAT(' 10GAP',5I4,4F10.3)
      CX2=CX1+BM/SCALE-(GAP1+GAP2)
      CZ2=CZ1
      IF(MAT.EQ.1)GO TO 1170
      IF(BSTIF(LN2,IV).LE.1.E-15)GO TO 1150

```

```

C
C  DRAW A REAL BEAM
C
      IF(BSTIF(LN2,IV).EQ.WALL)GO TO 1150
      IITYPE=4HLINE
      CALL PLOT(CX2,CZ2,2)
      GO TO 1160

```

```

C
C  DRAW A DUMMY BEAM OR A WALL
C

```

```

1150 CONTINUE
      C11(2)=CX2
      C22(2)=CZ1
      IF(IPLT(9,J).EQ.0.AND.BSTIF(LN2,IV).LE.1.E-15)GO TO 1160
      CALL DASH(C11,C22,2,.1,.2)
      IITYPE=4HDASH

```

```

1160 CONTINUE
      IF(LV2.EQ.2.AND.IRUG.EQ.53)PRINT #92
#92  FORMAT(' ** DEBUG BEAM ** FROM JOINT   CX       CZ       TO JOINT'
1,*,   CX       CZ       BLEN   BNTYP*)
      LN4=LN2+LV*4
      IF(IRUG.EQ.53)
        *PRINT 893,LN2,CX(LN2,IV),CZ(LN2,IV),LN4,CX(LN4,IV),CZ(LN4,IV),
        *BM,IITYPE

```

```

893  FORMAT(' ** DEBUG BEAM **',I4,F9.2,F8.2,8X,I4,F9.2,F8.2,F9.2,
*4X,A4)
      LN2=LN4
      GO TO 1050

```

```

1170 CONTINUE

```

```

C
C  DRAW A REAL BEAM

```

```

      IF(BSTIF(LN2,IV).EQ.WALL)GO TO 1180
      IF(IV.EQ.1)NTEM=MOD(MKB(LN2,IV),10000)
      IF(IV.EQ.2)NTEM=MKB(LN2,IV)/10000
      NLI=NTEM/100
      NSP=NTEM-NLI*100
      IF(WIDTH(NLI,NSP).EQ.4..AND.DEPTH(NLI,NSP).EQ.4.)GO TO 1180
      IITYPE=4HLINE
      CALL PLOT(CX2,CZ2,2)
      GO TO 1182

```

```

C
C  DRAW A DUMMY BEAM OR A WALL
C

```

```

1180 CONTINUE
      C11(2)=CX2
      C22(2)=CZ1
      IF(IPLT(9,J).EQ.0.AND.BSTIF(LN2,IV).LE.1.E-15)GO TO 1182
      CALL DASH(C11,C22,2,.1,.2)

```

```

      IITYPE=AMDASH
1182 CONTINUE
      IF (LV2.EQ.2.AND.IBUG.EQ.53) PRINT 892
      LN4=LN2+LV*4
      IF (IPUG.EQ.53)
        *PRINT 893, LN2, CX(LN2, IW), CZ(LN2, IW), LN4, CX(LN4, IW), CZ(LN4, IW),
        *BM, IITYPE
      LN2=LN4
      GO TO 1050
C
1200 CONTINUE
C
C ***** BRACING PLOT *****
C
      IF (PSW.EQ.1) GO TO 520
C
C UNPACK BRACING INFORMATION
C
      IF (IFUG.EQ.53)
        PRINT 996
996 FORMAT(' ** DEEUG BRACING INFORMATION**')
      KB1=KB-1
      DO 498 I=KA, KB1
      DO 390 LV=2, LV*4
      LN2=(I-1)*LV*4+LV
      IF (IBR(LN2, IW).LT.10) GO TO 380
      JR=JBR(LN2, IW)/(10**5)
      JL=JBR(LN2, IW)-JR*10**5
      IC=MOD(IBR(LN2, IW)/10, 10)
      DO 300 II=1, KB
      LN1=LV*4+II+LN2
      IF (CX(LN1, IW).EQ.999.AND.CZ(LN1, IW).EQ.999) GO TO 300
      C1X=CX(LN1, IW)/SCALE
      C1Z=CZ(LN1, IW)/SCALE
      GO TO 310
300 CONTINUE
310 CONTINUE
      GO TO (320, 330, 320, 340, 350), IC
      CALL GOTOER
C
C RBR TYPE 1 AND RIGHT BRACE OF TYPE 3(XBR)
C
320 CONTINUE
      C2X=CX(JR, IW)/SCALE
      C2Z=CZ(JR, IW)/SCALE
      IF (BSTIF(LN2, IW).EQ.WALL.AND.IC.EQ.3) GO TO 322
      CALL PLOT(C1X, C1Z, 3)
      CALL PLOT(C2X, C2Z, 2)
      IF (IC.EQ.1) GO TO 380
      GO TO 330
322 CONTINUE
      C11(1)=C1X
      C11(2)=C2X
      C22(1)=C1Z
      C22(2)=C2Z
      CALL DASH(C11, C22, 2, 1, 2)
330 CONTINUE

```


C
C LBR TYPE 2 AND LEFT BRACE OF TYPE 3 (XBR)

C
C1X=CX(LN2,IW)/SCALE
C1Z=CZ(LN2,IW)/SCALE
C2X=CX(JL,IW)/SCALE
C2Z=CZ(JL,IW)/SCALE
IF(RSTIF(LN2,IW).EQ.WALL.AND.IC.EQ.3)GO TO 332
CALL PLOT(C1X,C1Z,3)
CALL PLOT(C2X,C2Z,2)
GO TO 340
332 CONTINUE
C11(1)=C1X
C11(2)=C2X
C22(1)=C1Z
C22(2)=C2Z
CALL DASH(C11,C22,2,.1,.2)
GO TO 340
340 CONTINUE

C
C ABR TYPE 4

C
C1=(C1X*SCALE+CX(LN2,IW))/SCALE*2.
C1Z=CZ(LN2,IW)/SCALE
C2X=CX(JL,IW)/SCALE
C2Z=CZ(JL,IW)/SCALE
C3X=CX(JR,IW)/SCALE
C3Z=CZ(JR,IW)/SCALE
CALL PLOT(C1,C1Z,3)
CALL PLOT(C2X,C2Z,2)
CALL PLOT(C1,C1Z,3)
CALL PLOT(C3X,C3Z,2)
GO TO 350
350 CONTINUE

C
C VBR TYPE 5

C
C1=(CX(JL,IW)+CX(JR,IW))*5/SCALE
C2=CZ(JL,IW)/SCALE
C2X=CX(LN2,IW)/SCALE
C2Z=CZ(LN2,IW)/SCALE
CALL PLOT(C1,C1Z,3)
CALL PLOT(C1,C2,2)
CALL PLOT(C2X,C2Z,3)
CALL PLOT(C1,C2,2)
380 CONTINUE
IF(LY.EQ.2.AND.IBUE.EQ.53)PRINT 997
997 FORMAT(' ** DEBUG ** TYPE LN1 CX CZ JR',
1' CX CZ')
IF(IBUE.NE.53)GO TO 383
IF(IC.EQ.1.OR.IC.EQ.3)PRINT 998,IC,LN1,CX(LN1,IW),CZ(LN1,IW),JR,
1CX(JR,IW),CZ(JR,IW)
998 FORMAT(' ** DEBUG ** ',I2,3X,I4,2F8.2,2X,I4,2F8.2)
383 CONTINUE
IF(IBUE.EQ.53)GO TO 385
IF(IC.EQ.2.OR.IC.EQ.3)PRINT 998,IC,LN1,CX(JL,IW),CZ(JL,IW),LN2,
1CX(LN2,IW),CZ(LN2,IW)

385 CONTINUE

IF(IC.EQ.4.AND.IBUG.EQ.5)PPINT 998,IC,JL,CX(JL,IW),CZ(JL,IW),JR,
ICX(JF,IW),CZ(JR,IW)IF(IC.EQ.5.AND.IBUG.EQ.5)RRINT 998,IC,JL,CX(JL,IW),CZ(JL,IW),LN2,
CZ(LN2,IW),CZ(LN2,IW)

390 CONTINUE

400 CONTINUE

420 KB1=KB-1

C

C

WRITE FEAM MARKS

C

CALL NEWPEN(1)

CALL ZERO(DUM,1890)

DO 5PC LV=2,LVM4

LINE=0

DO 550 I=KA,KB1

LN2=(I-1)*LVM4+LV

IF(CX(LN2,IW).EQ.999..AND.CZ(LN2,IW).EQ.999.)GO TO 550

IF(LEN(LN2,IW).LE.0.)GO TO 550

C1X=CX(LN2,IW)/SCALE

C1Z=CZ(LN2,IW)/SCALE

DO 530 II=1,KB

LN1=II*LVM4+LN2

IF(CX(LN1,IW).EQ.999..AND.CZ(LN1,IW).EQ.999.)GO TO 530

C2X=CX(LN1,IW)/SCALE

C2Z=CZ(LN1,IW)/SCALE

GO TO 540

530 CONTINUE

540 CONTINUE

C1=(C1X+C2X)*.5

C2=C1Z

IF(IW.EQ.1)MARK=MOD(MKB(LN2,IW),10000)

IF(IW.EQ.2)MARK=MKB(LN2,IW)/10000

NLI=MARK/100

NSP=MARK-NLI*100

IF(IPILOT(18,J).EQ.0.AND.IPILOT(19,J).EQ.0)GO TO 535

IF(MIND2(LV).LE.0) GO TO 535

CALL STINDX(12,MIND2,91,0)

CALL READMS(12,LIND2,171,LV-1)

CALL STINDX(12,LIND2,171,0)

IF(LIND2(NLI+1).LE.0) GO TO 535

IF(NLI.EQ.LINE)GO TO 541

CALL READMS(12,BMSH,21,NLI)

LINE=NLI

541 CONTINUE

BMSH(NLI,NSP)=BMSH(NSP)

535 CONTINUE

K1=K2=0

CALL UNP53(KST(LN2,IW),A1,A2,B1)

IF(A1.GT..05)K1=1

IF(A2.GT..05)K2=1

IF(MAT.EQ.1)GO TO 548

IF(IPILOT(18,J).EQ.0)GO TO 542

CALL SIZE5(4,J,IW,C1,C2,LN2,SCALE)

542 CONTINUE

IF(IPILOT(9,J).EQ.0.AND.BSTIE(LN2,IW).LE.1.E-15)GO TO 550

IF(IPILOT(5,J).EQ.0)GO TO 545

```

      IF(K1.EQ.0)GO TO 545
      CALL SYMB5(1,K1,SCALE,C1X,C1Z,LN2,IV)
545 CONTINUE
      IF(IPL0T(14,J).EQ.0)GO TO 547
      CALL LABEL5(1,J,C1,C2,KA,KB,LVM4,IV,LN2,SCALE)
547 CONTINUE
      IF(IPL0T(5,J).EQ.0)GO TO 550
      IF(K2.EQ.0)GO TO 550
      CALL SYMB5(2,K2,SCALE,C2X,C2Z,LN2,IV)
      GO TO 550
548 CONTINUE
C
C MARKS AND SYMBOLS FOR BEAMS (CONCRETE STRUCTURES)
C
      WID=WIDTH(NLI,NSP)
      DEP=DEPTH(NLI,NSP)
      IF(MAT.EQ.1)GO TO 546
      IF(IPL0T(18,J).EQ.0)GO TO 546
      IF(I.6T.MA)GO TO 543
      CALL ENCO(30)
543 CONTINUE
      CALL SIZE5(4,J,IV,C1,C2,LN2,SCALE)
546 CONTINUE
      IF(IPL0T(9,J).EQ.1)GO TO 549
      IF(WID.EQ.4..AND.DEP.EQ.4.)GO TO 550
549 CONTINUE
      IF(IPL0T(5,J).EQ.0)GO TO 552
      IF(K1.EQ.0)GO TO 552
      IF(MAT.EQ.1)GO TO 552
      CALL SYMB5(1,K1,SCALE,C1X,C1Z,LN2,IV)
552 CONTINUE
      IF(IPL0T(14,J).EQ.0)GO TO 554
      CALL LABEL5(1,J,C1,C2,KA,KB,LVM4,IV,LN2,SCALE)
554 CONTINUE
      IF(IPL0T(5,J).EQ.0)GO TO 550
      IF(K2.EQ.0)GO TO 550
      IF(MAT.EQ.1)GO TO 550
      CALL SYMB5(2,K2,SCALE,C2X,C2Z,LN2,IV)
550 CONTINUE
580 CONTINUE
      IF(MAT.EQ.1)GO TO 590
      IF(IPL0T(17,J).EQ.0)GO TO 590
      CA=KA
      CB=KB
      CALL SIZE5(3,J,IV,CA,CB,LVM4,SCALE)
590 CONTINUE
      IF(IPL0T(8,J).EQ.0)GO TO 595
      CALL DIM5(1,J,KA,KB,IV,LVM4,LV,SCALE,STZMX)
595 CONTINUE
C
C LEVEL NUMBERS AND COLUMN MARKS *****
      CALL LABEL5(2,J,C1,C2,KA,KB,LVM4,IV,LN2,SCALE)
2000 CONTINUE
      IF(IFPL1.EQ.0)GO TO 4305
      CALL PLOT(ORGX1,ORGY1,-3)
5000 CONTINUE
5160 CONTINUE

```

```

      IF (IRUG.EQ.53)
        1PRINT A'2,IPL0T(1,J)
      802 FORMAT(' DEBUG PLOT ***.14)
      IF (IPL0T(1,J).EQ.4) GO TO 4012
      IF (IPL0T(1,J).EQ.6) GO TO 4012
      GO TO 5200
    4012 CONTINUE
      SCALE=IPL0T(20,J)
      IF (SCALE.EQ.0.) SCALE=20.
      CALL ZERO(ICFL,140)
      ORG=ORGY=0.
      ICOUNT=0
      MCOUNT=0
      IFPL3=IFPL3+1
      REWIND LVU
      IF (IFPL3.EQ.1) IFPL4=IFPL
      DO 4300 LV3=2,LEVEL
      CALL ZERO(IFLAG1,200)
      LV2=LEVEL-LV3+2
      IF (IRUG.EQ.53.AND.LV3.EQ.2) PRINT A'5
    885 FORMAT(' ** DEBUG PL3 ** LEV LVMX X/Y FRTOT FR.NO. STNB.
      1. ST1 ST2 MKC LVNC FROM TO BN CX ')
      IF (IFPL.EQ.0) GO TO 4015
      IQ=2
      IF (IRUG.EQ.53)
        4PRINT A'73,IQ,IN,INWAY(IN),J,KPLOT,IFPL,KA,IPL0T(1,J),(KKPLOT(IK),
        2JK=1,20)
    4015 CONTINUE
      IF (ACC(3).EQ.2) GO TO 4011
      WRITE(16) IPL0T,JPL,MB1,IFR2,COL1,ICFL,IFRM,
      1NCOL,MKC,IBUG,IFLAG1,IFLAG,KKPLOT,IGRID1,IFRB,MKB1,INUMX
      2,STPX,STPX1,STPY,STPY1,CORD,GRID1,CLEV,
      1STZZ1,STZZ,SIDE,MARG,DUMMY,XB1,YB1,BMSH,BIMSH,OB2,PIM
    4011 CONTINUE
      IFPL2=1
      CALL R53(1,LVU,LIG,JIG,NIG,LVR)
      IF (ACC(3).EQ.2) GO TO 5325
      BACKSPACE (16)
      READ(16) IPL0T,JPL,MB1,IFR2,COL1,ICFL,IFRM,
      1NCOL,MKC,IBUG,IFLAG1,IFLAG,KKPLOT,IGRID1,IFRB,MKB1,INUMX
      2,STPX,STPX1,STPY,STPY1,CORD,GRID1,CLEV,
      1STZZ1,STZZ,SIDE,MARG,DUMMY,XB1,YB1,BMSH,BIMSH,OB2,PIM
      DO 865 IN=1,2
      NCOL1=NCOL(IN)
      DO 865 INN1=1,NCOL1
      IF (GRID1(INN1,IN).GE.77777.) GRID1(INN1,IN)=77777.777
    865 CONTINUE
    5325 CONTINUE
      IF (MAT.EQ.1) GO TO 4014
      IF (IPL0T(18,J).EQ.0.AND.IPL0T(19,J).EQ.0) GO TO 4014
      CALL STINDX(12,MIND2,91,0)
      IF (MIND2(LVR+1).LE.0) GO TO 4014
      CALL READMS(12,LIND2,171,LVR)
      CALL STINDX(12,LIND2,171,0)
      IF (LIND2(LMM+1).LE.0) GO TO 4014
      DO 5300 LMM=1,LIG
      CALL READMS(12,BMSH,21,LMM)

```

```

DO 5250 JMM=1,J16
BIMSP(LMM,JMM)=BIMSH(JMM)
IF(IPUG.EQ.53)
1PRINT 5330,LMM,JMM,LIND2(LMM+1),BIMSP(LMM,JMM)
5330 FORMAT(' POSIT IC',2I10,125,1X,A15)
5250 CONTINUE
5300 CONTINUE
4014 CONTINUE
DO 4013 IW=1,2
MB=MB1(IW)
IFR=IFR2(IW)
IW1=INVER(IW)
DO 4013 I=1,IFR
KA=COL1(I,IW)
IF(I.LT.IFR)KB=COL1(I+1,IW)-1
IF(I.EQ.IFR)KB=MP
DO 4013 K=KA,KB
MK=MKC(K,IW)
IF(KSC(MK).EQ.0)GO TO 4013
ICFL(MK)=1
4013 CONTINUE
CALL ZERO(IFBB,1000)
IF(KKPL0T(LVR).EQ.0.AND.IPL0T(1,J).NE.6)GO TO 4300
IQ=3
IF(IRUG.EQ.53)
*PRINT A73,IQ,J,LV3,LEVEL,LV2,LVR
ICOUNT=ICOUNT+1
IPLA=1
ORGX3=ORGY1
IF(IFPL3.GE.2)ORGY3=ORGY1
IF(IPS.EQ.0)IPS=IPL0T(21,J)
IF(IPS.EQ.0)IPS=3
CALL FRAME(IPLA,ICOUNT,MCOUNT,LV2,KB,IW,LVM4,STPYMN,ORGX,ORGY,
*SCALE,ORGX1,ORGY1,IPS)
IF(IFPL3.EQ.1)ORGY3=ORGY1
IF(MCOUNT.NE.999)GO TO 4016
PRINT 878
878 FORMAT(' ** MESSAGE PLOT-2' *** FRAMING PLAN NEEDS',
1' LARGER PAGE ****)
CALL REMARK(' DRAWING SIZE')
CALL REMARK(' TOO SMALL')
STOP
4016 CONTINUE
IF(MCOUNT.EQ.0.AND.IFPL3.EQ.1)GO TO 4017
IF(MCOUNT.EQ.0)GO TO 4022
ORGX3=ORGX1
ORGY3=ORGY1
C
4022 CONTINUE
IF(IFPL3.GE.2.AND.MCOUNT.EQ.0)ORGX3=0.
IF(IFPL3.GE.2.AND.MCOUNT.EQ.0)ORGY3=0.
IF(ICOUNT.GT.1.AND.MCOUNT.EQ.0)GO TO 4019
CALL PLOT(ORGX3,ORGY3,-3)
CALL PLOT(1,0,-3)
4017 CONTINUE
IF(MCOUNT.EQ.0.AND.ICOUNT.GT.1)GO TO 4019
SIDEY=SIDE(IPS+1)
SIDEY=SIDE(IPS)

```

```

CALL NEWPEN(11)
CALL BORDER(0,0,SIDEX,SIDEY,43,-3,2,03)
CALL NEWPEN(1)
CALL PLOT(0.5,ORGY,3)
XARR(1)=-.5
YARR(2)=-.5
YARR(1)=-.5
YARR(2)=SIDE(IPS)+.5
XARR(3)=SIDE(IPS+1)+.5
YARR(3)=YARR(2)
XARR(4)=XARR(3)
YARR(4)=YARR(1)
XARR(5)=YARR(1)
YARR(5)=YARR(1)
CALL DASH(XARR,YARR,5,1,.5)
IF(1BUG.EQ.5)PRINT 807,ICOUNT,MCOUNT,K,ORGX1,ORGY1
IFPL4=IFPL4-1
IFPL5=IFPL-IFPL4
IS=2
IM=2
IF(IPL0T(22,J).EQ.8)IM=3
IF(IPL0T(22,J).EQ.7)IM=1
CALL LOGO5(IM,IS,LVR,IM,IFPL5,IPS,ORGX,ORGY,SCALE)
401= CONTINUE
CALL PLOT(ORGX,ORGY,-3)
IF(MCOUNT.EQ.998)
*CALL FRAME(IPLA,ICOUNT,MCOUNT,LV2,KP,IV,LVM4,STPYMN,ORGX,ORGY,
*SCALE,ORX1,ORGY1,IPS)
CALL PLOT(STPYMN,STPYMN,3)
CALL SYMBOL(STPYMN,STPYMN,.01,3,0,-1)
CALL PLOT(STPYMX,STPYMX,3)
CALL SYMBOL(STPYMX,STPYMX,.01,3,0,-1)

C
C
IF(LVR.NE.(LV2-1))STOP
CALL NEWPEN(11)
IF(MAT.EQ.1)60 TO 4295
DO 4250 IW=1,2
MB=MB1(IW)
IFR=IFR2(IW)
IW1=INVER(IW)
DO 4250 I=1,IFR
KA=COL1(I,IW)
IF(I.LT.IFR)KB=COL1(I+1,IW)-1
IF(I.EQ.IFR)KB=MB
DO 4250 K=KA,KB
MK=MKC(K,IW)
LVNC=(K-1)*LVM4+LV2
C1=CORD(MK,IW)/SCALE
C2=CORD(MK,IW1)/SCALE
IF(IV.EQ.2)C1=C2
IF(IV.EQ.2)C2=CORD(MK,IW)/SCALE
C11(1)=C1
C22(1)=C2
IF(KSC(MK).EQ.9)60 TO 4820
IF(KSC(MK).EQ.8)60 TO 4200
CALL PLOT(C1,C2,3)

```

```

      CALL SYMB5(3,K1,SCALE,C1,C2,K,IW)
4020 CONTINUE
4025 CONTINUE
      CALL PL0T(C1,C2,3)
      LVNC=(K-1)*LVH4+LV2
      IF(K.EQ.KR)GO TO 4200
      IF(CX(LVNC,IW).EQ.999..AND.CZ(LVNC,IW).EQ.999.)GO TO 4200
      RM=0.
      IF(BLEN(LVNC,IW).LE.0.)GO TO 4100
      RM=1./BLEN(LVNC,IW)
4100 CONTINUE
      CALL UNP53(KST(LVNC,IW),A1,A2,P)
      K1=K2=0
      IF(A1.GT..05)K1=1
      IF(A2.GT..05)K2=1
      GAP1=GAP2=0.
      MK1=MKC(K+1,IW)
      ALC=ALFAC(MK)
      ALC1=ALFAC(MK1)
      COSAL=COS(ALC)
      SIN1=SIN(ALC)
      COSAL1=COS(ALC1)
      SIN11=SIN(ALC1)
      GAP=FCL(MK,4)*1.5/(24.*SCALE)
      GAPP=FCL(MK1,4)*1.5/(24.*SCALE)
      IF(IW.EQ.2)GO TO 4102
      GAP3=GAP4=0.
      IF(KSC(MK).EQ.12)GO TO 4103
      IF(ALC.NE.0.)GO TO 4101
      IF(ALFA.NE.0.)GO TO 4104
4101 CONTINUE
      IF(ABS(ALFA-SIN1).GT..05)GO TO 4104
      GAP3=GAP+COSAL
      GAP4=GAP+SIN1
      GO TO 4104
4103 CONTINUE
      IF(ALC.EQ.0.)GO TO 4104
      IF(ALC.LT.0.)GO TO 4105
      IF(ABS(SIN1-COSAL).GT..05)GO TO 4104
      GAP3=GAP+SIN1
      GAP4=-GAP+COSAL
      GO TO 4104
4105 CONTINUE
      IF(ALFA.LE.0.)GO TO 4104
      IF(ABS(COSAL+SIN1).GT..05)GO TO 4104
      GAP3=GAP+SIN1
      GAP4=GAP+COSAL
      GO TO 4104
4102 CONTINUE
      GAP3=GAP4=0.
C
C   Y DIRECTION
C
      IF(KSC(MK).EQ.12)GO TO 4109
      IF(ALC.EQ.0.)GO TO 4104
      IF(ALC.LT.0.)GO TO 4107
      IF(ABS(COSAL+ALFA).GT..05)GO TO 4104

```

```

      GAP3=GAP*COSAL
      GAP4=GAP*SINAL
      GO TO 4104
4107 CONTINUE
      IF(ABS(COSAL-ALFA).GT..05)GO TO 4104
      GAP3=GAP*COSAL
      GAP4=GAP*SINAL
      GO TO 4104
4109 CONTINUE
      IF(ALC.NE.0.)GO TO 4110
      IF(ALFA.NE.0.)GO TO 4104
4110 CONTINUE
      IF(ABS(ALFA-SINAL).GT..05)GO TO 4104
      GAP3=-GAP*SINAL
      GAP4=GAP*COSAL
      IF(ALC1.GT.0.)GAP3=-GAP3
      IF(ALC1.GT.0.)GAP4=-GAP4
4104 CONTINUE
      IF(M1.EQ.0)GAP1=.0833
      IF(M2.EQ.0)GAP2=.0933
      C1C=C1+GAP1*COSA+GAP3
      C2C=C2+GAP1*ALFA+GAP4
      IF(IW.EQ.1)GO TO 4106
      C1C=C1-GAP1*ALFA+GAP3
      C2C=C2-GAP1*COSA+GAP4
4106 CONTINUE
      CALL PLOT(C1C,C2C,3)
      IF(IW.EQ.2)GO TO 4120
      GAP3=GAP4=0.
      IF(MSC(MK1).EQ.12)GO TO 4112
      IF(ALC1.NE.0.)GO TO 4111
      IF(ALFA.NE.0.)GO TO 4108
4111 CONTINUE
      IF(ABS(ALFA-SINAL1).GT..05)GO TO 4108
      GAP3=-GAPP*COSAL1
      GAP4=-GAPP*SINAL1
      GO TO 4108
4112 CONTINUE
      IF(ALC1.EQ.0.)GO TO 4108
      IF(ALC1.LT.0.)GO TO 4114
      IF(ABS(SINAL1-COSA).GT..05)GO TO 4108
      GAP3=-GAPP*SINAL1
      GAP4=-GAPP*COSAL1
      GO TO 4108
4114 CONTINUE
      IF(ALFA.LE.0.)GO TO 4108
      IF(ABS(COSA*SINAL1).GT..05)GO TO 4108
      GAP3=GAPP*SINAL1
      GAP4=-GAPP*COSAL1
4108 CONTINUE
      C3=C1+BM*COSA/SCALE-GAP2*COSA+GAP3
      C2=C2+BM*ALFA/SCALE-GAP2*ALFA+GAP4+FCL(MK1,IW1)*1.5/(12.*SCALE)
4120 CONTINUE
      IF(IW.EQ.1)GO TO 4125
      GAP3=GAP4=0.

```

C
C Y DIRECTION

C

```

      IF(KSC(MK1).EQ.12)GO TO 4116
      IF(ALC1.EQ.0.)GO TO 4122
      IF(ALC1.LT.0.)GO TO 4115
      IF(APS(COSAL1+ALFA).GT..05)GO TO 4122
      GAP3=-GAPP*COSAL1
      GAP4=-GAPP*SINAL1
      GO TO 4122
4115 CONTINUE
      IF(APS(COSA+SINAL1).GT..05)GO TO 4122
      GAP3=-GAPP*COSAL1
      GAP4=-GAPP*SINAL1
      GO TO 4122
4116 CONTINUE
      IF(ALC1.NE.0.)GO TO 4117
      IF(ALFA.NE.0.)GO TO 4122
4117 CONTINUE
      IF(APS(ALFA-SINAL1).GT..05)GO TO 4122
      GAP3=-GAPP*SINAL1
      GAP4=GAPP*COSAL1
      IF(ALC1.LE.0)GAP4=-GAP4
4122 CONTINUE
      C1=C1-BM*ALFA/SCALE+GAP2*ALFA+GAP3*FCL(MK1,IW1)*1.5/(12.*SCALE)
      C3=C2+BM*COA/SCALE-GAP2*COA+GAP4
4125 CONTINUE
      IF(BSTIF(LVNC,IW).LE.1.E-15)GO TO 4150
      IF(IW.EQ.2)GO TO 4130
      CALL PLOT(C3,C2,2)
      GO TO 4200
4130 CALL PLOT(C1,C3,2)
      GO TO 4200
4150 CONTINUE
      C11(2)=C3
      C22(2)=C2
      IF(IW.EQ.2)C11(2)=C1
      IF(IW.EQ.2)C22(2)=C3
      IF(IPL0T(9,J).EQ.0)GO TO 4200
      CALL DASH(C11,C22,2,.1,.2)
4200 CONTINUE
      MKBM=MKB(LVNC,IW)
      IF(IW.EQ.1)MARK=MOD(MKBM,10000)
      IF(IW.EQ.2)MARK=MKBM/10000
      LI1=MARK/100
      IS1=MARK-LI1*100
      CALL SQZBITS(2,IFBB(LI1),IS1,1,IER)
      IF(1BUG.EQ.53)
        1PRINT 883,IW,MK,MK1,KSC(MK),KSC(MK1),C1C,C2C,C3,C2,
        2C1,C3,GAP1,GAP2,GAP3,GAP4,FCL(MK,4),FCL(MK1,4),COA,ALFA
        3,COSAL,SINAL
      883 FORMAT(* **DEBUG RXY**,.5I5,6F6.2,19F5.2)
      C11(2)=C3
      C22(2)=C2
      IF(IW.EQ.2)C11(2)=C1
      IF(IW.EQ.2)C22(2)=C3
      IF(1BUG.EQ.53)PRINT 886,LV2,LEVEL,IW,I,IFR,K,KA,KB,
      1MK,LVNC,C11(1),C22(1),C11(2),C22(2),ALEN(LVNC,IW),CX(LVNC,IW)
      886 FORMAT(* **DEBUG PL3 **,.10I4,6F8.2)

```

```

4250 CONTINUE
      IF(IPL0T(16,J).EQ.0)GO TO 4269
      CALL SEC9M(1,J,IVMX,LIG,NIG,LV2,LV4)
4260 CONTINUE
      DO 4280 IW=1,IVMX
        MB=MB1(IW)
        IFR=IFR2(IW)
        IW1=INVER(IW)
        DO 4280 I=1,IFR
          KA=COL1(I,IW)
          IF(I.LT.IFR)KP=COL1(I+1,IW)-1
          IF(I.EQ.IFR)KP=MB
          DO 4280 K=KA,KP
            MK=MKC(K,IW)
            LVNC=(K-1)*LV4+LV2
            C1=CORD(MK,IW)/SCALE
            C2=CORD(MK,IW1)/SCALE
            IF(IW.EQ.2)C1=C2
            IF(IW.EQ.2)C2=CORD(MK,IW)/SCALE
            K1=K2=0
            CALL UNP53(KST(LVNC,IW),A1,A2,R)
            IF(A1.GT..05)K1=1
            IF(A2.GT..05)K2=1
            GAP1=GAP2=0.

```

```

C
C COMPUTE GAP FOR DRAWING SYMBOLS AND WRITING BEAM MARKS

```

```

C C1X,C1Z SYMR. COORD AT LEFT END
C C2X C2Z SYMB. COORD. AT RIGHT END
C C1W,C2W MRK COORD.
C GAP1 GAP AT LEFT END
C GAP2 GAP AT RIGHT END
C K1=K2=0 PINNED CONN.
C K1=K2=1 RIGTD CONN.
C

```

```

      MK1=MKC(K+1,IW)
      ALC=ALFAC(MK)
      ALC1=ALFAC(MK1)
      COSAL=COS(ALC)
      SINAL=SIN(ALC)
      COSA1=COS(ALC1)
      SINAL1=SIN(ALC1)
      GAP=FCL(MK,4)*1.5/(24.*SCALE)
      GAPP=FCL(MK1,4)*1.5/(24.*SCALE)
      IF(IW.EQ.2)GO TO 4282
      GAP3=GAP4=0.
      IF(KSC(MK).EQ.12)GO TO 4287
      IF(ALC.NE.0.)GO TO 4281
      IF(ALFA.NE.0.)GO TO 4283

```

```

4281 CONTINUE
      IF(ABS(ALFA-SINAL).GT..05)GO TO 4283

```

```

      GAP3=GAP+COSAL
      GAP4=GAP+SINAL
      GO TO 4283

```

```

4287 CONTINUE
      IF(ALC.EQ.0.)GO TO 4283
      IF(ALC.LT.0.)GO TO 4288
      IF(ABS(SINAL-COSA).GT..05)GO TO 4283

```

```
GAP3=GAP*SINAL
GAP4=-GAP*COSAL
GO TO 4283
4288 CONTINUE
IF (ALFA.LE.0.)GO TO 4283
IF (ABS(COSA+SINAL).GT..05)GO TO 4283
GAP3=GAP*SINAL
GAP4=GAP*COSAL
GO TO 4283
4282 CONTINUE
GAP3=GAP4=0
C
C Y DIRECTION
C
IF (KSC(MK).EQ.12)GO TO 4291
IF (ALC.EQ.0.)GO TO 4283
IF (ALC.LT.0.)GO TO 4289
IF (ABS(COSAL+ALFA).GT..05)GO TO 4283
GAP3=GAP*COSAL
GAP4=GAP*SINAL
GO TO 4283
4289 CONTINUE
IF (ABS(COSAL-ALFA).GT..05)GO TO 4283
GAP3=GAP*COSAL
GAP4=GAP*SINAL
GO TO 4283
4291 CONTINUE
IF (ALC.NE.0.)GO TO 4292
IF (ALFA.NE.0.)GO TO 4283
4292 CONTINUE
IF (ABS(ALFA-SINAL).GT..05)GO TO 4283
GAP3=-GAP*SINAL
GAP4=GAP*COSAL
IF (ALC.GT.0.)GAP3=-GAP3
IF (ALC.GT.0.)GAP4=-GAP4
4283 CONTINUE
IF (K1.EQ.0)GAP1=.0833
IF (K2.EQ.0)GAP2=.0833
C1X=C1+GAP1*ALFA+GAP3
C1Z=C2-GAP1*COSA+GAP4
IF (IW.EQ.1)GO TO 4270
C1X=C1-GAP1*ALFA+GAP3
C1Z=C2+GAP1*COSA+GAP4
4278 CONTINUE
BM=0.
IF (K.EQ.KB)GO TO 5320
IF (BLEN(LVNC,IW).EQ.0.)GO TO 5320
BM=1./(BLEN(LVNC,IW)*SCALE)
IF (IW.EQ.2)GO TO 4276
GAP3=GAP4=0.
IF (KSC(MK1).EQ.12)GO TO 4294
IF (ALC1.NE.0.)GO TO 4293
IF (ALFA.NE.0.)GO TO 4274
4293 CONTINUE
IF (ABS(ALFA-SINAL1).GT..05)GO TO 4274
GAP3=-GAPP*COSAL1
GAP4=-GAPP*SINAL1
```

```

COSC=COS(ALC1)
CWEB1=CWEB
IF(ALC1.LT.0.)CWEB1=-CWEB1-.094/COSC
C1L=CWEB+COSC-CFLAN+SINC+.047
C2L=-CFLAN-W67
C3L=C1+C1L
C4L=C2+C2L
IF(IW.EQ.2)GO TO 4371
CALL UNF53(MST(LVNC,IW),A1,A2,B)
IF(ALFA.6F.0.)GO TO 4371
ALFA1=ASIN(ALFA)
ALFA2=ATAN(C2L/C1L)
HIPO=SQRT(C1L**2+C2L**2)
C3L=C1+HIPO*COS(ALFA1+ALFA2)
C4L=C2+HIPO*SIN(ALFA1+ALFA2)-.047
4371 CONTINUE
IF(BUG.EQ.53)
1PRINT 999,MK,CWEB,CFLAN,SINC,COSC,C1,C2,C3L,C4L,C1L,C2L,ALFA1,ALFA
999 FORMAT(' COL,MK ',I4,12F8.3)
CALL PLOT(C1,C2,3)
CALL NEWPEN(1)

C
C WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)
C
IF(IPL0T(9,J).EQ.0.AND.BSTIF(LVNC,IW).LE.1.E-15)GO TO 4375
IF(K1.EQ.0)GO TO 4375
IF(MAT.EQ.1)GO TO 4375
IF(IPL0T(5,J).EQ.0)GO TO 4375
CALL SYMB5(4,K1,SCALE,C1X,C1Z,K,IW)
4375 CONTINUE
LVNC=(K-1)*LV4+LV2
IF(IPL0T(13,J).EQ.0)GO TO 4357
IF(IW.EQ.1)MARK=MOD(MKB(LVNC,IW),10000)
IF(IW.EQ.2)MARK=MKB(LVNC,IW)/10000
LI1=MARK/100
IS1=MARK-LI1*100
JMX1=JMX(LI1)
IF(KSC(MK).LE.0)GO TO 4380
CALL POSIT(J,IW,MK,K,KA,JMX1,LVNC,C1L,C2L,C3L,C4L,C1W,C2W,SCALE)
IF(IPL0T(9,J).EQ.0.AND.KSC(MK).EQ.9)GO TO 4357
CALL LABEL5(3,J,C3L,C4L,KA,KB,LV4,IW,MK,SCALE)
4357 CONTINUE
IF(K.EQ.KB)GO TO 4380
IF(SPAN(LI1,IS1).LE.0)GO TO 4380
IF(IPL0T(10,J).EQ.0)GO TO 4358
RATIO=.5
IF(DEPTH(LI1,IS1).EQ.0)GO TO 4359
RATIO=WIDTH(LI1,IS1)/DEPTH(LI1,IS1)
4359 CONTINUE
IF(RATIO.GE.5.)GO TO 4358
IF(BSTIF(LVNC,IW).EQ.NALL)GO TO 4358
IF(BSTIF(LVNC,IW).LE.1.E-15)GO TO 4358
WIDT2=WIDTH(LI1,IS1)/(24.*SCALE)
CALL SIZE5(1,J,IW,C1W,C2W,LVNC,SCALE)
IF(JMX(LX1).EQ.JX1.AND.JMX(LX2).EQ.JX2)GO TO 4383
IF(IW.EQ.2)GO TO 4381
C1W=C1W+(WIDT1-WIDT2)*ALFA

```

```

        GO TO 4274
4294 CONTINUE
        IF (ALC1.EQ.0.) GO TO 4274
        IF (ALC1.LT.0.) GO TO 4296
        IF (ABS(SINAL1-COSA).GT..05) GO TO 4274
        GAP3=-GAPP*SINAL1
        GAP4=-GAPP*COSAL1
        GO TO 4274
4296 CONTINUE
        IF (ALFA.LE.0.) GO TO 4274
        IF (ABS(COSA+SINAL1).GT..05) GO TO 4274
        GAP3=GAPP*SINAL1
        GAP4=-GAPP*COSAL1
4274 CONTINUE
        C2X=C1+BM*COSA-GAP2*COSA+GAP3
        C2Z=C2+BM*ALFA-GAP2*ALFA+GAP4+FCL(MK1,IN1)*1.5/(12.*SCALE)
4276 CONTINUE
        IF (IW.EQ.1) GO TO 4284
        GAP3=GAP4=0.
C
C   Y DIRECTION
C
        IF (KSC(MK1).EQ.12) GO TO 429E
        IF (ALC1.EQ.0.) GO TO 4278
        IF (ALC1.LT.0.) GO TO 4297
        IF (ABS(COSAL1+ALFA).GT..05) GO TO 4278
        GAP3=-GAPP*COSAL1
        GAP4=-GAPP*SINAL1
        GO TO 4278
4297 CONTINUE
        IF (ABS(COSA+SINAL1).GT..05) GO TO 4278
        GAP3=-GAPP*COSAL1
        GAP4=-GAPP*SINAL1
        GO TO 4278
4298 CONTINUE
        IF (ALC1.NE.0.) GO TO 4299
        IF (ALFA.NE.0.) GO TO 4278
4299 CONTINUE
        IF (ABS(ALFA-SINAL1).GT..05) GO TO 4278
        GAP3=-GAPP*SINAL1
        GAP4=GAPP*COSAL1
        IF (ALC1.LE.0.) GAP4=-GAP4
4278 CONTINUE
        C2X=C1-BM*ALFA+GAP2*ALFA+GAP3+FCL(MK1,IN1)*1.5/(12.*SCALE)
        C2Z=C2+BM*COSA-GAP2*COSA+GAP4
4284 CONTINUE
C
C   COMPUTE C1W,C2W
C
        MK2=MKC(K+1,IN)
        C2W=(CORD(MK2,IN1)+CORD(MK,IN1))/(SCALE*2.)
        C1W=C1+BM*.5*COSA
        IF (IW.EQ.2) C1W=C1-BM*.5*ALFA
        IF (IW.EQ.2) C2W=C2+BM*.5*COSA
        IF (IBUG.EQ.53)
1PRINT A7R,K,IN,LVNC,MK,MK1,KSC(MK),KSC(MK1),K1,K2,C1,C2
2,C1X,C1Z,C2X,C2Z,C1W,C2W,ALFA,COSA,SINAL,COSAL

```

7,GAP1,GAP2,GAP3,GAP4
 870 FORMAT(' **DEBUG CORD**',3I3,2I4,4I3,AF6.2,AF5.2)

C
 C COMPUTE C3L,C4L FOR COLUMN MARKS (STEEL)

C
 5320 FACTOR=.75
 HGT=FACTOR*.125
 CMER=1.5*FCL(MK,4)/(24.*SCALE)
 CFLAN=1.5*FCL(MK,5)/(24.*SCALE)
 ALC1=ALFAC(MK)
 IF(KSC(MK).EQ.12.AND.ALC1.LT.0.)ALC1=0.
 IF(KSC(MK).EQ.11.AND.ALC1.GT.0.)ALC1=0.
 SINC=SIN(ALC1)
 COSC=COS(ALC1)
 C1L=CMER*SINC+CFLAN*COSC+.047
 C2L=-.047-HGT
 IF(KSC(MK).EQ.11)C1L=CMER*COSC-CFLAN*SINC+.047
 C3L=C1+C1L
 C4L=C2+C2L
 IF(IV.EQ.2)GO TO 4279
 CALL UNP53(KST(LVNC,IV),A1,A2,B)
 IF(ALFA.GE.0.)GO TO 4279
 ALFA1=ASIN(ALFA)
 ALFA2=ATAN(C2L/C1L)
 HIPO=SQRT(C1L**2+C2L**2)
 C3L=C1+HIPO*COS(ALFA1+ALFA2)
 C4L=C2+HIPO*SIN(ALFA1+ALFA2)-.047

4279 CONTINUE
 CALL PLOT(C1,C2,3)
 CALL NEWPEN(1)

C
 C WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)

C
 IF(IPILOT(9,J).EQ.0.AND.BSTIF(LVNC,IV).LE.1.E-15)GO TO 4285
 IF(K1.EQ.0)GO TO 4285
 IF(IPILOT(5,J).EQ.0)GO TO 4285
 CALL SYMB5(4,K1,SCALE,C1X,C1Z,K,IV)
 4285 CONTINUE

LVNC=(K-1)*LVM4+LV2
 IF(IPILOT(13,J).EQ.0)GO TO 4261
 IF(IV.EQ.1)MARK=MOD(MKB(LVNC,IV),10000)
 IF(IV.EQ.2)MARK=MKB(LVNC,IV)/10000
 LI1=MARK/100
 JMN1=MX(LI1)

IF(KSC(MK).LE.0)GO TO 4260
 CALL POSIT(J,IV,MK,K,KA,JMN1,LVNC,C1L,C2L,C3L,C4L,C1V,C2V,SCALE)
 IF(IPILOT(9,J).EQ.1)GO TO 4260
 IF(CPRO(MK).EQ.4)MC3=6)GO TO 4261

4260 CONTINUE
 CALL LABEL5(3,J,C3L,C4L,KA,KB,LVM4,IV,MK,SCALE)

4261 CONTINUE
 IF(IPUG.EQ.53)WRITE(6,(' **OV53PL**',30I4))
 MK,J,IPILOT(JJ1,J),JJ1=1,23)
 IF(K.EQ.KB)GO TO 4286
 IF(BSTIF(LVNC,IV).EQ.VALL)GO TO 4262
 IF(BSTIF(LVNC,IV).LE.1.E-15.AND.KSC(MK).EQ.0)GO TO 4280
 IF(BSTIF(LVNC,IV).LE.1.E-15.AND.IPILOT(9,J).EQ.0)GO TO 4280

```

      IF (BSTIF(LVNC,IV).LE.1.E-15)GO TO 4262
      IF (IPL0T(18,J).EQ.0)GO TO 4262
      IF (IPU6.EQ.53)WRITE(6,*)('OV53PL1=0.3074')
      MK=J.(IPL0T(JJ1,J).JJ1=1.25)
      CALL SIZE5(1,J,IV,C1W,C2W,LVNC,SCALE)
4262 CONTINUE
      IF (IPL0T(14,J).EQ.0)GO TO 4286
      CALL LABEL5(4,J,C1W,C2W,KA,KB,LVM4,IV,LVNC,SCALE)
      IF (RSTIF(LVNC,IV).EQ.WALL)GO TO 4280
      IF (ESTIF(LVNC,IV).LE.1.E-15)GO TO 4280
4286 IF (K2.EQ.0)GO TO 4280
      IF (IPL0T(5,J).EQ.0)GO TO 4280
      CALL SYMP5(5,K2,SCALE,C2X,C2Z,K,IV)
4280 CONTINUE
      IF (IPL0T(15,J).EQ.0)GO TO 4264
      CALL SECBM(2,J,IMMX,L16,N16,LV2,LVM4)
      IF (IPL0T(19,J).EQ.0)GO TO 4264
      CALL SIZE5(2,J,IV,C1W,C2W,LVNC,SCALE)
4264 CONTINUE
      LVV=LV2-1
      KAA=COL1(1,1)
      MK=MKC(KAA,1)
      C1=STP1V/2.
      C2=STPYMN
      IF (IPL0T(12,J).EQ.0)GO TO 4290
      MB=MP1(1)
      IFR1=IFP2(1)
      SMX1=0.
      SMX2=0.
      DO 4304 IF=1,IFR1
      KA=COL1(IF,1)
      IF (IF.LT.IFR1)KB=COL1(IF+1,1)-1
      IF (IF.EQ.IFR1)KB=MB
      DO 4304 K=KA,KB
      K1=KB-K+KA
      MK=MKC(K,1)
      MK1=MKC(K1,1)
      LVP1=(K-1)*LVM4+LV2
      LVP2=(K1-1)*LVM4+LV2
      IF (CX(LVP1,1).EQ.999..AND.CZ(LVP1,1).EQ.999.)GO TO 4303
      IF (CORD(MK,1).LT.SMX1)SMX1=CORD(MK,1)
4303 IF (CX(LVP2,1).EQ.999..AND.CZ(LVP2,1).EQ.999.)GO TO 4304
      IF (CORD(MK1,1).GT.SMX2)SMX2=CORD(MK1,1)
4304 CONTINUE
      C1=(SMX2-SMX1)*.5/SCALE
      CALL LABEL5(6,J,C1,C2,KA,KB,LVM4,IV,LVW,SCALE)
4290 CONTINUE
      IF (IPL0T(18,J).EQ.0)GO TO 4265
      CALL DIM5(2,J,KA,KB,IV,LVM4,LV2,SCALE,STPYMN)
4265 CONTINUE
      CALL PLOT(0.,0.,-3)
      GO TO 4300

C
C CONCRETE STRUCTURE
C
4295 CONTINUE
      DO 4360 IW=1,2

```

```

      MB=MR1(IW)
      IFR=IFR2(IW)
      IM1=INVER(IW)
      DO 4360 I=1,IFR
      KA=CPL1(I,IW)
      IF(I.LT.IFR)KP=COL1(I+1,IW)-1
      IF(I.EQ.IFR)KB=MR
      DO 4360 K=KA,KP
      MK=MKC(K,IW)
      LVNC=(K-1)*LVW4+LV2
      C1=CORD(MK,IW)/SCALE
      C2=CORD(MK,IM1)/SCALE
      IF(IM.EQ.2)C1=C2
      IF(IM.EQ.2)C2=CORD(MK,IW)/SCALE
      C11(1)=C1
      C22(1)=C2
      IF(CX(LVNC,IW).EQ.999..AND.CZ(LVNC,IW).EQ.999.)GO TO 4360
      IF(KSC(MK).EQ.9)GO TO 4322
      IF(KSC(MK).EQ.0)GO TO 4360
      CALL PLOT(C1,C2,3)
      CALL SYMB6(3,K1,SCALE,C1,C2,K,IW)
4322 CONTINUE
      LVNC=(K-1)*LVW4+LV2
      IF(K.EQ.KB)GO TO 4355
      BM=0.
      IF(BLEN(LVNC,IW).LE.0.)GO TO 4330
      BM=1./BLEN(LVNC,IW)
4330 CONTINUE
      CALL UNP53(KST(LVNC,IW),A1,A2,B)
      K1=K2=0
      IF(A1.GT..05)K1=1
      IF(A2.GT..05)K2=1
      GAP1=GAP2=0.
      MK1=MKC(K+1,IW)
      IF(IM.EQ.2)GO TO 4332
      GAP1=X(MK)/(24.*SCALE)
      GAP2=X(MK1)/(24.*SCALE)
      GO TO 4334
4332 CONTINUE
      GAP1=X(MK)/(24.*SCALE)
      GAP2=X(MK1)/(24.*SCALE)
      IF(KSC(MK).EQ.3.OR.KSC(MK).EQ.10)GAP1=Y(MK)/(24.*SCALE)
      IF(KSC(MK1).EQ.3.OR.KSC(MK1).EQ.10)GAP2=Y(MK1)/(24.*SCALE)
4334 CONTINUE
      IF(K1.EQ.0)GAP1=GAP1+2.*HGT
      IF(K2.EQ.0)GAP2=GAP2+2.*HGT
      C1C=C1+GAP1*COXA
      C2C=C2+GAP1*ALFA+FCL(MK,IM1)/(12.*SCALE)
      IF(IM.EQ.1)GO TO 4336
      C1C=C1-GAP1*ALFA+FCL(MK,IM1)/(12.*SCALE)
      C2C=C2+GAP1*COXA
4336 CONTINUE
      CALL PLOT(C1C,C2C,3)
      IF(IM.EQ.2)GO TO 4340
      C2=C2+BM*ALFA/SCALE-GAP2*ALFA+FCL(MK1,IM1)/(12.*SCALE)
      C3=C1+BM*(1.-ALFA*ALFA)*5/SCALE-GAP2*COXA
4340 IF(IM.EQ.1)GO TO 4342

```


C3=C2+BM*(1.-ALFA*ALFA)**.5/SCALE-GAP2*COXA
 C1=C1-BM*ALFA/SCALE+GAP2*ALFA+FCL(MK1,IM1)/(12.*SCALE)

4342 CONTINUE

IF(BSTIF(LVNC,IM).LE.1.E-15)GO TO 4350

MKBM=MKP(LVNC,IM)

IF(IV.EQ.1)MARK=MOD(MKBM,10000)

IF(IV.EQ.2)MARK=MKBM/10000

LI1=MARK*100

IS1=MARK-LI1*100

RATIO=.5

IF(LI1.EQ.3.AND.IBUG.EQ.53)PRINT 989,MARK,MK,MK1,C1C,C2C,C2,C3,
 WIDTH(LI1,IS1),DEPTH(LI1,IS1),ALFA,GAP1,GAP2,X(MK),X(MK1)

989 FORMAT(' BM301',3I5,11F10.3)

IF(DEPTH(LI1,IS1).LE.0.)GO TO 4341

RATIO=WIDTH(LI1,IS1)/DEPTH(LI1,IS1)

4341 CONTINUE

IF(BSTIF(LVNC,IM).EQ.WALL)WIDT1=WIDTH(LI1,IS1)*.5/(SCALE*12.)

IF(IV.EQ.2)GO TO 4345

IF(BSTIF(LVNC,IM).NE.WALL)GO TO 4343

C2C=C2C+WIDT1

CALL PLOT(C1C,C2C,3)

C2=C2-WIDT1*2.

CALL PLOT(C3,C2C,2)

CALL PLOT(C3,C2,2)

CALL PLOT(C1C,C2,2)

C2=C2+WIDT1*2.

CALL PLOT(C1C,C2,2)

GO TO 4355

4343 CONTINUE

IF(RATIO.GE.5.)GO TO 4344

C11(1)=C1C

YBIN=WIDTH(LI1,IS1)/(SCALE*24.)

C22(1)=C2C-YBIN

C11(2)=C3

C22(2)=C2-YBIN

CALL DASH(C11,C22,2,.1,.1)

C22(1)=C2C+YBIN

C22(2)=C2+YBIN

CALL DASH(C11,C22,2,.1,.1)

LY1=LY(MK)

JY1=JY(MK)

LY2=LY(MK1)

JY2=JY(MK1)

IF(JY(MK).GT.1.AND.MK(LY1).NE.JY1)GO TO 4355

IF(LY1.EQ.0.AND.KSC(MK).EQ.9)GO TO 4355

YBIN2=YBIN

YBIN=Y(MK)/(24.*SCALE)

IF(KSC(MK).EQ.9)YBIN=Y(MK1)/(24.*SCALE)

IF(KSC(MK).EQ.2)YBIN=X(MK)/(SCALE*24.)

IF(YBIN2.GT.YBIN)YBIN=YBIN2

C2C2=C2C-YBIN

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C2C2=C2C+YBIN

CALL PLOT(C1C,C2C2,3)

YBIN1=Y(MK1)/(SCALE*24.)

IF(KSC(MK1).EQ.2)YBIN1=X(MK1)/(SCALE*24.)

IF(KSC(MK1).EQ.9)YBIN1=Y(MK1)/(24.*SCALE)

IF(YBIN2.GT.YBIN1)YBIN1=YBIN2

```

C2C2=C2-YRIM1
IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C2C2=C2+YRIM1
CALL PLOT(C3,C2C2,2)
GO TO 4355
4344 CONTINUE
C
C SYMPOI FOR SLABS YDIRECTION
C
ALFA1=15./57.2*578
ALFA2=ASIN(ALFA1)
ARROY=.1*COS(ALFA1+ALFA2)
ARROY=.1*SIN(ALFA1+ALFA2)
LY1=LY(MK)
JY1=JY(MK)
LY2=LY(MK1)
JY2=JY(MK1)
IF(JY(MK).GT.1.AND.JMX(LY1).NE.JY1)GO TO 4352
IF(LY1.EQ.0.AND.KSC(MK).EQ.9)GO TO 4352
YRIM=Y(MK)/(24.*SCALE)
IF(KSC(MK).EQ.9)YRIM=Y(MK1)/(24.*SCALE)
C3C=C2C-YRIM
IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C3C=C2C+YRIM
CALL PLOT(C1C,C3C,3)
YRIM=Y(MK1)/(24.*SCALE)
IF(KSC(MK1).EQ.9)YRIM=Y(MK)/(24.*SCALE)
C3C=C2-YRIM
IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C3C=C2+YRIM
CALL PLOT(C3,C3C,2)
4352 CONTINUE
IF(IPL0T(14,J).EQ.0)GO TO 4355
C3C=C1C+(GAP1+.125)*COSA
C2C=C2C+(GAP1+.125)*ALFA
C4C=C3-(GAP2+.125)*COSA
C5C=C2-(GAP2+.125)*ALFA
CALL PLOT(C3C,C2C,3)
CALL PLOT(C4C,C5C,2)
CALL PLOT(C3C,C2C,3)
C3C=C3C+ARROY
C2C=C2C+ARROY
CALL PLOT(C3C,C2C,2)
CALL PLOT(C4C,C5C,3)
C4C=C4C-ARROY
C5C=C5C-ARROY
CALL PLOT(C4C,C5C,2)
GO TO 4355
4345 CONTINUE
IF(BST)IF(LVNC,IV).NE.WALL)GO TO 4346
C1=C1-WIDT1
XD=1.5*X(MK)/(SCALE*24.)
IF(KSC(MK).EQ.3.OR.KSC(MK).EQ.10)XD=XD+Y(MK)/X(MK)
CALL PLOT(C1,C2C,3)
CALL PLOT(C1,C3,2)
C1=C1+WIDT1*2.
CALL PLOT(C1,C3,2)
CALL PLOT(C1,C2C,2)
C1=C1-WIDT1*2.
CALL PLOT(C1,C2C,2)

```

GO TO 4355

4346 CONTINUE

IF(RATIO.GE.5.)GO TO 4347

YBIN=WIDTH*(I1-IS1)/(SCALE*24.)

C11(1)=C1C-YBIN*COXA

C22(1)=C2C-YBIN*ALFA

C11(2)=C1-YBIN*COXA

C22(2)=C3-YBIN*ALFA

CALL DASH(C11,C22,2,.1,.1)

C11(1)=C1C+YBIN*COXA

C11(2)=C1+YBIN*COXA

C22(1)=C2C+YBIN*ALFA

C22(2)=C3+YBIN*ALFA

CALL DASH(C11,C22,2,.1,.1)

LY1=LX(MK)

JY1=JX(MK)

LY2=LX(MK1)

JY2=JX(MK1)

IF(JX(MK).GT.1.AND.JMX(LY1).NE.JY1)GO TO 4355

IF(LY1.EQ.0.AND.KSC(MK).EQ.9)GO TO 4355

YBIN2=YBIN

YBIN=X(MK)/(24.*SCALE)

IF(KSC(MK).EQ.9)YBIN=X(MK1)/(24.*SCALE)

IF(YBIN2.GT.YBIN)YBIN=YBIN2

C2C2=C1C-YBIN*COXA

IF(JPX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C2C2=C1C+YBIN*COXA

CALL PLOT(C2C2,C2C,3)

YBIN1=X(MK1)/(SCALE*24.)

IF(KSC(MK1).EQ.9)YBIN1=X(MK1)/(24.*SCALE)

IF(YBIN2.GT.YBIN1)YBIN1=YBIN2

C2C2=C1-YBIN1*COXA

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C2C2=C1+YBIN1*COXA

CALL PLOT(C2C2,C3,2)

GO TO 4355

4347 CONTINUE

C

C SYBOL FOR SLABS YDIRECTION

C

ALFA1=15./57.29578

ALFA2=ASIN(ALFA1)

LY1=LX(MK)

JY1=JX(MK)

LY2=LX(MK1)

JY2=JX(MK1)

IF(JX(MK).GT.1.AND.JMX(LY1).NE.JY1)GO TO 4353

IF(LY1.EQ.0.AND.KSC(MK).EQ.9)GO TO 4353

YBIN=X(MK)/(24.*SCALE)

IF(KSC(MK).EQ.9)YBIN=X(MK1)/(24.*SCALE)

C3C=C1C-YBIN

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C3C=C1C+YBIN

CALL PLOT(C3C,C2C,3)

YBIN=X(MK1)/(SCALE*24.)

IF(KSC(MK1).EQ.9)YBIN=X(MK1)/(24.*SCALE)

C3C=C1-YBIN

IF(JMX(LY1).EQ.JY1.AND.JMX(LY2).EQ.JY2)C3C=C1+YBIN

CALL PLOT(C3C,C3,2)

4353 CONTINUE

```

      IF(I)PLOT(14,J).EQ.0)GO TO 4355
      C3C=C3C-(GAP1+.125)*ALFA
      C2C=C2C+(GAP1+.125)*COSA
      C4C=C1+(GAP2+.125)*ALFA
      C5C=C3-(GAP2+.125)*CCSA
      CALL PLOT(C3C,C2C,3)
      CALL PLOT(C4C,C5C,2)
      CALL PLOT(C3C,C2C,3)
      C3C=C3C-.1*SIN(ALFA1+ALFA2)
      C2C=C2C+.1*COS(ALFA1+ALFA2)
      CALL PLOT(C3C,C2C,2)
      CALL PLOT(C4C,C5C,3)
      C4C=C4C+.1*SIN(ALFA2+ALFA1)
      C5C=C5C-.1*COS(ALFA2+ALFA1)
      CALL PLOT(C4C,C5C,2)
      GO TO 4355
4350 CONTINUE
      C11(2)=C3
      C22(2)=C2
      IF(IW.EQ.2)C11(2)=C1
      IF(IW.EQ.2)C22(2)=C3
      IF(I)PLOT(9,J).EQ.0)GO TO 4355
      CALL DASH(C11,C22,2,.1,.2)
4355 CONTINUE
      CALL SGZBITS(2,IFBB(LI1),IS1,1,IER)
      IF(IRUB.EQ.53)
1PRINT 883,IW,MK,MK1,KSC(MK),KSC(MK1),C1C,C2C,C3,C2,
2C1,C3,GAP1,GAP2,X(MK),X(MK1)
      C11(2)=C3
      C22(2)=C2
      IF(IW.EQ.2)C11(2)=C1
      IF(IW.EQ.2)C22(2)=C3
      IF(1BUG.EQ.53)PRINT 886,LV2,LEVEL,IW,I,IFR,K,KA,KB,
1MK,LVNC,C11(1),C22(1),C11(2),C22(2),BLEN(LVNC,IW),CX(LVNC,IW)
4360 CONTINUE
      IF(I)PLOT(16,J).EQ.0)GO TO 4362
      CALL SECM(3,J,IWMX,LIG,MIG,LV2,LVN4)
4362 CONTINUE
      DO 4380 IW=1,IWMX
      MB=MB1(IW)
      IFR=IFR2(IW)
      IW1=INVER(IW)
      DO 4380 I=1,IFR
      KA=COL1(I,IW)
      IF(I.LT.IFR)KB=COL1(I+1,IW)-1
      IF(I.EQ.IFR)KB=MB
      DO 4380 K=KA,KB
      MK=MKC(K,IW)
      LVNC=(K-1)*LVN4+LV2
      C1=CORD(MK,IW)/SCALE
      C2=CORD(MK,IW1)/SCALE
      IF(IW.EQ.2)C1=C2
      IF(IW.EQ.2)C2=CORD(MK,IW)/SCALE
      K1=K2=8
      CALL UNP53(KST(LVNC,IW),A1,A2,B)
      IF(A1.GT..85)K1=1
      IF(A2.GT..85)K2=1

```

GAP1=GAP2=0.

C
C COMPUTE GAP FOR DRAWING SYMBOLS AND WRITING BEAM MARKS
C C1X,C1Z SYMB. COORD AT LEFT END
C C2X C2Z SYMB. COORD. AT RIGHT END
C C1W,C2W MRK COORD.
C GAP1 GAP AT LEFT END
C GAP2 GAP AT RIGHT END
C K1=K2=0 PINNED CONN.
C K1=K2=1 RIGID CONN.
C

MK1=MKC(K+1,1W)
LX1=LX(MK)
JX1=JX(MK)
LX2=LX(MK1)
JX2=JX(MK1)
IF(1W.EQ.2)GO TO 4364
LX1=LY(MK)
JX1=JY(MK)
LX2=LY(MK1)
JX2=JY(MK1)

4364 CONTINUE
GAP1=X(MK)/(24.*SCALE)
GAP2=X(MK1)/(24.*SCALE)
IF(1W.EQ.1)GO TO 4372

4370 CONTINUE
IF(KSC(MK).EQ.3.OR.KSC(MK).EQ.10)GAP1=Y(MK)/(24.*SCALE)
IF(KSC(MK1).EQ.3.OR.KSC(MK1).EQ.10)GAP2=Y(MK1)/(24.*SCALE)

4372 CONTINUE
IF(K1.EQ.0)GAP1=GAP1+.0833
IF(K2.EQ.0)GAP2=GAP2+.0833
C1X=C1+GAP1

C1Z=C2+GAP1*ALFA/COSA
BM=0.
IF(K.EQ.KB)GO TO 4373
IF(BLEN(LVNC,1W).EQ.0.)GO TO 4373
BM=1./(BLEN(LVNC,1W)*SCALE)
C2X=C1+BM*COSA-GAP2

C2Z=C2+BM*ALFA-GAP2*ALFA/COSA
IF(1W.EQ.1)GO TO 4374
C1X=C1+GAP1*ALFA/COSA
C1Z=C2+GAP1
C2Z=C2+BM*COSA-GAP2
C2X=C1+BM*ALFA-GAP2*ALFA/COSA

4374 CONTINUE

C
C COMPUTE C1W,C2W
C

MK2=MKC(K+1,1W)
MKBM=MKB(LVNC,1W)
MARK=MOD(MKBM,10000)
IF(1W.EQ.2)MARK=MKBM/10000
LI1=MARK/100
IS1=MARK-LI1*100
WIDT1=WIDTMLI1,IS1)/(SCALE*24.)
YBIN=0.
YBIN1=0.

```

IF(JX1.GT.1.AND.JMX(LX).NE.JX1)GO TO 4364
IF(LY1.EQ.0.AND.KSC(MK).EQ.9)GO TO 4368
IF(IW.EQ.1)GO TO 4367
YBM=X(MK)/(124.*SCALE)
IF(KSC(MK).EQ.9)YBM=X(MK1)/(124.*SCALE)
YBM1=Y(MK1)/(124.*SCALE)
IF(KSC(MK1).EQ.9)YBM1=X(MK)/(124.*SCALE)
GO TO 4364

```

```

4367 CONTINUE
YBM=Y(MK)/(124.*SCALE)
IF(KSC(MK).EQ.9)YBM=Y(MK1)/(124.*SCALE)
YBM1=Y(MK1)/(124.*SCALE)
IF(KSC(MK1).EQ.9)YBM1=Y(MK)/(124.*SCALE)

```

```

4368 CONTINUE
YBM2=(YBM+YBM1)*.5
IF(DEPTH(LI1,IS1).EQ.0.)GO TO 4369
RATIO=WIDTH(LI1,IS1)/DEPTH(LI1,IS1)
IF(RATIO.GE.5.)WIDT1=0.

```

```

4369 CONTINUE
WIDT1=AMAX1(YBM2,WIDT1)
C2W=(CORD(MK2,IW1)+CORD(MK,IW1))/(SCALE*2.)+WIDT1*COSA
C1W=C1+BM*.5*COSA-WIDT1*ALFA
IF(IW.EQ.2)C1W=C1-BM*.5*ALFA-WIDT1*COSA
IF(IW.EQ.2)C2W=C2+BM*.5*COSA-WIDT1*ALFA
IF(IPUG.EQ.53)
1PRINT R70,K,IW,LVNC,MK,MK1,KSC(MK),KSC(MK1),K1,K2,C1,C2
2,C1X,C1Z,C2X,C2Z,C1W,C2W,ALFA,COSA

```

C
C COMPUTE C3L,C4L FOR COLUMN MARKS (CONCRETE)
C

```

4373 CONTINUE
FACTOR=.75
HGT=FACTOR*.125
LIX=LX(MK)
JIX=JX(MK)
LIY=LY(MK)
JIY=JY(MK)
IF(DEPTH(LIX,JIX).EQ.0..OR.DEPTH(LIY,JIY).EQ.0.)GO TO 4376
WIDX=WIDTH(LIX,JIX)
WIDY=WIDTH(LIY,JIY)
RATIOX=WIDX/DEPTH(LIX,JIX)
RATIOY=WIDY/DEPTH(LIY,JIY)
IF(RATIOX.GE.5.)WIDX=0.
IF(RATIOY.GE.5.)WIDY=0.
GO TO 4379

```

```

4376 CONTINUE
WIDX=0.
WIDY=0.

```

```

4379 CONTINUE
CWB1=AMAX1(X(MK),WIDX,WIDY)
CPLA1=AMAX1(Y(MK),WIDX,WIDY)
IF(KSC(MK).EQ.3)CPLA1=AMAX1(Y(MK),WIDX,WIDY)
CWB=1.5*CWB1/(124.*SCALE)
CPLA=1.5*CPLA1/(124.*SCALE)
ALC1=ALFAC(MK)
IF(ALC1.GT.0.)ALC1=0.
SINC=SINC(ALC1)

```

```

      COSC=COS(ALC1)
      CWEB1=CWEB
      IF (ALC1.LT.0.) CWEB1=-CWEB1-.094/COSC
      C1L=CWEB+COSC-CFLAN+SINC+.047
      C2L=-CFLAN-WGT
      C3L=C1+C1L
      C4L=C2+C2L
      IF (IW.EQ.2) GO TO 4371
      CALL UNF53(KST(LVNC,IW),A1,A2,B)
      IF (ALFA.GF.0.) GO TO 4371
      ALFA1=ASIN(ALFA)
      ALFA2=ATAN(C2L/C1L)
      HIPO=SQRT(C1L**2+C2L**2)
      C3L=C1+HIPO*COS(ALFA1+ALFA2)
      C4L=C2+HIPO*SIN(ALFA1+ALFA2)-.047
4371 CONTINUE
      IF (IBUG.EQ.53)
        1PRINT 999,MK,CWEB,CFLAN,SINC,COSC,C1,C2,C3L,C4L,C1L,C2L,ALFA1,ALFA
999 FORMAT(' COL,MK ',I4,12F8.3)
      CALL PLOT(C1,C2,3)
      CALL NEWPEN(1)
C
C      WRITE BEAM MARKS AND CONNECTION TYPE (FLOOR PLAN)
C
      IF (IPLOT(9,J).EQ.0.AND.BSTIF(LVNC,IW).LE.1.E-15) GO TO 4375
      IF (K1.EQ.0) GO TO 4375
      IF (MAT.EQ.1) GO TO 4375
      IF (IPLOT(5,J).EQ.0) GO TO 4375
      CALL SYMB5(4,K1,SCALE,C1X,C1Z,K,IW)
4375 CONTINUE
      LVNC=(K-1)*LVW4+LV2
      IF (IPLOT(13,J).EQ.0) GO TO 4357
      IF (IW.EQ.1) MARK=MOD(MKB(LVNC,IW),10000)
      IF (IW.EQ.2) MARK=MKB(LVNC,IW)/10000
      LI1=MARK/100
      IS1=MARK-LI1*100
      JMX1=JMX/LI1
      IF (KSC(MK).LE.0) GO TO 4380
      CALL POSIT(J,IW,MK,K,KA,JMX1,LVNC,C1L,C2L,C3L,C4L,C1W,C2W,SCALE)
      IF (IPLOT(9,J).EQ.0.AND.KSC(MK).EQ.9) GO TO 4357
      CALL LABEL5(3,J,C3L,C4L,KA,KB,LVW4,IW,MK,SCALE)
4357 CONTINUE
      IF (K.EQ.KB) GO TO 4380
      IF (SPAN(LI1,IS1).LE.0.) GO TO 4380
      IF (IPLOT(18,J).EQ.0) GO TO 4358
      RATIO=.5
      IF (DEPTH(LI1,IS1).EQ.0.) GO TO 4359
      RATIO=WIDTH(LI1,IS1)/DEPTH(LI1,IS1)
4359 CONTINUE
      IF (RATIO.GE.5.) GO TO 4350
      IF (BSTIF(LVNC,IW).EQ.WALL) GO TO 4350
      IF (BSTIF(LVNC,IW).LE.1.E-15) GO TO 4350
      WIDT2=WIDTH(LI1,IS1)/(24.*SCALE)
      CALL SIZE5(1,J,IW,C1W,C2W,LVNC,SCALE)
      IF (JMX1.LX1).EQ.JX1.AND.JMX1.LX2).EQ.JX2) GO TO 4383
      IF (IW.EQ.2) GO TO 4381
      C1W=C1W+(WIDT1-WIDT2)*ALFA

```

```

      C2W=C2W-(WIDT1-WIDT2)*COSA
      GO TO 4383
4381 CONTINUE
      C1W=C1W+(WIDT1-WIDT2)*COSA
      C2W=C2W+(WIDT1-WIDT2)*ALFA
4383 CONTINUE
4385 CONTINUE
      IF(IPL0T(14,J).EQ.0)GO TO 4377
      WIDT2=WIDTM(L1,IS1)/(24.*SCALE)
      IF(RATIO.GE.5.)WIDT2=0.
      IF(IW.EQ.1)GO TO 4385
      C1W=C1W+2.*WIDT1*COSA
      C2W=C2W+2.*WIDT1*ALFA
      IF(JMY(LV1).EQ.JX1.AND.JMY(LV2).EQ.JX2)GO TO 4387
      C1W=C1W-(WIDT1-WIDT2)*COSA
      C2W=C2W-(WIDT1-WIDT2)*ALFA
      GO TO 4387
4385 CONTINUE
      C1W=C1W+2.*WIDT1*ALFA
      C2W=C2W-2.*WIDT1*COSA
      C1W=C1W-(WIDT1-WIDT2)*ALFA
      C2W=C2W+(WIDT1-WIDT2)*COSA
4387 CONTINUE
      CALL LAPEL(4,J,C1W,C2W,KA,KR,LVM4,IW,LVNC,SCALE)
4377 IF(K2.EQ.0)GO TO 4380
4380 CONTINUE
      IF(IPL0T(15,J).EQ.0)GO TO 4384
      CALL SECBM(2,J,IMX,LIG,NIG,LV2,LVM4)
4382 CONTINUE
      IF(IPL0T(19,J).EQ.0)GO TO 4384
      CALL SIZE5(2,J,IW,C1W,C2W,LVNC,SCALE)
4384 CONTINUE
      IF(IPL0T(8,J).EQ.0)GO TO 4386
      CALL DIM5(2,J,KA,KB,IW,LVM4,LV2,SCALE,STPXMM)
4386 CONTINUE
      LVV=LV2-1
      KAA=COL1(1,1)
      MK=MKC(KAA,1)
      C2=STPYMM
      IF(IPL0T(12,J).EQ.0)GO TO 4390
      MB=MB1(1)
      IFR1=IFR2(1)
      SMX1=0.
      SMX2=0.
      DO 4396 IF=1,IFR1
      KA=COL1(IF,1)
      IF(IF.LT.IFR1)KB=COL1(IF+1,1)-1
      IF(IF.EQ.IFR1)KB=MB
      DO 4396 K=KA,KB
      K1=KB-K+KA
      MK=MKC(K,1)
      MK1=MKC(K1,1)
      LVP1=(K-1)*LVM4+LV2
      LVP2=(K1-1)*LVM4+LV2
      IF(CX(LVP1,1).EQ.999..AND.CZ(LVP1,1).EQ.999.)GO TO 4395
      IF(CORD(MK,1).LT.SMX1)SMX1=CORD(MK,1)
4395 CONTINUE

```



```
IF(CX(LVP2,1).EQ.999..AND.CZ(LVP2,1).EQ.999.)GO TO 4396
IF(CORD(MK1,1).GT.SMX2)SMX2=CORD(MK1,1)
```

```
4396 CONTINUE
```

```
C1=(SMX2-SMX1)*.5/SCALE
```

```
CALL LABEL5(6,J,C1,C2,KA,KB,LVM4,IM,LVV,SCALE)
```

```
4300 CONTINUE
```

```
CALL PLOT(0.,0.,-3)
```

```
4300 CONTINUE
```

```
IF(IFPL2.EQ.0)GO TO 4310
```

```
GO TO 5200
```

```
4305 CONTINUE
```

```
C
```

```
C ERROR MESSAGE***** STACK NOS. DO NOT CORRESPOND TO
```

```
C ANY COLUMN IN FRAMES
```

```
C
```

```
PRINT 871
```

```
871 FORMAT(' *** MESSAGE PL-1 *** STACK NOS DO NOT
```

```
1. * CORRESPOND TO ANY FRAME * )
```

```
GO TO 9998
```

```
4310 CONTINUE
```

```
C
```

```
C ERROR MESSAGE ***** LEVEL NO. TO BE PLOTTED DO NOT
```

```
C CORRESPOND TO ANY LEVEL OF THE STRUCTURE
```

```
C
```

```
PRINT 872
```

```
872 FORMAT(' *** MESSAGE PL-2 *** LEVELS TO BE PLOTTED
```

```
1. * DO NOT CORRESPOND TO THE ACTUAL LEVELS * )
```

```
GO TO 9998
```

```
5200 CONTINUE
```

```
9999 CALL PLOT(0.,0.,999)
```

```
IF(LOTO(1).EQ.5)STOP
```

```
9998 CONTINUE
```

```
END
```

LE MAP--(LQ=A)

-ADDRESS-	-BLOCK-	----	PROPERTIES-----	-----	TYPE-----	-----	SIZE	-NAME---	ADDRESS--	-BLOCK---
13B	/CONSTS/				REAL		3	BMSH	1750B	/THETFL/
14355B					REAL			BON	642B	//
14356B					REAL			BSTIF	0B	/PLOT1/
0B	/ALFA/				REAL			B1	14331B	
0B	/ALFAC/				REAL		140	CA	14334B	
14413B					REAL			CB	14335B	
14414B					REAL			CFLAN	14404B	
0B	/ALS/				REAL		4	CFLAN1	14470B	
4B	/ALS/				REAL		4	CJP	253B	/DADAT2/
14442B					REAL			CLDAT	275B	/DADAT2/
14443B					REAL			CLENI	27010B	/PLOT1/
14126B					REAL			CLEV	2367B	/BMFL4/
14127B					REAL			CODE	423B	//
14130B					REAL			COL	14268B	
0B	/BATS/				REAL		10	COL1	440B	/BMFL4/
5050B	/BMFL4/	EQV			REAL		240	CONSTS	0B	/CONSTS/
1775B	/THETFL/	EQV			REAL		60B	CORD	140B	/BMFL4/
3650B	/PLOT1/				REAL		196B	CORDX	14171B	
14164B					REAL			CORDY	14172B	

SUBROUTINE ARROW(ID,C1,C12,C22)

C

C

ALFA1=15./57.29578
 ARROY=.07*SIN(ALFA1)
 ARROY=.07*COS(ALFA1)
 IF(ID.EQ.2)GO TO 100
 X1=C1

Y1=C12
 X2=C1
 Y2=C22
 XC1=Y1-ARROY
 YC1=Y1+ARROY
 XC2=X1+ARROY
 YC2=YC1
 XC3=X2-ARROY
 YC3=Y2-ARROY
 XC4=X2+ARROY
 YC4=YC3
 GO TO 120

100 CONTINUE
 ARROY=ARROY
 ARROY=.07*SIN(ALFA1)
 X1=C12
 Y1=C1
 X2=C22

Y2=C1
 XC1=X1+ARROY
 YC1=Y1+ARROY
 XC2=XC1
 YC2=Y1-ARROY
 XC3=X2-ARROY
 YC3=Y2+ARROY
 XC4=XC3
 YC4=Y2-ARROY

120 CONTINUE
 CALL PLOT(Y1,Y1,3)
 CALL PLOT(XC1,YC1,2)
 CALL PLOT(X1,Y1,3)
 CALL PLOT(XC2,YC2,2)
 CALL PLOT(X2,Y2,3)
 CALL PLOT(XC3,YC3,2)
 CALL PLOT(X2,Y2,3)
 CALL PLOT(XC4,YC4,2)

RETURN
 END

LE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
132B			REAL		ID	I	DUMMY-ARG
133B			REAL		XC1	141B	
134B			REAL		XC2	143B	
2	DUMMY-ARG		REAL		XC3	145B	
3	DUMMY-ARG		REAL		XC4	147B	
4	DUMMY-ARG		REAL		X1	135B	

SUBROUTINE BORDER (XORG,YORG,XFRAME,YFRAME,WBORDER,RADBORD
1,NHCBORD,P)

```

PMIN=.007
PENTHIK=P
WIDT=AMIN1(XFRAME,YFRAME)-2.*WBORDER
IF (PENTHIK.GT.WIDT/3) PENTHIK=WIDT/3
IF (PENTHIK.LT.PMIN) PENTHIK=PMIN
FRAME=WBORDER+RADBORD
DELTAX=XFRAME-2.*FRAME
DELTAY=YFRAME-2.*FRAME
ISMOTH=25
XINCR=1.570796/ISMOTH
X=XFRAME-FRAME+XORG
Y=FRAME+YORG
NHANCE=.5*WIDT/PENTHIK
NHANCE=MIN0(NHCBORD,NHANCE)
IF (PENTHIK*(NHANCE+1).GT.WIDT/2.) PENTHIK=WIDT/2./NHANCE
DO 3 N=1,NHANCE
XSTART=XFRAME-WBORDER+(N-.5)*PENTHIK+XORG
YSTART=YFRAME-FRAME+YORG
RAD=RADBORD-(N-.5)*PENTHIK
IF (RAD.GT.0.) GO TO 5
X=XFRAME-FRAME+XORG+RAD
Y=FRAME+YORG-RAD
YSTART=YFRAME-FRAME+YORG+RAD
DELTAX=XFRAME-2.*(FRAME-RAD)
DELTAY=YFRAME-2.*(FRAME-RAD)
5 CALL PLOT(XSTART,YSTART,3)
DO 2 M=1,4
ANGLE=M*1.570796
X=X+DELTAX*COS(ANGLE)
Y=Y+DELTAY*SIN(ANGLE)
IF (RAD.GT.0.) GO TO 4
CALL PLOT(X,Y,2)
GO TO 2
4 CONTINUE
ISMOD=ISMOTH+1
DO 1 I=1,ISMOD
XANGLE=ANGLE+(I-1)*XINCR-1.570796
XX=X+RAD*COS(XANGLE)
YY=Y+RAD*SIN(XANGLE)
1 CALL PLOT(XX,YY,2)
2 CONTINUE
3 CALL PLOT(XSTART,YSTART,2)
3 CONTINUE
RETURN
END

```

IF MAP--(LDEA)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
241B			REAL		ISMOTH	226B	
224B			REAL		L	243B	
225B			REAL		M	248B	
223B			REAL		N	253B	
242B			INTEGER		NHANCE	232P	

```

SUBROUTINE DIM5(ID,J,KA,KB,IW,LVM4,LVV,SCALE,STZMX)
COMMON MAT,MEN,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NI,IFLEV,IFFRM
COMMON 7ZZZ,NPE,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IOB,CODE(2),MD(20),UF(20)
COMMON ITC1,WEZ(90)
COMMON USF(10),BCN(20),LON(20)
COMMON TAXI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/PLOT1/RSTIF(980,2),PLEN(980,2),MKB(980,2),KST(980,2),
1IPR(980,2),CSTIF(992,2),CLENI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JPR(980,2)
COMMON/RMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)
5,SIDE(10),MARG,IPL0T(25,15),JPL,IGRID1(140),IGRID(140,2)
6,DUMMY(240)
COMMON/RMFL2/SPAN(30,20),XBM(30,20,4),LBM(30,20,4),JMX(30)
COMMON/CLFL2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)
REAL MARG(10)
INTEGER COL1(140,2)
COMMON/REIZ1/KUZ,UZ1,UZ2,UZ3,UZ4,UZ5,UZ6,UZ7,UZ8,UZ9,UZ10,UZ11,
UZ12,UZ13,UZ14,UZ15,UZ16,UZ17,UZ18
DIMENSION LEVE(90)

```

C

C

```

FACTOR=.75
HGT=FACTOR+.125
LEVEL=LVM4
IF(ID.EQ.2)GO TO 550
DO 50 K=KA,KB
KK=K
30 LVN=(KK-1)*LVM4+LEVEL
IF(CX(LVN,IW).EQ.999..AND.CZ(LVN,IW).EQ.999.)GO TO 40
GO TO 60
40 CONTINUE
IF(KK.EQ.KB)LEVEL=LEVEL-1
IF(KK.EQ.KB)GO TO 50
KK=KK+1
GO TO 30
50 CONTINUE
60 CONTINUE
C22=CLEV(LEVEL)/SCALE
C1=-1.25
COL=0.
LVN=(KA-1)*LVM4+1
C12=CZ(LVN,IW)/SCALE
CALL PLOT(C1,C12,3)
CALL PLOT(C1,C22,2)
DO 130 I=2,LEVEL
C C22 YCOORD. OF LEVEL I
C C12 YCOORD. OF LEVEL I-1
C
C22=CLEV(I)/SCALE
C12=CLEV(I-1)/SCALE

```

```
      IF(I.GT.2)GO TO 110
      C1Z=CZ(LVN,IW)/SCALE
      CALL PLOT(-.125,C1Z,3)
      CALL PLOT(-1.39,C1Z,2)
110  CONTINUE
      IF(IPUG.EQ.53)PRINT 997,I,LEVEL,C1Z,C2Z
997  FORMAT(' DIM5-1',2I4,2F10.2)
      CALL ARROW(1,C1,C1Z,C2Z)
130  CONTINUE
C
C  WRITE INTERSTORY HEIGHT
C
      DO 190 I=2,LEVEL
      DO 180 K=KA,KB
      LVN1=(K-1)*LVM4+I
      IF(CZ(LVN1,IW).EQ.999.)GO TO 180
      DIMEN=ABS(CZ(LVN1,IW)-CZ(LVN1-1,IW))/UZ11
      C1Z=(CLEV(I)-DIMEN+.5+UZ11)/SCALE
      C1X=-1.297
      IF(IPUG.EQ.53)PRINT 998,I,CZ(LVN1,IW),CZ(LVN1-1,IW),UZ11,DIMEN
998  FORMAT(' DIM5-2',I4,4F10.2)
      HGT=FACTOR*.125
      ANGLE=90.
      CALL IDIM5(1,I,LEVEL,DIMEN,C1Z,C1X,ANGLE,HGT,SCALE)
      GO TO 190
180  CONTINUE
190  CONTINUE
      DO 220 K=KA,KB
      DO 210 L=2,LEVEL
      L1=LEVEL-L+2
      LVN=(K-1)*LVM4+L1
      LEVE(K)=L1
      IF(CX(LVN,IW).EQ.999..AND.CZ(LVN,IW).EQ.999.)GO TO 210
      GO TO 220
210  CONTINUE
220  CONTINUE
      NTEM=0
      DO 400 K=KA,KB
      LVN=(K-1)*LVM4+LEVE(K)
      EXD=STZMX-.5-CZ(LVN,IW)/SCALE
      IF(IPUG.EQ.53)PRINT 996,IW,K,KB,LEVE(K),STZMX,CZ(LVN,IW)/SCALE,EXD
996  FORMAT(' DIM5-3',4I4,3F12.2)
      CALL LINE5(1,K,KB,LVN,IOU,IW,EXD,SCALE)
400  CONTINUE
C
C
      LVN1=(KA-1)*LVM4+(IDOS(3)+1)
      LVN2=(KB-1)*LVM4+(IDOS(3)+2)
      C1X=CX(LVN1,IW)/SCALE
      C2X=CX(LVN2,IW)/SCALE
      LEVI=2
      DO 410 K=KA,KB
      IF(LEVE(K).GT.LEVI)LEVI=LEVE(K)
      IF(LEVI.EQ.LEVE(K))K1=K
410  CONTINUE
      LVN3=(K1-1)*LVM4+LEVI
      C1Z=CZ(LVN3,IW)/SCALE
```

```

7 C22=C12+.5
CALL PLOT(C1X,C22,3)
CALL PLOT(C2X,C22,2)
C22=C22+.375
CALL PLOT(C1X,C22,3)
CALL PLOT(C2X,C22,2)
CALL ARROW(2,C22,C1X,C2X)
DIMEN=0.
DIMEN1=0.
DO 500 K=KA,KB
LVNN=K+LVM4+(IDOS(3)+1)
LVN=(K-1)+LVM4+(IDOS(3)+1)
C32=C12+.5
C22=C12+.5
C1X=CX(LVN,IW)/SCALE
IF(K.EQ.KP)GO TO 415
C2X=CX(LVNN,IW)/SCALE
CALL ARROW(2,C22,C1X,C2X)
DIMEN=1./(PLEN(LVN,IW)*U711)
DIMEN1=DIMEN+DIMEN1
GO TO 420
415 CONTINUE
DIMEN=DIMEN1
C22=C22+.375
C1X=C1X-DIMEN*U211/SCALE
420 CONTINUE
ANGLE=0.
HGT=FACTOR*.125
CALL IDIM5(2,K,KB,DIMEN,C32,C1X,ANGLE,HGT,SCALE)
500 CONTINUE
GO TO 1000

C
C FLOOR PLAN DIMENSIONS
C
550 CONTINUE
C
C FIND GRID LINE FOR X/Y DIR. PER FLOOR
C
CALL ZERO(IGRID,280)
IGRX1=AMAX0(NCOL(1),NCOL(2))
IGRY1=AMAX0(NCOL(1),NCOL(2))
IGRY2=0
IGRX2=0
MB=MB1(1)
IW2=2
IFR=IFR2(1)
DO 600 IF=1,IFR
KAA=COL1(IF,1)
IF((K.LT.IFR)&KBB=COL1(IF+1,1))-1
IF((IF.EQ.IFR)&KBB=MB)
DO 600 K=KAA,KBB
LVN=(K-1)+LVM4+LVV
IF((CX(LVN,1).EQ.999..AND.CZ(LVN,1).EQ.999.)GO TO 600
MK1=MKC(K,1)
IGRX=IGRID(MK1)/100
IGRY=MOD(IGRID(MK1),100)
IF(IGRY.EQ.0.OR.IGRX.EQ.0)GO TO 600

```

```
IGRY=AMAX0(NCOL(1),NCOL(2))-IFR+IGR1
```

```
IF(IGRX.LT.IGRX1)IGRX1=IGRX
```

```
IF(IGRX.GT.IGRX2)IGRX2=IGRX
```

```
IF(IGRY.LT.IGRY1)IGRY1=IGRY
```

```
IF(IGRY.GT.IGRY2)IGRY2=IGRY
```

```
IGRID(IGRX,1)=1
```

```
IGRID(IGRY,2)=1
```

```
IF(IBUG.EQ.53)WRITE(6,*(='DIMSP1+',2I4)*)
```

```
*MK1,K,IGRX1,IGRY1,IGRX2,IGRY2,IGRID(IGRX,1),IGRID(IGRY,1)
```

```
600 CONTINUE
```

```
650 CONTINUE
```

```
C
```

```
C FIND OUT IF FIRST/LAST COL OF EACH FRAME IS LOCATED
```

```
C INSIDE LIMITS IF IT IS SO CALL GRIDL OTHERWISE
```

```
C DRAW GRID LINES CALLING LINES
```

```
C
```

```
DO 700 I=1,2
```

```
MB=MB1(I)
```

```
IM2=INVER(I)
```

```
IFR=IFR2(I)
```

```
DO 700 IF=1,IFR
```

```
KAA=COL1(IF,I)
```

```
IF(IF.LT.IFR)KBB=COL1(IF+1,I)-1
```

```
IF(IF.EQ.IFR)KBB=MB
```

```
DO 680 K=KAA,KBB
```

```
K1=K
```

```
IF(I.EQ.2)K1=KBB-K+KAA
```

```
LVN=(K1-1)*LVH4+LVV
```

```
IF(CX(LVN,I).EQ.999..AND.CZ(LVN,I).EQ.999.)GO TO 680
```

```
MK1=MKC(K1,I)
```

```
IGRX=IGRID1(MK1)/100
```

```
IGR1=MOD(IGRID1(MK1),100)
```

```
IF(IGR1.EQ.0.OR.IGRX.EQ.0)GO TO 680
```

```
IGRY=AMAX0(NCOL(1),NCOL(2))-IFR2(I)+IGR1
```

```
IF(IBUG.EQ.53)WRITE(6,*(='DIMSP2+',10I5,F10.3)*)
```

```
*MK1,K1,I,IF,IGRX,IGRY,IGRX1,IGRY1,IGRX2,IGRY2
```

```
IF(I.EQ.2)GO TO 655
```

```
IF(IGRY.EQ.IGRY2)GO TO 660
```

```
IF(IGRY.EQ.IGRY1)GO TO 660
```

```
IF(IGRX.EQ.IGRY1)GO TO 660
```

```
GO TO 658
```

```
655 CONTINUE
```

```
IF(IGRX.EQ.IGRX1)GO TO 660
```

```
IF(IGRX.EQ.IGRX2)GO TO 660
```

```
IF(IGRY.EQ.IGRY1)GO TO 660
```

```
658 CONTINUE
```

```
IF(IBUG.EQ.53)WRITE(6,*(='DIMSP1-1+',11I5)*)
```

```
*MK1,I,K,KAA,KBB,IGRX,IGRY,IGRX1,IGRY1,IGRX2,IGRY2
```

```
K1=KBB
```

```
IF(I.EQ.2)K1=KAA
```

```
CALL GRIDL(MK1,I,K,K1,IGRX,IGRY,LVH4,LVV)
```

```
660 CONTINUE
```

```
TOTAL=STZMX+.5
```

```
IF(I.EQ.2)TOTAL=STPY1/SCALE+MARG(9)-.5
```

```
IF(IBUG.EQ.53)WRITE(6,*(='DIMSP11+',10I5,F10.3)*)
```

```
*MK1,K1,I,IF,IGRX,IGRY,IGRX1,IGRY1,IGRX2,IGRY2,TOTAL
```

```
CALL LINE5(2,IGRX,IGRY,MK1,LVV,I,TOTAL,SCALE)
```

60 TO 700

680 CONTINUE

700 CONTINUE

C

C DIMENSION LINES TOTAL X/Y DIRECTION.

C

DO 750 I=1,2

HGT=FACTOR*125

IF(I.EQ.2) 60 TO 720

ANGLE=90.

CX1=GRID1(IGRX1,1)/SCALE-1.25

CY1=GRID1(IGRY2,2)/SCALE

CY2=GRID1(IGRY1,2)/SCALE

DIMEN=(GRID1(IGRY1,2)-GRID1(IGRY2,2))/UZ11

CALL PLOT(CY1,CY1,3)

CALL PLOT(CX1,CY2,2)

CALL ARROW(1,CX1,CY1,CY2)

CALL IDIM5(3,IGRY1,IGRY2,DIMEN,CX1,CY1,ANGLE,HGT,SCALE)

CX1=CX1+.375

CALL PLOT(CX1,CY1,3)

CALL PLOT(CX1,CY2,2)

60 TO 750

720 CONTINUE

ANGLE=0.

DIMEN=(GRID1(IGRX2,1)-GRID1(IGRX1,1))/UZ11

CX1=GRID1(IGRX1,1)/SCALE

CX2=GRID1(IGRX2,1)/SCALE

CY1=GRID1(IGRY1,2)/SCALE+1.25

CALL PLOT(CX1,CY1,3)

CALL PLOT(CX2,CY1,2)

CALL ARROW(2,CY1,CX1,CX2)

CALL IDIM5(2,IGRX1,IGRX2,DIMEN,CY1,CX1,ANGLE,HGT,SCALE)

CY1=CY1-.375

CALL PLOT(CX1,CY1,3)

CALL PLOT(CX2,CY1,2)

750 CONTINUE

C

C DIMENSION LINES BETWEEN GRID LINES AND ARROW HEADS

C

DO 800 I=1,2

MB=MB1(I)

IFR=IFR2(I)

IN2=INVER(I)

DIMEN=0.

CORD2=0.

DO 800 IF=1,IFR

KAA=COL1(IF,I)

IF(1F.LT.IFR)KBB=COL1(IF+1,I)-1

IF(1F.EQ.IFR)KBB=MB

DO 780 K=KAA,KBB

K1=K

IF(1.EQ.2)K1=KBB-K+KAA

LVN=(K1-1)*LVN4+LVV

IF(CX(LVN,I).EQ.999..AND.CZ(LVN,I).EQ.999.) 60 TO 780

PK=MKC(K1,I)

IGRK=IGRID1(MK)/100

IGR1=MOD(IGRID1(MK),100)


```
IF (IGR1.EQ.0.OR.IGRX.EQ.0)GO TO 780
IGRY=AMAX0(MCOL(1),MCOL(2))-IFR2(1)+IGR1
HGT=FACTOR+.125
IF (IFR.EQ.IF)GO TO 780
IF (I.EQ.2)GO TO 760
IS=3
ID=1
CORD2=CORD(MK,IW2)
IF (IGRY.EQ.IGRY1)GO TO 800
IYY=IGRY-1
755 CONTINUE
IF (IYY.LT.IGRY1)GO TO 800
IF (IGRID(IYY,2).NE.0)GO TO 757
IYY=IYY-1
GO TO 755
757 CONTINUE
CORD1=GRID1(IYY,2)
IGR1=IGRY1
IGR2=IGRY2
IGR=IGRY
ANGLE=90.
CX1=GRID1(IGRX1,1)/SCALE+.875
CY1=CORD2/SCALE
CY2=CORD1/SCALE
GO TO 770
760 CONTINUE
IS=2
ID=2
CORD2=CORD(MK,IW2)
IF (IGRX.EQ.IGRX2)GO TO 800
IXX=IGRX+1
765 CONTINUE
IF (IXX.GT.IGRX2)GO TO 800
IF (IGRID(IXX,1).NE.0)GO TO 767
IXX=IXX+1
767 CONTINUE
CORD1=GRID1(IXX,1)
IGR1=IGRX2
IGR2=IGRX1
IGR=IGRX
ANGLE=0.
CX1=GRID1(IGRY1,2)/SCALE+.875
CY1=CORD2/SCALE
CY2=CORD1/SCALE
770 CONTINUE
IF (IBUG.EQ.55)WRITE(6,*(1+DIM5PS+.6I4,2F15.3)*)
*MK,1,IGRX,IGRY,IGRX1,IGRY1,CORD1,CORD2
DIMEN=(CORD1-CORD2)/UZ11
CALL IDIMS(15,IGR,IGR1,DIMEN,CX1,CY1,ANGLE,HGT,SCALE)
CALL ADDGRID,CX1,CY1,CY2)
GO TO 800
780 CONTINUE
800 CONTINUE
IF (IPLOT(16,J).EQ.0)GO TO 1000
DO 800 I=1,2
IMX=INVERC(I)
IFR3=IFR2(IW2)
```

```
IFR1=IFR2(I)
DO 800 ISE=1,IFR1
MGT=FACTOR*.125
IVOR=0
KAS=COL1(ISE,I)
IF(ISE.LT.IFR1)KBS=COL1(ISE+1,I)-1
IF(ISE.EQ.IFR1)KBS=MR1(I)
IF(I.EQ.2)GO TO 820
DO 815 K=KAS,KBS
LVNC=(K-1)*LVH4+LVV
IF(IBM6.EQ.53)PRINT 990,KAS,KBS,K,I,LVNC,CX(LVNC,I),CZ(LVNC,I)
IF(CX(LVNC,I).EQ.999..AND.CZ(LVNC,I).EQ.999.)GO TO 815
990 FORMAT(' DIM5-S1',5I4,2F20.3)
KAS=K
GO TO 816
815 CONTINUE
GO TO 820
816 CONTINUE
MK=MKC(KAS,I)
LX1=LY(MK)
JX1=JY(MK)
GO TO 825
820 CONTINUE
DO 822 K=KAS,KBS
K1=KBS-K+KAS
LVNC=(K1-1)*LVH4+LVV
IF(IBM6.EQ.53)PRINT 990,KAS,KBS,K1,I,LVNC,CX(LVNC,I),CZ(LVNC,I)
IF(CX(LVNC,I).EQ.999..AND.CZ(LVNC,I).EQ.999.)GO TO 822
KBS=K1
GO TO 823
822 CONTINUE
GO TO 880
823 CONTINUE
MK=MKC(KBS,I)
LX1=LX(MK)
JX1=JX(MK)
825 CONTINUE
IF(CX(LVNC,I).EQ.999..AND.CZ(LVNC,I).EQ.999.)GO TO 880
IF(IBM6.EQ.53)PRINT 999,I,MK,LX1,JX1
IF(LX1.EQ.0)GO TO 880
IF(JX1.EQ.1)GO TO 850
IF(SPAN(LX1,JX1-1).LE.0.)GO TO 850
C
C COL. IN MIDDLE OF RM LINE
C
DO 830 ISE1=1,IFR3
KAS1=COL1(ISE1,IW2)
IF(ISE1.LT.IFR3)KBS1=COL1(ISE1+1,IW2)-1
IF(ISE1.EQ.IFR3)KBS1=MR1(IW2)
DO 827 K=KAS1,KBS1
MK1=MKC(K,IW2)
IF(MK.EQ.MK1)GO TO 834
827 CONTINUE
830 CONTINUE
GO TO 880
834 CONTINUE
MK1=K-1
```

```
      MK2=MKC(MK1,IM2)
      IF(MK2.EQ.MK3)GO TO 850
      LX3=LX(MK3)
      JX3=JX(MK3)
      IF(SPAN(LX3,JX3).LE.0.)GO TO 860
      LX2=LX1
      JX2=JX1-1
      GO TO 860
850 CONTINUE
      IF(SPAN(LX1,JX1).LE.0.)GO TO 880
      LX2=LX1
      JX2=JX1
      MK2=MK
860 CONTINUE
      DBM=DBM1=0.
      VAL=0.
      IF(IPL0T(14,J).EQ.1)VAL=.052
      DO 870 KN=1,8
      NGT=FACTOR*.125
      IJL=JLBM5(LX2,JX2,KN)
      IF(1BUG.EQ.53)PRINT 999,I,MK,LX2,JX2,IJL,MK2,MK3
999 FORMAT(' DIM5-S*,7I5)
      IF(IJL.EQ.0.AND.MK.EQ.MK2)GO TO 875
      IF(IJL.EQ.0)GO TO 850
      DBM=FXBM5(LX2,JX2,KN)/12.
      LI1=IJL/100
      IS1=IJL-LI1*100
      LI2=LX(MK)
      IS2=JX(MK)
      IF(I.EQ.2)LI2=LY(MK)
      IF(I.EQ.2)IS2=JY(MK)
      SP2=SPAN(LI2,IS2)/12.
      SP1=SPAN(LI1,IS1)/12.
      SP=SP1-SP2
      IF(SP2.GT.SP1)SP=0.
      IF(SP2.EQ.0.)SP=DBM
      SIN1=SP/DBM
      ALF=ASIN(SIN1)
      COS1=COS(ALF)
      IF(1BUG.EQ.53)WRITE(6,*(='DIM5P66*',0I5,6F10.3)*)
      *MK,I,LI1,IS1,LX2,JX2,LI2,IS2,SP1,SP2,SP,DBM,SIN1,COS1
      IF(I.EQ.2)GO TO 865
      DBM=DBM+COS1
      CX1=CORD(MK2,I)/SCALE-.5
      CY1=CORD(MK2,IM2)/SCALE+DBM1/SCALE
      DBM1=DBM1+DBM
      CY2=CORD(MK2,IM2)/SCALE+DBM1/SCALE
      CX2=CORD(MK2,I)/SCALE-.125-VAL
      CX3=CX2-.5
      IS=5
      ID=1
      ANGLE=90.
      CALL PLOT(CX1,CY1,3)
      CALL PLOT(CX1,CY2,2)
      CALL ARROW(ID,CX1,CY1,CY2)
      CALL PLOT(CX2,CY2,3)
      CALL PLOT(CX3,CY2,2)
```

```

      HGT=FACTOR*.125
      DBM2=DBM/UZ11
      CALL IDIM5(IIS,IWOR,MK,DBM2,CX1,CY1,ANGLE,HGT,SCALE)
      GO TO 870
265  CONTINUE
      DRN=DBM*SIN1
      CX1=CORD(MK2,IW2)/SCALE+DBM1/SCALE
      CY1=CORD(MK2,I)/SCALE+.5
      DBM1=DRN+DBM1
      CX2=CORD(MK2,IW2)/SCALE+DBM1/SCALE
      CY2=CORD(MK2,I)/SCALE+.125+VAL
      CY3=CY2+.5
      ID=2
      IS=4
      ANGLE=0.
      CALL PLOT(CX1,CY1,3)
      CALL PLOT(CX2,CY1,2)
      CALL ARROW(ID,CY1,CX1,CX2)
      CALL PLOT(CX2,CY2,3)
      CALL PLOT(CX2,CY3,2)
      HGT=FACTOR*.125
      DBM2=DBM/UZ11
      CALL IDIM5(IIS,IWOR,MK,DBM2,CY1,CX1,ANGLE,HGT,SCALE)
      870  CONTINUE
      875  CONTINUE
      IF(MK.EQ.MK2)MK3=MK2
      IF(MK.EQ.MK2)GO TO 880
      GO TO 850
      880  CONTINUE
      1000 CONTINUE
      RETURN
      END

```

LE MAP--(LO=A)

-ADDRESS--	BLOCK----	PROPERTIES-----	TYPE-----	SIZE	-NAME---	ADDRESS--	BLOCK---
38478			REAL		CY2	27788	
27238			REAL		CY3	38538	
36588	/PLOT1/		REAL	1968	CZ	366188	/PLOT1/
6428	//		REAL	28	C1	27118	
88	/PLOT1/		REAL	1968	C1X	27228	
278188	/PLOT1/		REAL	1984	C1Z	27138	
23678	/BNFL4/		REAL	98	C11	525768	/PLOT1/
4238	//		REAL	2	C2X	27338	
27128		*S*	REAL		C2Z	27188	
4428	/BNFL4/		INTEGER	288	C22	526888	/PLOT1/
188	/BNFL4/		REAL	288	C3Z	27428	
38888			REAL		DAT	558	//
27728			REAL		DBM	38538	
38588			REAL		DBM1	38528	
231188	/PLOT1/		REAL	1984	DBM2	38528	
327188	/PLOT1/		REAL	1984	DIREN	27218	
27668			REAL		DIREN1	27378	
27718			REAL		DUMMY	58588	/BNFL4/
38518			REAL		DV	464128	/PLOT1/
27678			REAL		EC1	1158	//

```

      GO TO 300
100 ENCODE(4,321,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
110 ENCODE(5,322,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
120 ENCODE(6,323,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
130 ENCODE(7,324,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
140 ENCODE(5,331,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
150 ENCODE(6,332,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
160 ENCODE(7,333,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
170 ENCODE(8,334,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
180 ENCODE(7,342,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
190 ENCODE(6,343,BIMSH(LM,JM))IWD,ISIGN,IDEF
      GO TO 300
200 ENCODE(9,344,BIMSH(LM,JM))IWD,ISIGN,IDEF
300 CONTINUE
      IF(IBUG.EQ.53)PRINT 999,LM,JM,IWD,IDEF,LETV,LETD,BIMSH(LM,JM)
310 CONTINUE
350 CONTINUE

```

C
C

```

311 FORMAT(I1,A1,I1)
312 FORMAT(I1,A1,I2)
313 FORMAT(I1,A1,I3)
321 FORMAT(I2,A1,I1)
322 FORMAT(I2,A1,I2)
323 FORMAT(I2,A1,I3)
324 FORMAT(I2,A1,I4)
331 FORMAT(I3,A1,I1)
332 FORMAT(I3,A1,I2)
333 FORMAT(I3,A1,I3)
334 FORMAT(I3,A1,I4)
342 FORMAT(I4,A1,I2)
343 FORMAT(I4,A1,I3)
344 FORMAT(I4,A1,I4)
999 FORMAT(' ENCO ',6I6,1X,A10)

```

C

RETURN
END

LE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
50500	/BMFL4/	EOV	REAL	240	DEPTH	1300	/BMFL1/
17750	/THETFL/	EOV	REAL	600	DUM	00	/THETFL/
17500	/THETFL/	EOV	REAL	21	DUMMY	50500	/BMFL4/
6420	//		REAL	20	DUM1	00	/BMFL4/
4230	//		REAL	2	EC1	1150	//
550	//		REAL	3	EC9	1100	//

```

SUBROUTINE ENCO(LIG)
COMMON MAT,MFM,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IO9,CODE(2),MD(20),UF(20)
COMMON ITC1,NEZ(90)
COMMON USF(10),BON(20),LON(20)
COMMON TAKI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/REI21/KU2,UZ1,UZ2,UZ3,UZ4,UZ5,UZ6,UZ7,UZ8,UZ9,UZ10,UZ11,
.UZ12,UZ13,UZ14,UZ15,UZ16,UZ17,UZ18
COMMON/RMFL1/WIDTH(30,20),DEPTH(30,20)
COMMON/RMFL2/SPAN(30,20),XBH(30,20,4),LBH(30,20,4),JMX(30)
COMMON/TMTEL/DUM(1890)
COMMON/RMFL4/DUM1(2600),DUMMY(240)
DIMENSION IFBB(250),XB1(250),YB1(250),MKP1(250),RMSH(21)
1,BIMSH(30,20),OB2(250),BIM(240)
EQUIVALENCE (DUM(1),IFBB(1)),(DUM(251),XB1(1)),(DUM(501),YB1(1))
EQUIVALENCE (DUM(751),MKP1(1))
EQUIVALENCE (DUM(1001),RMSH(1)),(DUM(1022),BIMSH(1,1))
EQUIVALENCE (DUM(1622),OB2(1)),(DUM(1872),INUMX)
EQUIVALENCE (DUMMY(1),BIM(1))
IBUG=LET(11)
CALL ZERO(BIMSH,600)
ISIGN=1H*
DO 350 LM=1,LIG
JMX1=JMX(LM)-1
DO 310 JM=1,JMX1
IVID=WIDTH(LM,JM)/UZ10+.01
IDEP=DEPTH(LM,JM)/UZ10+.01
LETU=1
LETD=1
IF(IVID.GT.9.AND.IVID.LE.99)LETU=2
IF(IVID.GT.99.AND.IVID.LE.999)LETU=3
IF(IVID.GT.999.AND.IVID.LE.9999)LETU=4
IF(IDEP.GT.9.AND.IDEP.LE.99)LETD=2
IF(IDEP.GT.99.AND.IDEP.LE.999)LETD=3
IF(IDEP.GT.999.AND.IDEP.LE.9999)LETD=4
GO TO(20,30,40,50)LETU
CALL GOTOER
20 CONTINUE
GO TO(60,70,80)LETD
CALL GOTOER
30 CONTINUE
GO TO(100,110,120,130)LETD
CALL GOTOER
40 CONTINUE
GO TO(140,150,160,170)LETD
CALL GOTOER
50 CONTINUE
GO TO(180,190,190,200)LETD
CALL GOTOER
60 ENCODE(3,311,BIMSH(LM,JM))IVID,ISIGN,IDEP
GO TO 300
70 ENCODE(4,312,BIMSH(LM,JM))IVID,ISIGN,IDEP
GO TO 300
80 ENCODE(5,313,BIMSH(LM,JM))IVID,ISIGN,IDEP

```

SUBROUTINE FRAME(IPLA,ICOUNT,MCOUNT,KAK,KB,IV,LVM4,STZMN,ORGX,ORGY,
*SCALE,ORGX1,ORGY1,IPS)

COMMON MAT,MEM,LEV,ISOLV,LVIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAYFR,ITEST,IPUN,LVU,LNU
COMMON KAK,LIST(10),IO9,CODE(2),MN(20),UF(20)
COMMON ITC1,NEZ(90)

COMMON USE(10),BON(20),LON(20)

COMMON TAKI(10),JET(10),KET(20),LET(20)

INTEGER KAK(90,2),NME(11,3)

COMMON/PLOT1/PSTIF(990,2),BLEN(980,2),MKR(980,2),KST(980,2),
1IBR(980,2),CSTIF(992,2),CLEN1(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JRR(990,2)

COMMON/BMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)
5,SIDE(10),MARG,IPLLOT(25,15),JPL,IGRID1(140),IGRID(140,2),
6,DUMMY(240)

INTEGER COL1(140,2)

DIMENSION ADIM(30),BDIM(30)

REAL MARG(10)

DATA MARG/5*2.,1.75,1.45,2.125,2*2.5/

SIZX=SIZX1=SIZZ=SIZZ1=0.

SIDE3=SIDE(IPS)-MARG(4)

SIDE4=SIDE(IPS+1)-MARG(5)-4.5

IF(IPLA.EQ.1)GO TO 100

DO 10 IIP=KA,KB

LN=(IIP-1)*LVM4+IDOS(3)+1

LN1=IIP*LVM4

LN2=(IIP-1)*LVM4+1

IF(CX(LN,IV).GT.SIZX)SIZX=CX(LN,IV)

IF(CX(LN1,IV).EQ.999..AND.CZ(LN1,IV).EQ.999.)GO TO 10

IF(CZ(LN1,IV).GT.SIZZ)SIZZ=CZ(LN1,IV)

IF(CZ(LN2,IV).LT.SIZZ1)SIZZ1=CZ(LN2,IV)

10 CONTINUE

SIZX=SIZX/SCALE

SIZZ1=SIZZ1/SCALE

SIZZ=SIZZ/SCALE

IF(IBUG.EQ.53)

*PRINT 991,ICOUNT,MCOUNT,KAK,LN,SIZX1,SIZX,SIZZ1,SIZZ,

*STZMN,ORGX,ORGY

991 FORMAT(' ** DEBUG LOCATION 1 ***',4I4,7F12.2)

IF(MCOUNT.NE.998)GO TO 12

MCOUNT=0

GO TO 15

12 CONTINUE

IF(ICOUNT.GT.1)GO TO 30

15 CONTINUE

ADIM(ICOUNT)=SIZX+MARG(1)+MARG(6)

BDIM(ICOUNT)=STZZ/SCALE-STZMN+MARG(2)+MARG(8)

IF(ADIM(ICOUNT).GT.SIDE4)GO TO 20

IF(BDIM(ICOUNT).GT.SIDE3)GO TO 20

ORGX=MARG(1)+MARG(6)

ORGY=SIDE(IPS)-MARG(2)-MARG(8)-STZZ/SCALE

KCOUNT=ICOUNT

SIZX=SIZX

```
      IF(MCOUNT.NE.998)SIZEX1=SIZEX
      ORGX2=0.
      GO TO 80
20  CONTINUE
      C
      C PAPER SIZE TOO SMALL
      C
      MCOUNT=199
      RETURN
      30  CONTINUE
      C
      C OTHER FRAMING ELEV. SIDE BY SIDE
      C
      ORGX2=0*GRX2+ORGX
      ADIM(ICOUNT)=ADIM(ICOUNT-1)+MARG(1)+MARG(6)+SIZX
      BDIM(ICOUNT)=MARG(2)+MARG(8)+SIZZ-STZMN
      IF(ADIM(ICOUNT).GT.SIDE4)GO TO 40
      IF(BDIM(ICOUNT).GT.SIDE5)GO TO 50
      C
      C SET NEW ORIGIN FOR FRAMING ELEV.
      C
      ORGX=SIZEX+MARG(1)+MARG(6)
      C
      ORGY=SIDE(3)-SIZZ+STZMN-(MARG(2)+MARG(8))-ORGX
      ORGY=0.
      SIZEY=SIZEX
      MCOUNT=0
      GO TO 80
      C
      C TRY TO LOCATE BELOW
      C
      C
      40  CONTINUE
      ADIM(ICOUNT)=MARG(1)+MARG(6)+SIZX
      IF(ADIM(ICOUNT).GT.SIDE4)GO TO 50
      BDIM(ICOUNT)=BDIM(KCOUNT)+BDIM(ICOUNT)
      IF(BDIM(ICOUNT).GT.SIDE3)GO TO 50
      KCOUNT=ICOUNT
      ORGX=MARG(1)+MARG(6)-ORGX2
      ORGY=-(MARG(2)+MARG(8)+SIZZ/SCALE-STZMN)
      SIZEX=SIZEX
      MCOUNT=0
      GO TO 80
      50  CONTINUE
      C
      C NEW PAGE
      C
      MCOUNT=998
      GO TO 15
      80  CONTINUE
      ADIM1=ADIM(ICOUNT)
      BDIM1=BDIM(ICOUNT)
      IF(MCOUNT.NE.998)GO TO 85
      ADIM1=ADIM(ICOUNT-1)
      BDIM1=BDIM(ICOUNT-1)
      85  CONTINUE
      XLEFT=SIDE(IPS+1)-ADIM1-4.5
      YLEFT=SIDE(IPS)-BDIM1
      IF(MCOUNT.NE.998)SIZEX1=SIZEX
```



```

    ORGX1=XLEFT+SIZE1+4.5
    ORGY1=STZMN-YLEFT
    IF(1PUG.EQ.53)
    1PRINT 990,ICOUNT,MCOUNT,KCOUNT,ADIM(ICOUNT),BDIM(ICOUNT),
    2STZMA,ORGX,ORGY,ORGX1,ORGY1,ORGX2,XLEFT,YLEFT
    990 FORMAT(' ** DEBUG LOCATION **',3I4,10F8.2)
    RETURN
    100 CONTINUE

```

```

C
C FLOOR PLANS
C

```

```

    SPX=SPX1=SPY=SPY1=0.
    DO 110 II=1,2
    MB=MB1(II)
    IFR=IFR2(II)
    DO 110 III=1,IFR
    KA1=COL1(III,II)
    IF(III.LT.IFR)KB1=COL1(III+1,II)-1
    IF(III.EQ.IFR)KB1=MB
    DO 110 KK=KA1,KB1
    MK=MKC(KK,II)
    LVR1=KA
    LN=(KK-1)*LVM4+LVR1
    IF(1CFL(MK).NE.0)GO TO 105
    IF(CX(LN,II).EQ.999..AND.CZ(LN,II).EQ.999.)GO TO 110
    105 CONTINUE
    IF(CORD(MK,1).LT.SPX)SPX=CORD(MK,1)
    IF(CORD(MK,1).GT.SPX1)SPX1=CORD(MK,1)
    IF(CORD(MK,2).LT.SPY)SPY=CORD(MK,2)
    IF(CORD(MK,2).GT.SPY1)SPY1=CORD(MK,2)
    110 CONTINUE
    SPX=SPX/SCALE
    SPX1=SPX1/SCALE
    SPY=SPY/SCALE
    SPY1=SPY1/SCALE
    IF(1BUG.EQ.53)
    *PRINT 991,ICOUNT,MCOUNT,LVR1,LVM4,SPX,SPX1,SPY1,SPY,
    *STPX,ORGX,ORGY
    IF(MCOUNT.NE.998)GO TO 112
    MCOUNT=0
    GO TO 115
    112 CONTINUE
    IF(ICOUNT.GT.1)GO TO 130
    115 CONTINUE
    ADIM(ICOUNT)=SPX1-SPX+MARG(9)+MARG(1)
    BDIM(ICOUNT)=SPY1-SPY+MARG(10)+MARG(2)
    IF(ADIM(ICOUNT).GT.SIDE4)GO TO 120
    IF(BDIM(ICOUNT).GT.SIDE3)GO TO 120
    ORGX=MARG(9)+MARG(1)-SPX
    ORGY=SIDE(IPS)-MARG(10)-SPY1-MARG(2)
    KCOUNT=ICOUNT
    SIZE1=SPX1-SPX
    IF(MCOUNT.NE.998)SIZE1=SIZE1
    ORGX2=0.
    GO TO 180
    120 CONTINUE

```

```

C

```

C PAPER SIZE TOO SMALL

C

MCOUNT=999

RETURN

130 CONTINUE

C

C OTHER FRAMING PLAN SIDE BY SIDE

C

ORGX2=ORGX2+ORGX

ADIM(ICOUNT)=ADIM(ICOUNT-1)+MARG(9)+MARG(1)+SPX1-SPX

BDIM(ICOUNT)=MARG(10)+SPY1-SPY+MARG(2)

IF(ADIM(ICOUNT).GT.SIDE4)GO TO 140

IF(BDIM(ICOUNT).GT.SIDE3)GO TO 150

C

C SET NEW ORIGIN FOR FRAMING PLAN

C

ORGX=SIZEX+MARG(9)+MARG(1)

ORGY=0.

SIZEX=SPX1-SPX

MCOUNT=1

GO TO 180

C

C TRY TO LOCATE BELOW

C

140 CONTINUE

ADIM(ICOUNT)=MARG(9)+MARG(1)+SPX1-SPX

IF(ADIM(ICOUNT).GT.SIDE4)GO TO 150

BDIM(ICOUNT)=BDIM(KCOUNT)+BDIM(ICOUNT)

IF(BDIM(ICOUNT).GT.SIDE3)GO TO 150

KCOUNT=ICOUNT

ORGX=MARG(9)+MARG(1)-ORGX2

ORGY=-((MARG(10)+MARG(2)+(STPY1-STPY)/SCALE)-1.

SIZEX=SPX1-SPX

MCOUNT=0

GO TO 180

150 CONTINUE

C

C NEW PAGE

C

MCOUNT=998

GO TO 115

180 CONTINUE

ADIM1=ADIM(ICOUNT)

BDIM1=BDIM(ICOUNT)

IF(MCOUNT.NE.998)GO TO 185

ADIM1=ADIM(ICOUNT-1)

BDIM1=BDIM(ICOUNT-1)

185 CONTINUE

XLEFT=SIDE(IPS+1)-ADIM1+SPX-4.5

YLEFT=SIDE(IPS)-BDIM1

IF(MCOUNT.NE.998)SIZEX1=SIZEX

ORGX1=XLEFT+SIZEX1+4.5

ORGY1=STPY/SCALE-YLEFT

IF(1/BUG.EQ.53)

IPRINT 998,ICOUNT,MCOUNT,KCOUNT,ADIM(ICOUNT),BDIM(ICOUNT),

2STPY/SCALE,ORGX,ORGY,ORGX1,ORGY1,ORGX2,XLEFT,YLEFT

RETURN

END

LE MAP--(LO=A)

-ADDRESS-	-BLOCK-	-PROPERTIES-	-TYPE-	-SIZE-	-NAME-	-ADDRESS-	-BLOCK-
671B			REAL	30	IRUN	121B	//
1003B			REAL		ISOLV	3B	//
727B			REAL	30	ITC1	475B	//
1004B			REAL		ITEST	120B	//
3650B	/PLOT1/		REAL	1960	IV	6	DUMMY-AR
642B	//		REAL	20	JBR	52602B	/PLOT1/
0B	/PLOT1/		REAL	1960	JET	724B	//
27010B	/PLOT1/		REAL	1984	JFL	4203B	/BMFL4/
2367B	/BMFL4/		REAL	90	JSTOP	111B	//
423B	//		REAL	2	KA	4	DUMMY-AR
440B	/BMFL4/		INTEGER	280	KAK	124B	//
10B	/BMFL4/		REAL	280	KA1	1017B	
23110B	/PLOT1/		REAL	1984	KB	5	DUMMY-AR
32710B	/PLOT1/		REAL	1984	KP1	1020B	
36610B	/PLOT1/		REAL	1984	KCOUNT	777B	
52576B	/PLOT1/		REAL	2	KET	736B	//
52600B	/PLOT1/		REAL	2	KK	1021B	
55B	//		REAL	3	KKPLOT	3154B	/BMFL4/
5050B	/BMFL4/		REAL	240	KST	13370B	/PLOT1/
46412B	/PLOT1/		REAL	1984	LET	762B	//
115B	//		REAL		LEV	2B	//
116B	//		REAL		LIST	410B	//
1305B	/BMFL4/		REAL	280	LN	774B	
17240B	/PLOT1/		INTEGER	1960	LNFIN	5B	//
2521B	/BMFL4/		INTEGER		LNMAX	7B	//
1070B	/BMFL4/		INTEGER	140	LNH	123B	//
2	DUMMY-ARG		INTEGER		LN1	775B	
103B	//		INTEGER	6	LN2	776B	
75B	//		INTEGER	6	LON	666B	//
74B	//		INTEGER		LVFIN	4B	//
60B	//		INTEGER	6	LVMAX	6B	//
66B	//		INTEGER	6	LVH4	7	DUMMY-AR
1304B	/BMFL4/		INTEGER		LVR1	1023B	
12B	//		INTEGER		LVU	122B	//
2740B	/BMFL4/		INTEGER	140	MARG	3402B	/BMFL4/
2524B	/BMFL4/		INTEGER	140	MAT	0B	//
11B	//		INTEGER		MAXFR	117B	//
1015B			INTEGER		MP	1014B	
6B	/BMFL4/		INTEGER	2	M81	4B	/BMFL4/
4420B	/BMFL4/		INTEGER	280	MCOUNT	3	DUMMY-AR
4204B	/BMFL4/		INTEGER	140	MD	425B	//
1013B			INTEGER		MEM	1B	//
1016B			INTEGER		MK	1022B	
773B			INTEGER		MKB	7520B	/PLOT1/
114B	//		INTEGER		MKC	1737B	/BMFL4/
42510B	/PLOT1/		INTEGER	2	NCOL	1735B	/BMFL4/
422B	//		INTEGER		NEZ	476B	//
1	DUMMY-ARG		INTEGER		NL	18B	//
3414B	/BMFL4/		INTEGER	375	NME	14B	//
113B	//		INTEGER		NSTOP	112B	//
14	DUMMY-ARG		INTEGER		ORGX	9	DUMMY-AR

```

SUBROUTINE GRIDL(MK,IM,K,K1,IGRX,IGRY,LVM4,LVV)
COMMON MAT,MFM,LEV,ISOLV,LVF IN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IPUN,LVU,LNU
COMMON KAK,LIST(10),909,CODE(2),MD(28),UF(20)
COMMON ITC1,NEZ(90)
COMMON USF(10),BON(20),LON(20)
COMMON TAKI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/PLOT1/BSTIF(980,2),BLEN(980,2),MKR(980,2),KST(980,2),
1IBR(980,2),CSTIF(992,2),CLEI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PMIC(992,2),OV(992,2),TRANS(90,2),C11(2),C22(2),
3JBR(980,2)
COMMON/CLEI2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)
COMMON/BMFL2/SPAN(30,20),XBM(30,20,4),LRM(30,20,4),JMY(30)
COMMON/RMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2),COL1
1,ICFL(140),IFRM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)
5,SIDE(10),MARG,IPL0T(25,15),JPL,IGRID1(140),IGRID(140,2)
REAL MARG(10)
INTEGER COL1(140,2)

```

C
C
C

IGR3=IGRY

MKX=MK

IM2=INVER(IM)

MB=MB1(IM)

IFR=IFR2(IM)

IFR1=IFR2(IM2)

KA=K

KB=K1

JX1=JY(MK)

IF(IM.EQ.1)GO TO 80

KA=K1

KB=K

JX1=JX(MK)

80 CONTINUE

IF(JX1.EQ.1)GO TO 150

110 CONTINUE

DO 130 IF=1,IFR1

KAA=COL1(IF,IM2)

IFC(IF,LT,IFR1)KBB=COL1(IF+1,IM2)-1

IFC(IF,EQ,IFR1)KBB=MB1(IM2)

DO 120 K1=KAA,KBB

MK1=MKC(K1,IM2)

IF(K1.EQ,KAA)GO TO 120

IF(MK1.NE,MKX)GO TO 120

K2=K1-1

MK2=MKC(K2,IM2)

IF1=IF

GO TO 140

120 CONTINUE

130 CONTINUE

140 CONTINUE

DO 145 IF=1,IFR

KAA=COL1(IF,IM)

```
IF(IF.LT.IFR)KPB=COL1(IF+1,IW)-1
IF(IF.EQ.IFR)KBB=PB
DO 145 K3=KAA,KBB
MK3=MKC(K3,IW)
IF(MK2.NE.MK3)GO TO 145
KX1=KAA
KX=K3
IF(IW.EQ.2)KX1=K3
IF(IW.EQ.2)KX=KBB
GO TO 146
145 CONTINUE
146 CONTINUE
DO 148 K1=KX1,KX
K2=K1
IF(IW.EQ.2)K2=KX-K1+KX1
LVN=(K2-1)*LVM4+LVV
IF(CX(LVN,IW).EQ.999..AND.CZ(LVN,IW).EQ.999.)GO TO 148
MK2=MKC(K2,IW)
LX1=LX(MK2)
JX1=JX(MK2)
IF(IW.EQ.1)LX1=LY(MK2)
IF(IW.EQ.1)JX1=JY(MK2)
IF(JX1.NE.8.AND.SPAN(LX1,JX1).LE.8.)GO TO 148
IF(K2.EQ.KX.AND.IW.EQ.1)GO TO 149
IF(K2.EQ.KX1.AND.IW.EQ.2)GO TO 149
IF(IW.EQ.1)IGRX=IGRID1(MK2)/100
IGR1=MOD(IGRID1(MK2),100)
IF(IGR1.EQ.0)GO TO 148
IF(IW.EQ.2)IGRY=AMAX0(NCOL(1),NCOL(2))-IFR2(1)+IGR1
GO TO 300
148 CONTINUE
149 CONTINUE
MKX=MK2
IF(IF1.EQ.2)GO TO 300
GO TO 110
150 CONTINUE
DO 200 K=KA,KB
K1=K
IF(IW.EQ.2)K1=KB-K+KA
LVN=(K1-1)*LVM4+LVV
IF(CX(LVN,IW).EQ.999..AND.CZ(LVN,IW).EQ.999.)GO TO 200
IF(K1.EQ.KA.AND.IW.EQ.1)GO TO 200
IF(K1.EQ.KB.AND.IW.EQ.2)GO TO 200
MKX=MKC(K1,IW)
JX1=JX(MKX)
IF(IW.EQ.1)JX1=JY(MKX)
IF(JX1.EQ.1)GO TO 200
GO TO 110
200 CONTINUE
300 CONTINUE
RETURN
END
```

SUBROUTINE GRID5(LVM4)

COMMON MAT,MEH,LEV,ISOLV,LVFIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFR

COMMON ZZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)

COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU

COMMON KAK,LIST(10),IO9,CODE(2),MD(20),UF(20)

COMMON ITC1,NE7(90)

COMMON USF(10),BON(20),LON(20)

COMMON TAKI(10),JET(10),KET(20),LET(20)

INTEGER KAK(90,2),NME(11,3)

COMMON/PLOT1/PSTIF(990,2),BLEN(990,2),MKB(990,2),KST(990,2),

1IBR(990,2),CSTIF(992,2),CLEN1(992,2),CY(992,2),CZ(992,2),

2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),

3JBR(990,2)

COMMON/SMFL4/STPX,STPX1,STPY,STPY1,MR1(2),IFP2(2),CORD(140,2)

1,COL1,IFCL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)

4,CLEV(90),IPUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)

5,SIDE(10),MARG,IPLOT(25,15),JPL,IGRID1(140),GRID(140,2)

INTEGER COL1(140,2)

REAL MARG(10)

CALL ZERO(GRID1,122)

CALL ZERO(GRID,290)

DO 100 IW=1,2

MAX=MB1(IW)

IFR1=IFR2(IW)

IW2=INVER(IW)

MAX1=MB1(IW2)

IFR3=IFR2(IW2)

DO 100 IS=1,IFR1

KA=COL1(IS,IW)

IF(IS.LT.IFR1)KB=COL1(IS+1,IW)-1

IF(IS.EQ.IFR1)KB=MAX

KFLAG=0

DO 100 K=KA,KB

LVN=(K-1)+LVM4+IDOS(3)+1

MK=MKC(K,IW)

IFLA1=0

DO 60 IS1=1,IFR3

KA1=COL1(IS1,IW2)

IF(IS1.LT.IFR3)KB1=COL1(IS1+1,IW2)-1

IF(IS1.EQ.IFR3)KB1=MAX1

DO 60 K1=KA1,KB1

MK1=MKC(K1,IW2)

IF(MK.EQ.MK1)60 TO 70

60 CONTINUE

IFLA1=1

70 CONTINUE

IF(IFLA1.EQ.1)GRID(MK,IW)=77777.777

IF(IFLA1.EQ.1)GRID(MK,IW2)=77777.777

IF(GRID(MK,IW).EQ.77777.777)60 TO 90

IF(GRID(MK,IW2).EQ.77777.777)60 TO 90

IF(KFLAG.EQ.1)60 TO 80

GRID(MK,IW)=CORD(MK,IW)

GRID(MK,IW2)=CORD(MK,IW2)

```
KFLAG=1
GO TO 80
80 CONTINUE
GRID(MK,IW2)=77777.777
GRID(MK,IW)=CORD(MK,IW)
90 CONTINUE
KFLAG=1
IF(IPUG.EQ.53)PRINT 997,IW,IS,MK,K,KFLAG,CSTIF(LYN,IW),
*GRID(MK,IW2),GRID(MK,IW),CORD(MK,IW)
997 FORMAT(' GRID5 ',5I5,E12.5,3F20.5)
NCOL(IW)=NCOL(IW)+1
100 CONTINUE
DO 240 IW=1,2
NCOL1=NCOL(IW)
IW2=INVER(IW)
NCOL2=NCOL(IW2)
NCOL1=AMAX0(NCOL1,NCOL2)
DO 210 I=1,NCOL1
GNUM=GRID(I,IW)
IF(GNUM.EQ.-9999..OR.GNUM.EQ.77777.777)GO TO 210
DO 200 I1=1,NCOL1
IF(ABS(GNUM-GRID(I1,IW)).GE..5)GO TO 200
IF(I1.EQ.I)GO TO 200
GRID(I1,IW)=77777.777
200 CONTINUE
210 CONTINUE
DO 230 I=1,NCOL1
GNUM=GRID(I,IW)
DO 220 I1=1,NCOL1
IF(IPUG.EQ.53)PRINT 998,IW,I,I1,NCOL1,
*GNUM,GRID(I,IW),GRID(I1,IW)
998 FORMAT(' GRID5-2 ',4I4,3F20.5)
IF((GRID(I1,IW)-GNUM).LT.0..AND.IV.EQ.1)GNUM=GRID(I1,IW)
IF((GRID(I1,IW)-GNUM).GT.0..AND.IV.EQ.2)GNUM=GRID(I1,IW)
IF(ABS(GNUM-GRID(I1,IW)).LE..5)GO TO 215
GO TO 220
215 CONTINUE
COLI=GRID(I,IW)
GRID(I,IW)=GNUM
GRID(I1,IW)=COLI
220 CONTINUE
230 CONTINUE
240 CONTINUE
DO 250 IW=1,2
NCOL1=NCOL(IW)
IW2=INVER(IW)
NCOL2=NCOL(IW2)
NCOL1=AMAX0(NCOL1,NCOL2)
DO 250 IN=1,NCOL1
GRID1(IN,IW)=GRID(IN,IW)
250 CONTINUE
CALL ZERO(IGRID1,140)
CALL ZERO(GRID,280)
DO 249 IW=1,2
MAX=MB1(IW)
IFR1=IFR2(IW)
IW2=INVER(IW)
```

```

      NCOL1=NCOL(IW2)
      DO 249 IF=1,IFR1
      KA=COL1(IF,IW)
      IF(IF.LT.IFR1)KB=COL1(IF+1,IW)-1
      IF(IF.EQ.IFR1)KB=MAX
      DO 246 KK=KA,KB
      KK1=KB-KK+KA
      IF(IW.EQ.1)KK1=KK
      LVN=(KK1-1)*LVM4+INDOS(3)+1
      MK1=MKC(KK,IW)
      IF(IW.EQ.2)MK1=MKC(KK1,IW)
      IF(CSTIF(LW,IW).LE.1.E-10)GO TO 246
      GRID(IF,IW2)=CORD(MK1,IW2)
      GO TO 247
246 CONTINUE
247 CONTINUE
249 CONTINUE
      DO 252 IW=1,2
      MAX=MB1(IW)
      IFR1=IFR2(IW)
      IW2=INVER(IW)
      NCOL1=NCOL(IW2)
      NCOL2=NCOL(IW)
      NCOL1=AMAX0(NCOL1,NCOL2)
      DO 252 IF=1,IFR1
      KA=COL1(IF,IW)
      IF(IF.LT.IFR1)KB=COL1(IF+1,IW)-1
      IF(IF.EQ.IFR1)KB=MAX
      DO 252 K=KA,KB
      MK=MKC(K,IW)
      MK1=MKC(KA,IW)
      MK2=MKC(KB,IW)
      MKN=MK1
      IF(IW.EQ.2)MKN=MK2
      DO 254 I=1,NCOL1
      IF(IGRID1(I,IW2).EQ.77777.777.AND.I.EQ.NCOL1)GO TO 251
      IF(IGRID1(I,IW2).EQ.77777.777)GO TO 254
      DO 248 I1=1,IFR1
      IF(IBUG.EQ.53)
      *WRITE(6,*(I0GRID5-1*,816,3F10.5))
      *IW2,I,I1,I3,MK,MKN,IGRID1(MKN),IGRID1(MK),GRID1(I,IW2),
      *GRID(I1,IW2),CORD(MK,IW2)
      I3=IFR1-I1+1
      IF(ABS(GRID1(I,IW2)-GRID(I1,IW2)).GT..1)GO TO 248
      IF(ABS(GRID1(I,IW2)-CORD(MK,IW2)).GT..1)GO TO 254
      IF(IW2.EQ.1)IGRID1(MK)=I3*100+IGRID1(MK)
      IF(IW2.EQ.2)IGRID1(MK)=IGRID1(MK)+100+I3
      GO TO 251
248 CONTINUE
254 CONTINUE
      IF(IW2.EQ.1)IGRID1(MK)=IGRID1(MKN)
      IF(IW2.EQ.2)IGRID1(MK)=IGRID1(MK)+100+IGRID1(MKN)
251 CONTINUE
      IF(IBUG.EQ.53)
      1PRINT 996,IBUG,IW,NCOL1,MK,MKN,
      *IGRID1(MK),IGRID1(MKN)
      996 FORMAT(10 *I0GRID5-22*,7I8)

```



```

      IF (IPUG.EQ.53)
        *WRITE(6,*)('IGRID5-2',61E)*)IN,NCOL1,MK,MKN,
        *IGRID1(MK),IGRID1(MKN)
252 CONTINUE
      IF (IBUG.NE.53) GO TO 270
      DO 260 IW=1,2
        NCOL1=NCOL(IW)
        IW2=INVER(IW)
        NCOL2=NCOL(IW2)
        NCOL1=AMAX0(NCOL1,NCOL2)
      DO 260 IN=1,NCOL1
        PRINT 999,IW,IN,GRID1(IN,IW)
999 FORMAT(' GRID5-1',2I5,F14.3)
260 CONTINUE
270 CONTINUE
      RETURN
      END

```

LE MAP--(LO=A)

-ADDRESS--BLOCK----		PROPERTIES-----	TYPE-----	SIZE	-NAME---ADDRESS--BLOCK----	
36508	/PLOT1/		REAL	1960	IFLA1	11338
6428	//		REAL	20	IFLEV	118 //
08	/PLOT1/		REAL	1960	IFR1	11168
270108	/PLOT1/		REAL	1984	IFR2	68 /BMFL4/
23678	/BMFL4/		REAL	90	IFR3	11218
4238	//		REAL	2	IGRID1	42048 /BMFL4/
11508			REAL		ILINK	1148 //
4408	/BMFL4/		INTEGER	280	IN	11518
108	/BMFL4/		REAL	280	INWAY	425108 /PLOT1/
231108	/PLOT1/		REAL	1984	IO9	4228 //
327108	/PLOT1/		REAL	1984	IPLOT	34148 /BMFL4/
366108	/PLOT1/		REAL	1984	IPRINT	1138 //
525768	/PLOT1/		REAL	2	IRUN	1218 //
526008	/PLOT1/		REAL	2	IS	11228
558	//		REAL	3	ISOLV	38 //
464128	/PLOT1/		REAL	1984	IS1	11348
1158	//		REAL		ITC1	4758 //
1168	//		REAL		ITEST	1208 //
11448			REAL		IW	11148
44208	/BMFL4/		REAL	280	IW2	11178
13058	/BMFL4/		REAL	280	I1	11458
11438			INTEGER		I3	11638
172408	/PLOT1/		INTEGER	1960	JBR	526028 /PLOT1/
25218	/BMFL4/		INTEGER		JET	7248 //
10708	/BMFL4/		INTEGER	148	JPL	42038 /BMFL4/
1038	//		INTEGER	6	JSTOP	1128 //
758	//		INTEGER	6	K	11278
748	//		INTEGER		KA	11248
608	//		INTEGER	6	KAK	1248 //
668	//		INTEGER	6	KA1	11358
11528			INTEGER		KB	11258
13048	/BMFL4/		INTEGER		KB1	11368
128	//		INTEGER		KET	7368 //
27408	/BMFL4/		INTEGER	148	KFLAG	11268
25248	/BMFL4/		INTEGER	148	KK	11538

```

SUPROUTINE IDIMS (IS,K,KP,DIMEN,C12,C1X,ANGLF,HGT,SCALE)
COMMON PAT,MEM,LEV,ISOLV,LVIN,LNFIN,LVPAX,LNMAX,NI,IFLEV,IFFRM
COMMON ZZZZ7,NME,DAT(3),IDOC(6),IDCS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IOP,CODE(2),MD(20),UF(20)
COMMON ITC1,NF7(90)
COMMON USE(10),BON(20),LCN(20)
COMMON TAKI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)

```

C

```

COMMON/PLOT1/PSTIF(980,2),BLEN(980,2),MKB(980,2),KST(980,2),
11BR(980,2),CSTIF(992,2),CLEVI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHYC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JRR(980,2)

```

```

COMMON/REIZI/KUZ,UZ1,UZ2,UZ3,UZ4,UZ5,UZ6,UZ7,UZ8,UZ9,UZ10,UZ11,
UZ12,UZ13,UZ14,UZ15,UZ16,UZ17,UZ18
COMMON/EMFL4/STPX,STPY,STPY1,MP1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZ1,STZ2,IFLAG(140),IFLAG(140)
5,KKPLCT(140),SIDE(10),MARG,IPLT(25,15),JPL,GRID1(140)

```

```

6,IGRID(140,2)
REAL MARG(10)
INTEGER COL1(140,2)
DIMENSION IDIM(15),ICC(7)
DATA ICC/1,10,100,1000,10000,100000,1000000/
DATA IDIM/4H1/16,3H1/8,4H3/16,3H1/4,4H5/16,3H3/8,

```

```

*4H7/16,3H1/2,4H9/16,3H5/8,5H11/16,3H3/4,5H13/16,
*3H7/8,5H15/16/

```

```

IBUG=LET(11)

```

```

ICHR=0

```

```

ICHR1=0

```

```

ICHR2=0

```

```

ICHR3=0

```

```

IWD1=4H++++

```

```

IWD2=4Hxxxx

```

```

IWD3=4Hssss

```

```

IF(KUZ.EQ.2)GO TO 160

```

```

IBIAN=1H

```

```

ISIGN=1H-

```

```

ICOLF=DIMEN+.01

```

```

ICOLI=((DIMEN+10000-ICOLF+10000)*12/10000)+.1

```

```

ICOLD=((DIMEN+10000-ICOLF+10000)*12-ICOLI*10000)*16/10000+.1

```

```

IF(IBUG.EQ.53)PRINT 999,KUZ,ICOLF,ICOLI,ICOLD,DIMEN

```

```

999 FORMAT(' IDIMS',4I6,F15.6)

```

```

5 CONTINUE

```

```

IF(ICOLF.LE.9.AND.ICOLF.NE.0)ICHR1=1

```

```

IF(ICOLF.GT.9.AND.ICOLF.LE.99)ICHR1=2

```

```

IF(ICOLF.GT.99.AND.ICOLF.LE.999)ICHR1=3

```

```

IF(ICOLF.GT.999.AND.ICOLF.LE.9999)ICHR1=4

```

C

```

IF(ICOLI.EQ.0.AND.ICOLD.NE.0)ICHR2=1

```

```

IF(ICOLI.GT.9)ICHR2=2

```

```

IF(ICOLI.GT.9.AND.ICOLI.LE.9)ICHR2=1

```

```

IF(ICHR1.EQ.8)GO TO 70

```

```

GO TO(10,20,30,40)ICHR1

```

```

CALL GETDEF

```

```

10 CONTINUE

```

```
      ENCODE(7,81,IMD1)ICOLF,ISIGV
      ICHR1=2
      GO TO 50
20  CONTINUE
      ENCODE(7,82,IMD1)ICOLF,ISIGV
      ICHR1=3
      GO TO 50
30  CONTINUE
      ENCODE(4,83,IMD1)ICOLF,ISIGV
      ICHR1=4
      GO TO 50
40  CONTINUE
      ENCODE(5,84,IMD1)ICOLF,ISIGV
      ICHR1=5
50  CONTINUE
      IF(ICHR1.EQ.0)ICMR2=1
      IF(ICHR2.EQ.1)GO TO 60
      ENCODE(3,82,IMD2)ICOLI,IPLAN
      ICHR2=3
      GO TO 100
60  CONTINUE
      ENCODE(2,81,IMD2)ICOLI,IPLAN
      ICHR2=2
      GO TO 100
70  CONTINUE
      IF(ICHR2.EQ.0)GO TO 150
      IF(ICHR2.EQ.1)GO TO 80
      ENCODE(3,82,IMD2)ICOLI,IPLAN
      ICHR2=3
      GO TO 100
80  CONTINUE
      ENCODE(2,81,IMD2)ICOLI,IPLAN
      ICHR2=2
100 CONTINUE
      IF(ICOLD.EQ.0)GO TO 150
      GO TO(110,120,110,120,110,120,110,120,130,120,130
      *,120,130,140)ICOLD
      CALL GOTOER
110 CONTINUE
      ICHR3=4
      ENCODE(4,85,IMD3)IDIM(ICOLD)
      GO TO 150
120 CONTINUE
      ICHR3=3
      ENCODE(3,86,IMD3)IDIM(ICOLD)
      GO TO 150
130 CONTINUE
      ICHR3=5
      ENCODE(5,87,IMD3)IDIM(ICOLD)
      GO TO 150
140 ICOLI=ICOLI+1
      ICOLD=0
      ICOLI1=ICOLI/12
      ICOLI=MOD(ICOLI,12)
      ICOLF=ICOLF+ICOLI1
      GO TO 5
150 CONTINUE
```

```

      ICHR=ICHR1+ICHR2+ICHR3
      GO TO 190
      R1 FORMAT(I1,A1)
      R2 FORMAT(I2,A1)
      R3 FORMAT(I3,A1)
      R4 FORMAT(I4,A1)
      R5 FORMAT(A4)
      R6 FORMAT(A3)
      R7 FORMAT(A5)
      160 CONTINUE
C
C   SI UNITS
C
      ICOLF=(IDIMEN+.002)*1000
      IRES=MOD(ICOLF,100)
      IF(IRES.LE.4) ICOLF=ICOLF-IRES
      DO 170 IC=1,7
      IF(ICOLF.LT.ICC(IC)) GO TO 175
      IF(ICOLF.EG.ICC(IC)) GO TO 170
      170 CONTINUE
      175 CONTINUE
      ICHR=IC-1
      GO TO 185
      180 CONTINUE
      ICHR=IC
      185 CONTINUE
C
      WRITE(6,*(10IDIM-1*,I10,3F15.5*)) ICOLF,DIMEN,HGT,FACTOR
      IWD1=ICOLF
      190 CONTINUE
      IF(IS.EQ.2.OR.IS.EQ.4) GO TO 210
      IF(IS.EQ.3.OR.IS.EQ.5) GO TO 250
      IF(IPUE.EQ.53)
      *PRINT 998,ICHR,ICHR1,ICHR2,ICHR3,DIMEN,IWD1,IWD2,IWD3
      998 FORMAT(10IDIM5*,4I4,F8.2,3(1X,A10))
      ANGL=ANGLE
      ILEV=0
      LEVEL=KB
      LEV1=K
      IF(LEV1.EQ.LEVEL) GO TO 192
      CLEV1=(CLEV(LEV1+1)-CLEV(LEV1))/SCALE
      IF(ICHR+HGT.GT.CLEV1) ILEV=1
      192 CONTINUE
      IF(KUZ.EQ.2) GO TO 200
      IF(ILEV.EQ.0) GO TO 193
      C1X=C1X-.093-ICHR+HGT
      C2Z=C1Z+.047
      ANGL=0.
      GO TO 194
      193 C2Z=C1Z-ICHR+HGT+.5
      194 CONTINUE
      CALL SYMBOL(C1X,C2Z,HGT,IWD1,ANGL,IC+R1)
      IF(ICHR.EQ.ICHR1) GO TO 600
      IF(ILEV.EQ.0) GO TO 195
      C1X=C1X+ICHR1+HGT
      GO TO 196
      195 C2Z=C2Z+ICHR1+HGT
      196 CONTINUE

```

```
CALL SYMBOL(C1X,C2Z,HGT,IWD2,ANGL,ICHR2)
IF(ICHR.EQ.(ICHR1+ICHR2))GO TO 600
IF(ILEV.EQ.0)GO TO 197
C1X=C1X+ICHR2*HGT
GO TO 198
197 C2Z=C2Z+ICHR2*HGT
198 CONTINUE
CALL SYMBOL(C1X,C2Z,HGT,IWD3,ANGL,ICHR3)
GO TO 600
200 CONTINUE
IF(ILEV.EQ.0)GO TO 203
C1X=C1X-.093-ICHR*HGT
C2Z=C1Z-.047
ANGL=0.
GO TO 204
203 C2Z=C1Z-ICHR*HGT*.5
204 CONTINUE
CALL NUMBER(C1X,C2Z,HGT,FLOAT(IWD1),ANGL,-1)
GO TO 600
210 CONTINUE
C2=C1Z+.047
FACTOR=1.
ELENG=DIMEN*UZ11/SCALE
HGT=HGT*FACTOR
WORD=ICHR*HGT
IF(ELENG.GT.WORD.AND.IS.EQ.4)GO TO 214
IF(ELENG.GT.WORD)GO TO 212
FACTOR=.75
HGT=FACTOR*HGT
WORD=ICHR*HGT
IF(ELENG.GT.WORD.AND.IS.EQ.4)GO TO 214
IF(ELENG.GT.WORD)GO TO 212
IF(IWOR.EQ.1)GO TO 211
C2=C2+.125
IWOR=1
GO TO 215
211 CONTINUE
C2=C2-.094
212 CONTINUE
IF(K.EQ.KB)C2=C1Z+.422
214 CONTINUE
IWOR=0
215 CONTINUE
FAC1=1.
IF(ICHR3.EQ.0)FAC1=.75
IF(IBUG.EQ.53)PRINT 997,ICMR,ICHR3,DIMEN,HGT
997 FORMAT(' IDIM5-1',2I5,2F20.5)
C1=C1X+DIMEN*.5*UZ11/SCALE-ICHR*.5*HGT*FAC1
IF(KUZ.EQ.2)GO TO 220
CALL SYMBOL(C1,C2,HGT,IWD1,ANGLE,ICHR1)
IF(ICHR.EQ.ICHR1)GO TO 600
C1=C1+ICHR1*HGT
CALL SYMBOL(C1,C2,HGT,IWD2,ANGLE,ICHR2)
IF(ICHR.EQ.(ICHR1+ICHR2))GO TO 600
C1=C1+ICHR2*HGT
CALL SYMBOL(C1,C2,HGT,IWD3,ANGLE,ICHR3)
GO TO 600
```

```

220 CONTINUE
C  WRITE(6,*(='IWDIM-3',I7,I4,3F15.5)*)
C  *IWD1,ICRHR,C1,C2,HGT,ANGLE
  CALL NUMBER(C1,C2,HGT,FLOAT(IWD1),ANGLE,-1)
  GO TO 600
250 CONTINUE
  C1=C17-.047
  FACTOR=1.
  ELENG=DIMEN*U211/SCALE
  HGT=HGT*FACTOR
  WORD=ICHR*HGT
  IF(ELENG.GT.WORD.AND.IS.EQ.4)GO TO 254
  IF(ELENG.GT.WORD)GO TO 252
  FACTOR=.75
  HGT=FACTOR*HGT
  WORD=ICHR*HGT
  IF(ELENG.GT.WORD.AND.IS.EQ.4)GO TO 254
  IF(ELENG.GT.WORD)GO TO 252
  IF(IWOR.EQ.1)GO TO 251
  C2=C2+.125
  IWOR=1
  GO TO 255
251 CONTINUE
  C2=C2-.094
252 CONTINUE
  IF(K.EQ.KB)C1=C17-.422
254 CONTINUE
  IWOR=0
255 CONTINUE
  FAC1=1.
  IF(ICHR3.EQ.0)FAC1=.75
  C2=C1X+DIMEN*.5*U211/SCALE-ICHR*.5*HGT*FAC1
  IF(KU2.EQ.2)GO TO 280
  CALL SYMBOL(C1,C2,HGT,IWD1,ANGLE,ICHR1)
  IF(ICHR.EQ.ICHR1)GO TO 300
  C2=C2+ICHR1*HGT
  CALL SYMBOL(C1,C2,HGT,IWD2,ANGLE,ICHR2)
  IF(ICHR.EQ.(ICHR1+ICHR2))GO TO 300
  C2=C2+ICHR2*HGT
  CALL SYMBOL(C1,C2,HGT,IWD3,ANGLE,ICHR3)
  GO TO 300
280 CONTINUE
  CALL NUMBER(C1,C2,HGT,FLOAT(IWD1),ANGLE,-1)
300 CONTINUE
600 CONTINUE
  RETURN
  END

```

LE MAP--(ILO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
12338	7	DUMMY-ARG	REAL		CLEN1	270108	/PLOT1/
36508		/PLOT1/	REAL	1960	CLEV	23678	/BMFL4/
6428		//	REAL	20	CLEV1	12378	
08		/PLOT1/	REAL	1960	CODE	4238	//
					COL1	4408	/BMFL4/

 FUNCTION FXBM5(L,J,K)

INTEGER XBM

 COMMON/BMFL2/SPAN(30,20),XBM(30,20,4),LBM(30,20,4),JMX(30)

C

C THIS FUNCTION COMPUTES THE DISTANCE BETWEEN

C SECONDARY BEAMS

C K IS THE NO. OF SECONDARY BEAMS PER MAIN BEAMS

C K MAX. 8

C

IF(K.GT.4)GO TO 2

FXBM5=MOD(XBM(L,J,K),100000)

FXBM5=FXBM5*.1

RETURN

2 CONTINUE

FXBM5=XPM(L,J,K-4)/100000

FXBM5=FXBM5*.1

RETURN

END

 LE MAP--(LO=A)

-ADDRESS--BLOCK-----PROPERTIES-----TYPE-----SIZE -NAME---ADDRESS--BLOCK---

NONE

2 DUMMY-ARG

REAL

INTEGER

L

I

DUMMY-AR

12430B /BMFL2/

INTEGER

30

LBM

5670R

/BMFL2/

3 DUMMY-ARG

INTEGER

SPAN

0R

/BMFL2/

 URES--(LO=A)

----TYPE-----ARGS-----CLASS-----

GENERIC

2

INTRINSIC

 ENT LABELS--(LO=A)

ADDRESS-----PROPERTIES-----DEF

22B

14

 POINTS--(LO=A)

-ADDRESS--ARGS---

4B

3

 TICS--

 M-UNIT LENGTH

368 = 30

ELLED COMMON LENGTH

124668 = 5430

RAGE USED

200000R = 65536

E TIME

0.148 SECONDS

FUNCTION JLBMS(L,J,K)

COMMON/BMFL2/SPAN(30,20),XBM(30,20,4),LBM(30,20,4),JMX(30)

C

C THIS FUNCTION LOOKS FOR THE MARK OF THE SECONDARY BEAM

C

IF(K.GT.4)GO TO 2

JLBMS=MOD(LBM(L,J,K),10000)

RETURN

2 CONTINUE

JLBMS=LBM(L,J,K-4)/10000

RETURN

END

LE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
2	DUMMY-ARG		INTEGER		L	1	DUMMY-AR
NONE			INTEGER		LBM	56708	/BMFL2/
124308	/BMFL2/		INTEGER	30	SPAN	08	/BMFL2/
3	DUMMY-ARG		INTEGER		XBM	11308	/BMFL2/

URES--(LO=A)

---TYPE-----ARGS-----CLASS-----

GENERIC

2

INTRINSIC

ENT LABELS--(LO=A)

ADDRESS-----PROPERTIES-----DEF

21B

9

POINTS--(LO=A)

---ADDRESS--ARGS---

4B

3

TICS--

M-UNIT LENGTH

338 = 27

ELLED COMMON LENGTH

124668 = 5438

RAGE USED

2808008 = 65536

E TIME

0.132 SECONDS


```

SUBROUTINE LABEL5(IM,J,C1,C2,KA,KP,LVMA,IV,LN2,SCALE)
COMMON/PI OT1/PSTIF(980,2),RLEN(980,2),MKB(980,2),KST(980,2),
1IBR(980,2),CSTIF(992,2),CLENI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JBR(980,2)
COMMON/ALFA/SINA,COSA,ID3
COMMON/THETFL/DUM(1890)
DIMENSION IFBB(250),XB1(250),YB1(250),MKB1(250),PMSH(21)
1,BIMSH(30,20),OB2(250),BIM(240)
EQUIVALENCE (DUM(1),IFBB(1)),(DUM(251),XB1(1)),(DUM(501),YB1(1))
EQUIVALENCE (DUM(751),MKB1(1))
EQUIVALENCE (DUM(1001),BIMSH(1)),(DUM(1022),BIMSH(1,1))
EQUIVALENCE (DUM(1622),OB2(1)),(DUM(1872),INUMX)
EQUIVALENCE (DUM(1),BIM(1))
COMMON/BMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140)
5,KKPL0T(140),SIDE(10),MARG,IPL0T(25,15),JPL,IGRID1(140)
6,IGRID(140,2),DUMMY(240)
REAL MARG(10)
INTEGER COL1(140,2)
DIMENSION LETER(26)
DATA LETER/1MA,1MB,1MC,1MD,1ME,1MF,1MG,1MH,1MI,1MJ,1MK,
*1ML,1MN,1MO,1MP,1MQ,1MR,1MS,1MT,1MU,1MV,1MW,1MX,1MY,1MZ/

C
C COMPUTE HEIGHT AND CLEARANCE FOR MARKS AND TITLES
C
C
C IM=1 BM MARKS (ELEV.)
C 2 COL MARKS (ELEV.)
C 3 COL MARKS (PLAN)
C 4 BM MARKS (PLAN)
C 5 SEC. BM MARKS (PLAN)
C 6 PLAN VIEW TITLES
C
FACTOR=.75
HGT=.125*FACTOR
HGT2=HGT/FACTOR
COSA1=COSA
SINA1=SINA
HGT1=HGT
IF(IM.EQ.5)HGT1=.75*HGT
IF(IM.EQ.5)SINA1=0.
IF(IM.EQ.5)COSA1=1.
IF(IM.EQ.4.AND.IFBM.EQ.2)HGT1=.75*HGT1
IF(IM.EQ.5.AND.IFBM.EQ.2)HGT1=.75*HGT1

C
C WRITE BEAM MARKS
C
IF(IM.EQ.2)60 TO 150
IF(IM.EQ.3)60 TO 450
IF(IM.EQ.6)60 TO 470
IF(IPL0T(14,JPL).EQ.0)60 TO 110
IF(IV.EQ.1)NTEM=MOD(MKB(LN2,IV),10000)
IF(IV.EQ.2)NTEM=MKB(LN2,IV)/10000
I11=NTEM/100
BLENG=3.*HGT1

```

```

      IF(LI1.GT.9)BLENG=4.*HGT1
      C3=C1-1.5*HGT
      C2B=C2-5.*HGT/3.
      IF(IM.EQ.1)GO TO 60
      IF(IV.EQ.2)GO TO 20
      C2B=C2-BLENG*.5*SINA1-(.047*HGT1)*COSA1
      C3=C1+(.047*HGT1)*SINA1-BLENG*.5*COSA1
      GO TO 60
20  CONTINUE
      C2B=C2+(.047*HGT1)*SINA1-BLENG*.5*COSA1
      C3=C1+(.047*HGT1)*COSA1+BLENG*.5*SINA1
60  CONTINUE
      ALFA=0.
      ALFA=ASIN(SINA1)*57.29579
      ANGLE=0.
      IF(IM.EQ.5)NTEM=LN2
      IF(IM.EQ.1)GO TO 90
      ANGLE=ALFA
      IF(IV.EQ.2)ANGLE=90+ALFA
90  CONTINUE
      IF(IBUG.EQ.5)
1PRINT 999,IM,IV,NTEM,IFBM,C3,C2B,ALFA,ANGLE,STMX,HGT
999 FORMAT(* ** DEBUG BEAM MARKS ***,4I6,2F6.2,4F8.4)
      IF(IPL0T(14,JPL).EQ.0)GO TO 110
      IF(IM.EQ.4.AND.IFBM.NE.3)GO TO 100
      IF(IM.EQ.5.AND.IFBM.NE.3)GO TO 100
      IF(IM.EQ.1)GO TO 100
      C3=C1-.5*HGT1
      C2B=C2-.047-.5*HGT1
      IF(IV.EQ.1)GO TO 95
      C3=C1+.047+.5*HGT1
      C2B=C2-.5*HGT1
95  CONTINUE
      CALL SYMBOL(C3,C2B,HGT1,***,ANGLE,1)
      GO TO 110
100 CALL NUMBER(C3,C2B,HGT1,Float(NTEM),ANGLE,-1)
110 CONTINUE
      IF(IBUG.EQ.53)PRINT 998,LN2,IV,C3,C2B
998 FORMAT(* SECMARK*,2I6,2F8.2)
      IF(IM.EQ.4)GO TO 460
      RETURN
C
C  COLUMN MARKS *****
C
150 CONTINUE
      DO 200 I=KA,KB
      LN2=(I-1)*LVM4+1
      C1X=CX(LN2,IV)/SCALE+HGT
      C1Z=CZ(LN2,IV)/SCALE-HGT
      IF(IPL0T(13,JPL).EQ.0)GO TO 200
      CALL SYMBOL(C1X,C1Z,HGT,*C*.0..1)
      C1X=C1X+HGT
      NTEM=NKC(I,IV)
      CALL NUMBER(C1X,C1Z,HGT,Float(NTEM),0..-1)
200 CONTINUE
C
C  LEVEL NUMMERS *****

```

C

```

      LEVEL=LVM4
      IF(IPUG.EQ.53)
        1PRINT 990
      990 FORMAT(' ** DEBUG LAB ** FIRST ST LAST ST LVMX LEV LV COOR*')
      DO 220 LL=KA,KP
      KK=LL
      210 LVM=(KK-1)*LVM4+LEVEL
      IF(ICSTIF(LVN,IV).LE.1.E-20)GO TO 215
      GO TO 230
      215 CONTINUE
      IF(KK.EQ.KB)LEVEL=LEVEL+1
      IF(KK.EQ.KB)GO TO 220
      KK=KK+1
      GO TO 210
      220 CONTINUE
      230 CONTINUE
      DO 350 I=2,LEVEL
      IF(IPUG.EQ.53)
        1PRINT 991,KA,KB,LVM4,I,CLEV(I)
      991 FORMAT(' ** DEBUG LAB **',4I8,F8.2)
      C1X=0.
      C1Z=CLEV(I)/SCALE
      ELENG=HGT
      IF(I.GT.9)ELENG=2.*HGT
      C1=C1X/SCALE-.125
      C2=C1Z+.0938
      C3=C1-1.265
      IF(I.PLOT(12,JPL).EQ.0)GO TO 350
      CALL PLOT(C1,C1Z,3)
      CALL PLOT(C3,C1Z,2)
      NTEM=I-1
      C4=C1-.5625-ELENG*.5
      CALL NUMBER(C4,C2,HGT,FLOAT(NTEM),0.,-1)
      350 CONTINUE
      C1=C4-2.*HGT
      C2=C2+1.75*HGT
      IF(I.PLOT(12,JPL).EQ.0)GO TO 360
      CALL SYMBOL(C1,C2,HGT,'LEVELS',0.,6)
      360 CONTINUE
      LN1=(KA-1)*LVM4+1
      LN3=(KB-1)*LVM4+1
      C1=CX(LN1,IV)/SCALE
      C2=CX(LN3,IV)/SCALE
      C33=STZZ1/SCALE-10.*HGT2
      C44=(C1+C2)*.5-16*HGT2
      CSX=(C1+C2)*.5
      CSY=C33-2.*HGT2
      C
      CALL NEUPEN(11)
      CALL SYMBOL(C44,C33,1.6*HGT2,'FRAMING ELEVATION',0.,18)
      C44=C44+28.8*HGT2
      K=0
      IW2=INVER(IV)
      NCOL1=NCOL(IW2)
      MK=MKG(KA,IV)
      IGR2=IGRID1(MK)/100
      IF(IW2.EQ.2)IGR2=MJD(IGRID1(MK),100)

```

```

      DO 370 II=1,NCOL1
      IF(IPIUG,EQ,53)PRINT 993,MK,IW2,II,NCOL1,
      *CORD(MK,IW2),GRID1(II,IW2)
      993 FORMAT(' LAP5-T1=.4I4.2F20.5')
      IF(GRID1(II,IW2).EQ.77777.777)GO TO 370
      K=K+1
      IF(APS(CORD(MK,IW2)-GRID1(II,IW2)).LT..1)GO TO 390
      370 CONTINUE
      390 CONTINUE
      K=IGR2
      IF(IPIUG,EQ,53)PRINT 992,NCOL1,II,K,IW
      NTEM=LETER(K)
      992 FORMAT(' LAP5-T',5I5)
      CALL SYMPOI(C44,C33,1.6*HGT2,*LINE *.0.,5)
      RAD=.15
      C44=C44+.15*R.*HGT2
      C33=C33+.8*HGT2
      CALL ARC(C44,C33,RAD,0.,360.)
      IC=1
      IF(INWAY(IW),EQ,1)GO TO 390
      C44=C44-.5*HGT
      C33=C33-.5*HGT
      CALL SYMBOL(C44,C33,HGT,NTEM,0.,IC)
      CALL SYMBOL(CSX,CSY,HGT2,*SCALE 1 = *.0.,10)
      CSX= CSX+10.*HGT2
      NTEM= IPILOT(20,J)
      CALL NUMBER(CSX,CSY,HGT2,FLOAT(NTEM),0.,-1)
      GO TO 410
      390 CONTINUE
      NTEM=K
      IC=1
      IF(NTEM.GT.9.AND.NTEM.LE.99)IC=2
      C44=C44-.5*HGT*IC
      C33=C33-.5*HGT
      CALL NUMBER(C44,C33,HGT,FLOAT(NTEM),0.,-1)
      410 CONTINUE
      CALL NEWPEN(1)
      RETURN
      450 CONTINUE
      C
      C PLAN VIEW COLUMN MARKS
      C
      IF(IFLAG1(LN2).NE.0)GO TO 460
      IFLAG1(LN2)=1
      C1X=C1
      C1Z=C2
      CALL NUMBER(C1X,C1Z,HGT,FLOAT(LN2),0.,-1)
      460 CONTINUE
      RETURN
      C
      C PLAN VIEW TITLE *****
      C
      470 CONTINUE
      C2X=C1-17.8*HGT2
      C2Y=C2-18.8*HGT2
      C
      CALL NEWPEN(11)
      CALL SYMBOL(C2X,C2Y,1.6*HGT2,*FRAMING PLAN LEVEL *.0.,20)

```

```

NTEM=LN2
C1X=C2X+32.*HGT2
CALL NUMBER(C1X,C2Y,1.6*HGT2,FLOAT(NTEM),0.,-1)
C1X= C1-5.56*HGT2
C2Y= C2Y-2.*HGT2
NTEM= IPLOT(20,J)
CALL SYMBOL(C1X,C2Y,HGT2,*SCALE 1 = *,0.,10)
C1X=C1X+10.*HGT2
CALL NUMBER(C1X,C2Y,HGT2,FLOAT(NTEM),0.,-1)
CALL NEWPEN(1)
RETURN
END

```

LE MAP--(LO=A)

-ADDRESS-	-BLOCK-	PROPERTIES	TYPE	SIZE	-NAME-	ADDRESS	-BLOCK-
13428			REAL		HGT2	13318	
13438			REAL		I	13458	
50508	/BMFL4/	EQV	REAL	240	IBR	172408	/PLOT1/
17758	/THETFL/	EQV	REAL	600	IBUG	25218	/BMFL4/
36508	/PLOT1/		REAL	1960	IC	13768	
13378			REAL		ICFL	10708	/BMFL4/
17508	/THETFL/	EQV	REAL	21	ID3	28	/ALFA/
08	/PLOT1/		REAL	1960	IFBB	08	/THETFL/
270108	/PLOT1/		REAL	1984	IFRM	13048	/BMFL4/
23678	/BMFL4/		REAL	90	IFLAG	27408	/BMFL4/
4408	/BMFL4/		INTEGER	280	IFLAG1	25248	/BMFL4/
108	/BMFL4/		REAL	280	IFR2	68	/BMFL4/
18	/ALFA/		REAL		IGRID	44208	/BMFL4/
13328			REAL		IGRID1	42048	/BMFL4/
231108	/PLOT1/		REAL	1984	IGR2	13728	
13648			REAL		II	13738	
13658			REAL		IM	1	DUMMY-ARG
327108	/PLOT1/		REAL	1984	INUMX	35178	/THETFL/
366108	/PLOT1/		REAL	1984	INWAY	425108	/PLOT1/
3	DUMMY-ARG		REAL		IPLOT	34148	/BMFL4/
13478			REAL		IW	8	DUMMY-ARG
13508			REAL		IW2	13678	
525768	/PLOT1/		REAL	2	J	2	DUMMY-ARG
4	DUMMY-ARG		REAL		JBR	526028	/PLOT1/
13418			REAL		JPL	42038	/BMFL4/
13778			REAL		K	13668	
13808			REAL	1	KA	5	DUMMY-ARG
524008	/PLOT1/		REAL	2	KB	6	DUMMY-ARG
13408			REAL		KK	13538	
13628			REAL		KKPLOT	31548	/BMFL4/
13578			REAL		KST	133708	/PLOT1/
13638			REAL		LETTER	12758	
08	/THETFL/	EQV	REAL	1890	LEVEL	13518	
50508	/BMFL4/	EQV	REAL	240	LI1	13368	
464128	/PLOT1/		REAL	1984	LL	13528	
13568			REAL		LN1	13608	
13278			REAL		LN2	9	DUMMY-ARG
13058	/BMFL4/		REAL	280	LN3	13618	
13388			REAL		LVH4	7	DUMMY-ARG
13348			REAL		LVN	13548	

```

SUBROUTINE LINES(IS,KA,KP,LVN,I2G,IV,EXD,SCALE)
COMMON PAT,MEI,LEV,ISOLV,LVEIN,LFEIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(4),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFP,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IOP,CODE(2),MD(20),UF(20)
COMMON ITC1,NE7(90)
COMMON USF(10),BON(20),LON(20)
COMMON TAKI(10),JET(10),KET(20),MET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/PLT1/PSTIF(980,2),MLEN(980,2),MKB(980,2),KST(980,2),
1IBR(980,2),CSTIF(992,2),CLEMI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JBR(980,2)
COMMON/CLFL2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)
COMMON/CLFL3/X(140),Y(140),Z(140),XMAX(140),YMAX(140),KSC(140)
COMMON/CMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2),COL1
1,ICFL(140),IFRM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IRUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)
5,SIDE(10),MARG,IPLLOT(25,15),JPL,IGRID1(140),IGRID(140,2)
REAL MARG(10)
INTEGER COL1(140,2)
DIMENSION LETER(26)
DATA LETER/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,
1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/
C
IFPLOT=0
IFPLOT1=0
IW2=INVER(IW)
NCOL2=NCOL(IW2)
IF(IS.EQ.2)GO TO 200
C1=CX(LVN,IW)/SCALE
C2=CZ(LVN,IW)/SCALE+.125
C21=.5
C24=.0625
C23=.09375
EXD1=EXD-1.5625
MK=MKC(KA,IV)
IF(IW.EQ.1)IGR2=IGRID1(MK)/180
IF(IW.EQ.2)IGR2=MOD(IGRID1(MK),180)
10 CONTINUE
IF(IFPLOT1.EQ.3)EXD1=EXD-.71875
IF(EXD1.LE..5)GO TO 30
IF(EXD1.GT..5.AND.EXD1.LE..73)GO TO 30
IFPLOT=2
GO TO 40
30 CONTINUE
IFPLOT=1
40 CONTINUE
IF(IFPLOT1.EQ.3)GO TO 50
KN=2
50 CONTINUE
DO 60 I=1,KN
CALL PLOT(C1,C2,3)
C2=C2+C21
CALL PLOT(C1,C2,2)
C2=C2+C24
CALL PLOT(C1,C2,3)

```

```

3      C2=C2+C23
4      CALL PLOT(C1,C2,2)
5      C2=C2+C24
60 CONTINUE
7      IF(IFPLOT.EQ.2)GO TO 70
8      CALL PLOT(C1,C2,3)
9      C2=C2+EYD1
10     CALL PLOT(C1,C2,2)
11     GO TO 60
12     ^70 CONTINUE
13     EXD=EXD1
14     IFPLOT1=3
15     KN=2
16     GO TO 10
17     80 CONTINUE
18     C2=C2+.5*C21
19     RAD=.25
20     CALL ARC(C1,C2,RAD,0.,360.)
21     NCOL1=NCOL(IW)
22     NCOL1=AMAX0(NCOL1,NCOL2)
23     K=0
24     DO 90 I=1,NCOL1
25     IF(GRID1(I,IW).EQ.77777.777)GO TO 90
26     K=K+1
27     IF(IPUG.EQ.53)PRINT 998,I,IW,MK,K,NCOL1,
28     *IGR2,GRID1(IGR2,IW),GRID1(I,IW)
29     998 FORMAT(' LINE5-1.,6I4,2F20.5)
30     IF(ABS(CORD(MK,IW)-GRID1(I,IW)).LT..1)GO TO 92
31     90 CONTINUE
32     92 CONTINUE
33     IF(IPUG.EQ.53)PRINT 999,IW,I,K,NCOL1
34     IC=1
35     999 FORMAT(' LINE5.,5I5)
36     IF(IW.EQ.1)GO TO 94
37     K=IGR2
38     NTEM=K
39     IC=1
40     IF(NTEM.GT.9.AND.NTEM.LE.99)IC=2
41     94 CONTINUE
42     HGT=.125
43     C1=C1-.5*HGT+IC
44     C2=C2-.5*HGT
45     IF(IW.EQ.2)GO TO 97
46     NTEM=LETER(K)
47     CALL SYMBOL(C1,C2,HGT,NTEM,0.,IC)
48     GO TO 100
49     97 CONTINUE
50     CALL NUMBER(C1,C2,HGT,FLOAT(NTEM),0.,-1)
51     100 CONTINUE
52     GO TO 600
53     200 CONTINUE
54     C
55     C FLOOR PLAN GRID LINES
56     C
57     IW2=INVER(IW)
58     IGRX=KA
59     IGRY=KB

```

```
      MK=LVN
      IF (IPUG.EQ.53) WRITE (6,*(1+LINE5-1+*,4I4,7F10.3)*)
      *MK,IV,IGRX,IGRY,GRID1(IGPX,IV),GRID1(IGRY,IV2),
      *GRID1(IGRX,IV2),GRID1(IGRY,IV),EXD,FCL(MK,4),FCL(MK,5)
      IF (IV.EQ.2) GO TO 210
      IF (MAT.EQ.2) GO TO 207
      DIM1=X(MK)*.5/(12.*SCALE)
      GO TO 208
207 CONTINUE
      DIM1=1.*FCL(MK,4)/(24.*SCALE)
208 CONTINUE
      GRX=AMIN1(GRID1(IGRX,IV),CORD(MK,IV))/SCALE
      EXD=EXD-GRX+DIM1
      C1=GRX-.125-DIM1
      C2=GRID1(IGRY,IV2)/SCALE
      EXD1=ABS(EXD+1.5625)
      C21=-.5
      C24=-.0625
      C23=-.09375
      GO TO 220
210 CONTINUE
      IF (MAT.EQ.2) GO TO 215
      DIM1=Y(MK)*.5/(12.*SCALE)
      IF (KSC(MK).EQ.2) DIM1=X(MK)*.5/(12.*SCALE)
      GO TO 217
215 CONTINUE
      DIM1=1.5*FCL(MK,5)/(24.*SCALE)
217 CONTINUE
      GRY=AMAX1(GRID1(IGRY,IV),CORD(MK,IV))/SCALE
      EXD=EXD-GRY-DIM1
      C1=GRID1(IGRX,IV2)/SCALE
      C2=GRY+.125+DIM1
      EXD1=EXD-1.5625
      C21=.5
      C24=.0625
      C23=.09375
220 CONTINUE
      IF (IPUG.EQ.53) WRITE (6,*(1+LINE5-2+*,5I4,7F15.3)*)
      *MK,IV,IFPLOT1,IFPLOT,KN,EXD
      IF (IFPLOT1.EQ.3) EXD1=ABS(EXD)-.71875
      IF (EXD1.LE..5) GO TO 230
      IF (EXD1.GT..5.AND.EXD1.LE..73) GO TO 230
      IFPLOT=2
      GO TO 240
230 CONTINUE
      IFPLOT=1
240 CONTINUE
      IF (IFPLOT1.EQ.3) GO TO 250
245 KN=2
250 CONTINUE
      IF (IV.EQ.1) GO TO 290
      DO 260 I=1,KN
      CALL PLOT(C1,C2,3)
      C2=C2+C21
      CALL PLOT(C1,C2,2)
      C2=C2+C24
      CALL PLOT(C1,C2,3)
```



```
      C2=C2+C23
      CALL PLOT(C1,C2,2)
      C2=C2+C24
260  CONTINUE
      IF(IFPLOT.EQ.2)GO TO 270
265  CALL PLOT(C1,C2,3)
      C2=C2+EXD1
      CALL PLOT(C1,C2,2)
      GO TO 280
270  CONTINUE
      EXD=EXD1
      IFPLOT1=3
      KN=1
      GO TO 220
280  CONTINUE
      C2=C2+.5*C21
      I2=I1X
      I3=I2X
      GO TO 330
290  CONTINUE
      DO 300 I=1,KN
      CALL PLOT(C1,C2,3)
      C1=C1+C21
      CALL PLOT(C1,C2,2)
      C1=C1+C24
      CALL PLOT(C1,C2,3)
      C1=C1+C23
      CALL PLOT(C1,C2,2)
      C1=C1+C24
300  CONTINUE
      IF(IFPLOT.EQ.2)GO TO 310
305  CALL PLOT(C1,C2,3)
      C1=C1-EXD1
      CALL PLOT(C1,C2,2)
      GO TO 320
310  CONTINUE
      EXD=EXD1
      IFPLOT1=3
      KN=1
      GO TO 220
320  CONTINUE
      C1=C1+.5*C21
330  CONTINUE
      RAD=.25
      CALL ARC(C1,C2,RAD,0.,360.)
      IGR2=IGRID1(MK)/100
      IF(IM2.EQ.2)IGR2=MOD(IGRID1(MK),100)
      IF(IBUG.EQ.53)
      *WRITE(6,('(=IOLINE5-1=.4I8)'))IN,MK,IGR2,IGRID1(MK)
      MST=.125
      IF(IV.EQ.1)GO TO 360
      C1=C1-.5*MST
      C2=C2-.5*MST
      K=IGR2
      NTEM=LETER(K)
      IC=1
      IF(IBUG.EQ.53)WRITE(6,('(=IOLINE5-5=.4I4,1X,A2,3F15.3)'))
```

```

      *MK,IV,IGR2,IC,NTEM,C1,C2,HGT
      CALL SYMPO1(C1,C2,HGT,NTEM,C,IC)
      GO TO 600
360  CONTINUE
      NTEM=IGR2
      IC=1
      IF(NTEM.GT.9.AND.NTEM.LE.99)IC=2
      C1=C1-.5*HGT*IC
      C2=C2-.5*HGT
      IF(IPUG.EQ.5)WRITE(6,*(='*LINE5-5*',5I5,3F15.3)*)
      *MK,IV,IGR2,IC,NTEM,C1,C2,HGT
      CALL NUMBER(C1,C2,HGT,FLOAT(NTEM),0.,-1)
600  CONTINUE
      RETURN
      END

```

LE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
3650B	/PLOT1/	REAL		1960	IDAT	74B	//
642P	//	REAL		20	IDOC	60B	//
0B	/PLOT1/	REAL		1960	IDOS	66B	//
27010B	/PLOT1/	REAL		1984	IFBM	1304B	/BMFL4/
2367B	/BMFL4/	REAL		90	IFRM	12B	//
423B	//	REAL		2	IFLAG	2740B	/BMFL4/
440B	/BMFL4/	INTEGER		280	IFLAG1	2524B	/BMFL4/
10B	/BMFL4/	REAL		280	IFLEV	11B	//
23110B	/PLOT1/	REAL		1984	IFPLOT	1112B	
32710B	/PLOT1/	REAL		1984	IFPLOT1	1113B	
36610B	/PLOT1/	REAL		1984	IFR2	6B	/BMFL4/
1116B		REAL			IGRID	4420B	/BMFL4/
52576B	/PLOT1/	REAL		2	IGRID1	4204B	/BMFL4/
1117B		REAL			IGRX	1140B	
1120B		REAL			IGRY	1141B	
52600B	/PLOT1/	REAL		2	IGR2	1125B	
1122B		REAL			ILINK	114B	//
1121B		REAL			INWAY	42510B	/PLOT1/
55B	//	REAL		3	IO9	422B	//
1142B		REAL			IPLOT	3414B	/BMFL4/
46412B	/PLOT1/	REAL		1984	IPRINT	113B	//
115B	//	REAL			IRUN	121B	//
116B	//	REAL			IS	1	DUMMY-AR
7	DUMMY-ARG	REAL			ISOLV	3B	//
1123B		REAL			ITC1	475B	//
1060B	/CLFL2/	REAL		1260	ITEST	120B	//
1305B	/BMFL4/	REAL		280	IW	6	DUMMY-AR
1143B		REAL			IW2	1114B	
1144B		REAL			I1Y	1146B	
1137B		REAL			I2	1147B	
1127B		INTEGER			I26	5	DUMMY-AR
17240B	/PLOT1/	INTEGER		1960	I2X	1150B	
2521B	/BMFL4/	INTEGER			I3	1151B	
1135B		INTEGER			JBR	52602B	/PLOT1/
1070B	/BMFL4/	INTEGER		140	JET	724B	//
103B	//	INTEGER		6	JPL	4203B	/BMFL4/
75B	//	INTEGER		6	JSTOP	111B	//

```
SUBROUTINE LOGO5(IM,IS,MK,IL,IFPL3,IPS,ORGX,ORGY,SCALE)
COMMON MAT,MEM,LEV,ISOLV,LVIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),IO9,CODE(2),MD(20),UF(20)
COMMON ITC1,NEZ(90)
COMMON USE(10),BON(20),LON(20)
COMMON TAXI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/BMFL4/STPX,STPX1,STPY,STPY1,MR1(2),IFR2(2),CORD(140,2),COL1
1,ICFL(140),IFRM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)
5,SIDE(10),MARG,IPLT(25,15),JPL,IGRID1(140),IGRID(140,2)
CHARACTER*10,DATE,FIELD
REAL MARG(10)
INTEGER COL1(140,2)

C
DIMENSION ICC1(48),LOGO(5)
DATA LOGO/4HAPPD,4HCHKD,5HDRAWN,5HSCALE,4HDATE/
XZERO=SIDE(IPS+1)-.03
YZERO=0.
XLOGO=SIDE(IPS+1)-4.5
YLOGO=YZERO+8.0
YLOGO2=YZERO+21.98
YLOGO3=YZERO+21.25
XLOGO1=XLOGO-.02
YLOGO1=YLOGO+.02
CALL PLOT(XLOGO,YZERO,3)
CALL PLOT(XLOGO,YLOGO2,2)
CALL PLOT(XLOGO,YLOGO3,3)
CALL PLOT(XZERO,YLOGO3,2)
CALL PLOT(XLOGO,YLOGO,3)
CALL PLOT(XZERO,YLOGO,2)
CALL PLOT(XLOGO,YZERO+.5,3)
CALL PLOT(XZERO,YZERO+.5,2)
CALL PLOT(XLOGO,YZERO+1.7,3)
CALL PLOT(XZERO,YZERO+1.7,2)
CALL PLOT(XLOGO,YZERO+4.43,3)
CALL PLOT(XZERO,YZERO+4.43,2)
CALL PLOT(XLOGO,YZERO+6.43,3)
CALL PLOT(XZERO,YZERO+6.43,2)
CALL PLOT(XZERO-2.75,YZERO+1.7,3)
CALL PLOT(XZERO-2.75,YZERO,2)
CALL PLOT(XZERO-2.,YZERO+1.7,3)
CALL PLOT(XZERO-2.,YZERO+.5,2)
CALL PLOT(XZERO-1.,YZERO+1.7,3)
CALL PLOT(XZERO-1.,YZERO,2)
CALL PLOT(XZERO-2.75,YZERO+.8,3)
CALL PLOT(XZERO,YZERO+.8,2)
CALL PLOT(XZERO-2.75,YZERO+1.1,3)
CALL PLOT(XZERO,YZERO+1.1,2)
CALL PLOT(XZERO-2.75,YZERO+1.4,3)
CALL PLOT(XZERO,YZERO+1.4,2)
CALL NEWPEN(11)
IF(IM.EQ.1)GO TO 80
IF(IM.EQ.2)GO TO 90
MET=.2
```

```
      IC=10
      XWR=XLOGO-IC*.5*HGT+4.5*.5
      YWR=YZERO+7.315
      CALL SYMBOL(XWR,YWR,HGT,'CARLETON UNIVERSITY',0.,10)
      HGT=.15
      IC=23
      XWR=XLOGO+4.5*.5-IC*HGT*.5
      YWR=YZERO+6.965
      CALL SYMBOL(XWR,YWR,HGT,'CIVIL ENGINEERING DEPT.',0.,20)
      GO TO 50
80  CONTINUE
      HGT=.3
      IC=4
      XWR=XLOGO-IC*.5*HGT+4.5*.5
      YWR=YZERO+7.315
      CALL SYMBOL(XWR,YWR,HGT,'PAID',0.,IC)
      HGT=.15
      IC=23
      XWR=XLOGO+4.5*.5-IC*HGT*.5
      YWR=YZERO+6.965
      CALL SYMBOL(XWR,YWR,HGT,'ENGINEERING CORPORATION',0.,IC)
90  CONTINUE
      HGT=.15
      XWR=XLOGO+1.875
      YWR=YLOGO3+.30
      CALL SYMBOL(XWR,YWR,HGT,'NOTES',0.,5)
      HGT=.0625
      XWR=XLOGO+HGT
      YWR=YZERO+.375
      CALL SYMBOL(XWR,YWR,HGT,'PROJECT NO.',0.,11)
      XWR=XZERO-2.75+HGT
      CALL SYMBOL(XWR,YWR,HGT,'DRAWING NO.',0.,11)
      XWR=XZERO-1.+HGT
      CALL SYMBOL(XWR,YWR,HGT,'REVISION',0.,9)
      HGT=.004
      XWR=YZERO-2.375-2.5*HGT
      AINC=0.
      YWR=YZERO+.65+AINC-HGT*.5
      FIELD=DATE( )
      IF (DAT(1).EQ.0.) READ(FIELD,'(A10)') DAT(1)
      DO 100 ILO=1,5
      ILO1=4
      IF (ILO.EQ.3.OR.ILO.EQ.4) ILO1=5
      NTEM=LOGO(ILO)
      IF (ILO.EQ.5) XWR=XZERO-1.+2.*HGT
      IF (ILO.EQ.5) YWR=YZERO+1.55-HGT*.5
      CALL SYMBOL(XWR,YWR,HGT,NTEM,0.,ILO1)
      IF (ILO.NE.3) GO TO 95
      XWR1=XZERO-1.+HGT
      CALL SYMBOL(XWR1,YWR,HGT,DAT(1),0.,10)
95  CONTINUE
      AINC=0.3
      YWR=YWR+AINC
100 CONTINUE
      NTEM=SCALE
      IC=1
      IF (NTEM.GT.9.AND.NTEM.LE.99) IC=2
```

```

IF (ITEM.GT.99.AND.NTEM.LE.999) IC=3
XWR=YZERO-1.5-(6.+IC)*.5*MGT
YWR=YZERO+1.55-MGT*.5
CALL SYMBOL(XWR,YWR,MGT,AS NOTED,0.08)
MGT=.125

```

```

C
C NME(I,1) I=9,11 KEEPS JOB NO. AND IF USER PUTS AN SLASH
C FOLLOWED BY SS IF STEEL OR SC IF CONCRETE
C NME(I,3) I=9,11 KEEPS REVISION NO.
C

```

```

DO 200 N=1,3,2
IRLA=0
DO 120 I=9,11
IF (NME(I,1).EQ.6H .AND.I.EQ.9) GO TO 160
IF (NME(I,3).EQ.6H .AND.I.EQ.9) GO TO 200
IF (NME(I,N).EQ.6H )IRLA=I-1
IF (IRLA.NE.0) GO TO 130
120 CONTINUE
IRLA=10
130 CONTINUE
IF (IRLA.EQ.9) IRLA=1
IF (IRLA.EQ.10) IRLA=2
IF (IRLA.EQ.11) IRLA=3
DO 132 I6=1,48
ICCI(I6)=10H
132 CONTINUE
NN1=1
NN2=6
IRLA2=IRLA+8
DO 134 II1=9,IRLA2
DECODE(6,887,NME(II1,N))(ICCI(KK),KK=NN1,NN2)
IF (II1.EQ.IRLA2) GO TO 133
NN1=NN1+6
NN2=NN1+5
134 CONTINUE
133 CONTINUE
867 FORMAT(6A1)
IF (N.EQ.3) GO TO 142
N3=IRLA+6
DO 135 I=1,N3
IF (ICCI(I).EQ.1H) GO TO 142
135 CONTINUE
142 CONTINUE
IF (N.EQ.1) GO TO 146

```

```

C
C REVISION NO.
C

```

```

MGT=.125
I2=I-1
I0=1
DO 143 I1=1,I2
IF (ICCI(I1).EQ.1H) GO TO 143
I3=I1
143 CONTINUE
I2=I3
XWR=XZERO-.5-MGT*I2*.5
YWR=YZERO+.1875

```

```
DO 144 I3=1,I2
NTEM=ICCI(I3)
CALL SYMBOL(XWR,YWR,HGT,NTEM,0.,1)
XWR=XWR+HGT
144 CONTINUE
GO TO 200

C
C JOB NO.
C
146 CONTINUE
IC=I-1
IO=2
HGT=1.55/IC
IF (HGT.GT..125) HGT=.125
XWR=XLOGO+1.75+.5-IC*HGT+.5
YWR=YZERO+.1875
DO 150 I1=1,IC
NTEM=ICCI(I1)
CALL SYMBOL(XWR,YWR,HGT,NTEM,0.,1)
XWR=XWR+HGT
150 CONTINUE
160 CONTINUE
HGT=.125
NWOR=3HSS-
IF (MAT.EQ.1) NWOR=3HSC-
XWR=YZERO-1.875-HGT*1.5
YWR=YZERO+.1875
CALL SYMBOL(XWR,YWR,HGT,NWOR,0.,3)
XWR=XWR+HGT*3.
IC=1
IF (IFPL3.GT.9) IC=2
NTEM=IFPL3
CALL NUMBER(XWR,YWR,HGT,FLOAT(NTEM),0.,-1)
200 CONTINUE

C
C CLIENT NAME NME(I,2) I=1,8
C IF SLASH IT IS ENTERED WRITE IN 2 LINES
C
HGT=.2
DO 260 N2=1,3
IBLA=0
DO 235 I=1,8
IF (N2.EQ.3.AND.HGT1.EQ..2) HGT=.15
IF (HGT1.LT..15.AND.N2.EQ.3) HGT=HGT1
IF (NME(I,1).EQ.6H .AND.I.EQ.1) GO TO 260
IF (NME(I,2).EQ.6H .AND.I.EQ.1) GO TO 260
IF (NME(I,N2).EQ.6H ) IBLA=I-1
IF (IBLA.NE.0) GO TO 237
235 CONTINUE
IBLA=8
237 CONTINUE
DO 236 I6=1,48
ICCI(I6)=10H
236 CONTINUE
NW1=1
NW2=6
DO 241 I11=1,IBLA
```

```
DECODE(6.887,NME(III,N2))(ICC1(KK),KK=NN1,N2)
IF(III.EQ.IBLA)GO TO 239
NN1=NN1+6
NN2=NN1+5
241 CONTINUE
239 CONTINUE
N1=IPLA+6
IBLA1=0
DO 238 I=1,N1
IF(ICC1(I).EQ.1H)GO TO 242
238 CONTINUE
DO 240 KK=1,N1
IF(ICC1(KK).EQ.1H)GO TO 241
IC=KK
240 CONTINUE
IBLA1=1
GO TO 243
242 CONTINUE
IC=I-1
243 CONTINUE
IC1=N1
IF(IC1.LE.0)IC1=IC
HGT=4./IC
IF(N2.EQ.3.AND.HGT.GT.HGT1)HGT=HGT1
IF(HGT.GT..2)HGT=.2
K1=IC
IF(IC.EQ.N1)K1=IC-1
IF(N2.EQ.1)HGT1=HGT
XWR=XLOGO+4.5+.5-K1+.5*HGT
YWR1=5.63
IF(N2.EQ.1)YWR1=3.73
IF(N2.EQ.3)YWR1=2.78
IF(N2.EQ.3.AND.(I-1).EQ.IC)YWR1=YWR1-.15
IF(N2.EQ.1.AND.(I-1).EQ.IC)YWR1=YWR1-.1
IF(N2.EQ.2.AND.(I-1).EQ.IC)YWR1=5.43
YWR=YZERO+YWR1+HGT+.5
DO 245 I=1,K1
NTEM=ICC1(I)
CALL SYMBOL(XWR,YWR,HGT,NTEM,0.,1)
XWR=XWR+HGT
245 CONTINUE
IF(IC.EQ.N1)GO TO 250
IF(IBLA1.EQ.1)GO TO 250
K1=IC+2
K2=IC1
DO 247 KK=K1,K2
IF(ICC1(KK).EQ.1H)GO TO 247
K3=KK
247 CONTINUE
K2=K3
YWR1=5.13
IF(N2.EQ.1)YWR1=3.53
IF(N2.EQ.3)YWR1=2.63
XWR=XLOGO+4.5+.5-(K2-K1+1)*.5*HGT
YWR=YZERO+YWR1+HGT+.5
DO 248 I=K1,K2
NTEM=ICC1(I)
```

```

CALL SYMBOL(XWP,YWR,HGT,NTEM,0.,1)
XWR=YWR+HGT
248 CONTINUE
250 CONTINUE
250 CONTINUE
HGT=.15
IC=1P
IF (IS.EQ.2) IC=13
XWR=XZERO-4.5+.5-IC+.5*HGT
YWR=YZERO+2.03
NTEM=10+ELEVATIONS
IF (IS.EQ.2) NTEM=5HPLANS
CALL SYMBOL(XWP,YWR,HGT,*FRAMING*,0.,P)
IC=1C
IF (IS.EQ.2) IC=5
XWR=XWR+P*HGT
CALL SYMBOL(XWP,YWR,HGT,NTEM,0.,IC)
CALL NEUPFN(1)
RETURN
END

```

LE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
1770B			REAL		IFPL3	5	DUMMY-AR
642B	//		REAL	20	IFR2	6B	/BMFL4/
2367B	/BMFL4/		REAL	90	IGRID	4420B	/BMFL4/
423B	//		REAL	2	IGRID1	4204B	/BMFL4/
440B	/BMFL4/		INTEGER	280	II1	2084B	
10B	/BMFL4/		REAL	280	ILINK	114B	//
55B	//		REAL	3	ILO	1771B	
115B	//		REAL		ILO1	1772B	
116B	//		REAL		IM	1	DUMMY-AR
1666B			CHAP*10		IO9	422B	//
1305B	/BMFL4/		REAL	280	IPLOT	3414B	/BMFL4/
1764B			REAL		IPRINT	113B	//
2821B			REAL		IPS	6	DUMMY-AR
1777B			INTEGER		IQ	2012B	
1776B			INTEGER		IRUN	121B	//
2025B			INTEGER		IS	2	DUMMY-AR
2003B			INTEGER		ISOLV	3B	//
2521B	/BMFL4/		INTEGER		ITC1	475B	//
1765B			INTEGER		ITEST	120B	//
1667B			INTEGER	42	IW	4	DUMMY-AR
1070B	/BMFL4/		INTEGER	140	I1	2013B	
103B	//		INTEGER	6	I2	2011B	
75B	//		INTEGER	6	I3	2014B	
2026B			INTEGER		I6	2800B	
74B	//		INTEGER		JET	724B	//
60B	//		INTEGER	6	JPL	4203B	/BMFL4/
66B	//		INTEGER	6	JSTOP	111B	//
1304B	/BMFL4/		INTEGER		KAK	124B	//
12B	//		INTEGER		KET	736B	//
2740B	/BMFL4/		INTEGER	140	KK	2086B	
2524B	/BMFL4/		INTEGER	140	KKPLOT	3154B	/BMFL4/
11B	//		INTEGER		K1	2027B	

SUBROUTINE POSIT(J,IW,MK,K,KA,JMX1,LVNC,C1,C2,C3L,C4L,C1W,C2W,
SCALE)

COMMON MAT,MEH,LEV,ISOLV,LVIN,LNFIN,LVMAX,LNMAX,NL,IFLEV,IFFRM
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(F),ICREC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,TRUN,LVU,LAU
COMMON KAK,LIST(10),IO9,CODF(2),MO(20),UF(20)
COMMON ITC1,NEZ(90)

COMMON USF(10),BON(20),LON(20)

COMMON TAKI(10),JET(10),KET(20),LET(20)

INTEGER KAK(90,2),NME(11,3)

COMMON/PLLOT1/BSTIF(980,2),BLEN(980,2),MKR(980,2),KST(980,2),

1IBR(980,2),CSTIF(992,2),CLEN1(992,2),CY(992,2),CZ(992,2),

2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),

3JBR(980,2)

COMMON/RMFL4/STPX,STPX1,STPY,STPY1,M31(2),IER2(2),CORD(140,2),COL1

1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MCC(140,2)

4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140),KKPLOT(140)

5,SIDE(10),MARG,IPLLOT(25,15),JPL,IGRID1(140),IGRID(140,2)

REAL MARG(10)

INTEGER COL1(140,2)

COMMON/ALFA/ALFA,COSA,ID3

COMMON/RMFL2/SPAN(30,20),XBM(30,20,4),LBM(30,20,4),JMX(30)

COMMON /RMFL1/WIDTH(30,20),DEPTH(30,20)

COMMON/RMFL3/EYE(30,20),ILINE(30)

COMMON/CLFL2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)

COMMON/CLFL3/X(140),Y(140),Z(140),XMAX(140),YMAX(140),KSC(140),

1KDC(140)

IFBM=0

IW3=INVER(IW)

FACTOR=.75

HGT=.125*FACTOR

IBUG=LET(11)

CWEB=1.5*FCL(MK,4)/(24.*SCALE)

IF(MAT.EQ.1)CWEB=1.5*X(MK)/(24.*SCALE)

IF(MAT.EQ.1.AND.KSC(MK).EQ.3)CWEB=1.5*Y(MK)/(24.*SCALE)

IF(1BUG.EQ.53)

1PRINT 990,J,IW,MK,K,KA,JMX1,C1,C2,C3L,C4L,C1W,C2W

990 FORMAT(* IQ POSIT1 *,6I4,6F3)

CLENG=HGT

IF(MK.GT.9)CLENG=2.*HGT

IF(MK.GT.99)CLENG=3.*HGT

C5L=C3L+CLENG

IF(IW.EQ.1)MARK=MOD(MKB(LVNC,IW),10000)

IF(IW.EQ.2)MARK=MKB(LVNC,IW)/10000

LI2=MARK/100

IS2=MARK-LI2*100

KASE=1

BLENG=3.*HGT

IF(MARK.GT.920)BLENG=4.*HGT

BLENG1=C1W-BLENG*.5-HGT/3.

LY1=LY(MK)

JY1=JY(MK)

IF(IW.EQ.1)GO TO 20

LY1=LI2

JY1=IS2

LI2=LX(MK)

IS2=JX(MK)

```

        IW3=IW
    20 CONTINUE
        IF(LI2.EQ.0.AND.LY1.EQ.0)GO TO 510
        SPANP=SPAN(LI2,IS2)/(12.*SCALE)
        IF(IW.EQ.2)SPANP=SPAN(LY1,JY1)/(12.*SCALE)
        IF(PLENG.GT.SPANP)IFBM=2
        IF(IFBM.EQ.2.AND.SPANP.LT..75*PLENG)IFBM=3
        IF(FLAG1(MK).EQ.1)GO TO 500
        IF(IPL0T(14,J).EQ.0)GO TO 500
        IF(LI2.EQ.0)GO TO 70
        IF(SPAN(LI2,IS2).LE.0.)GO TO 70
        IF(WIDTH(LI2,IS2).LT.2..AND.MAT.EQ.1)GO TO 70
        IF(IRUG.EQ.53)PRINT 992,KASE,LI2,IS2,MARK,WIDTH(LI2,IS2),
        *EYE(LY1,JY1)
        IF(IRUG.EQ.53)
        1PRINT 992,KASE,MK,IW,MARK,C4L,BLENG1,C3L,C4L,BLENG1
    992 FORMAT(* POSIT *,4I6,5F8.3)
        IF(C4L.GT.BLENG1.AND.ALFA.EQ.0.)GO TO 100
    70 IF(JY1.LE.1)GO TO 500
        IF(SPAN(LY1,JY1-1).EQ.0)GO TO 500
        IF(EYE(LY1,JY1-1).LE..01)GO TO 500
        BLENGY=3.*HGT
        IF(LY1.GT.9)BLENGY=4.*HGT
        BLENG2=(CORD(MK,IW3)-SPAN(LY1,JY1-1)/24.)/SCALE+BLENGY+.5*HGT/3.
        IF(IRUG.EQ.53)
        1PRINT 992,KASE,MK,IW,MARK,C4L,BLENG2
        IF(C4L.GE.BLENG2)GO TO 500

C
C TRY UPPER RIGHT CORNER
C
    100 CONTINUE
        MARKY=LY1+100+JY1
        C1L=C3L
        C2L=2.*CORD(MK,IW3)/SCALE-C4L-HGT
        IF(IW.EQ.2)C2L=2.*CORD(MK,IW)/SCALE-C4L-HGT
        C5L=C2L+HGT
        IF(IPL0T(18,J).EQ.1)GO TO 140
        KASE=2
        IF(JMX1.EQ.IS2)GO TO 500
        IF(SPAN(LY1,JY1).EQ.0)GO TO 500
        IF(EYE(LY1,JY1).LE..01)GO TO 500
        BLENGY=3.*HGT
        IF(LY1.GT.9)BLENGY=4.*HGT
        BLENGY1=(CORD(MK,IW3)+SPAN(LY1,JY1)/24.)/SCALE-.5*BLENGY-HGT/3.
        IF(IRUG.EQ.53)
        1PRINT 992,KASE,MK,IW3,MARKY,C5L,BLENGY1
        IF(C5L.LT.BLENGY1)GO TO 500

C
C TRY LOWER LEFT CORNER
C
    140 CONTINUE
        KASE=3
        C1L=2.*CORD(MK,IW)/SCALE-C3L-CLENG+.75
        C2L=C4L
        IF(IW.EQ.1)GO TO 145
        C1L=2.*CORD(MK,IW)/SCALE-C3L-CLENG+.75
    145 IF(K.EQ.KA)GO TO 500
    
```

```

SPAN1=0.
IF (LI2.EQ.0.AND.IPLOT(18,J).EQ.0) GO TO 500
IF (LI2.EQ.0) GO TO 150
SPAN1=SPAN(LI2,IS2-1)
IF (SPAN1.EQ.0.AND.IPLOT(18,J).EQ.0) GO TO 500
150 C4W=(CORD(MK,IM)-SPAN1/24.)/SCALE+.5*BLENG+HGT/3.
IF (IPLOT(18,J).EQ.0) GO TO 155
IF (C1L.GE.C4W.AND.C2L.GE.BLENG2) GO TO 500
155 C2L=2.*CORD(MK,IM3)/SCALE-C4L-HGT
IF (IW.EQ.2) C2L=2.*CORD(MK,IW)/SCALE-C3L-HGT
IF (IPLOT(18,J).EQ.0) GO TO 510
C SPAN1=SPAN(LI2,IS2)/(12.*SCALE)
C IF (SPAN1.LE.BLENG) GO TO 160
C GO TO 510
C
C REDUCE BEAM MARK SIZE TO 3/4 OF HGT
C
160 CONTINUE
KASE=4
C5L=C1L+CLENG
BLENG=.75*BLENG
BLENG1=C1W-BLENG+.5-HGT/4.
IF (C5L.LT.BLENG1) GO TO 500
200 CONTINUE
C
C WRITE FOOT NOTE
C
IFBM=3
KASE=5
500 CONTINUE
IF (KASE.EQ.4) IFBM=2
IF (KASE.EQ.1) GO TO 510
C3L=C1L
C4L=C2L
510 CONTINUE
IF (IRUG.EQ.53)
1PRINT 991,J,MK,MKB(LVNC,IM),IFBM,KASE,C3L,C4L,C1W,C2W
2,BLENG1
991 FORMAT(' IQ POSITS ',5I9,5F9.3)
RETURN
END

```

IF MAP--(LO=A)

-ADDRESS--BLOCK----		-PROPERTIES-----		-TYPE----	-SIZE----	-NAME---ADDRESS--BLOCK---	
0B	/ALFA/			REAL		CLEV	2367R /BMFL4/
3650B	/PLOT1/			REAL	1960	CODE	423B //
643B				REAL		COL1	448R /BMFL4/
650B				REAL		CORD	18B /BMFL4/
1 655B				REAL		COSA	1B /ALFA/
644B				REAL		CSTIF	23110R /PLOT1/
651B				REAL		CWEB	634R
642B	//			REAL	20	CX	32710R /PLOT1/
0R	/PLOT1/			REAL	1960	CZ	36610R /PLOT1/
635B				REAL		C1	0 DUMMY-AR
27010B	/PLOT1/			REAL	1964	C1L	653B

SUBROUTINE R53(I60,LU,LIG,JIG,NIG,LVR)

C

C COPYRIGHT (C) 1983 AMECO COMPUTING SYSTEMS INC.

C

INTEGER XRM

C COMMON MAT, MEM, LEV, ISOLV, LMF IN, LNF IN, LV MAX, LN MAX, NL, IFLEV, IFFRM

COMMON ZZZZ, NME, DAT(3), IDOC(6), IDOS(6), IDAT, ICRIS(4), ICRIC(6)

COMMON JSTOP, NSTOP, IPRINT, ILINK, FC1, FC5, MAXFR, ITEST, IRUN, LVU, LNU

COMMON KAK, LIST(10), IO9, CODE(2), MD(20), UF(20), ITC1

COMMON NEZ(90), USF(19), BON(20), LON(20)

COMMON TAKI(10)

INTEGER NME(11,3)

INTEGER KAK(90,2), NEO

C

C

COMMON /BMFL1/ WIDTH(30,20), DEPTH(30,20)

COMMON /BMFL2/ SPAN(30,20), XEM(30,20,4), LBM(30,20,4), JMX(30)

COMMON /BMFL3/ EYE(30,20), ILINE(30)

COMMON /BMFL4/ PJN(30,21,4)

COMMON /PNLN1/ FC1, FY1

COMMON /RIG1/ IGI(30,20)

COMMON /BRACE/ IBRA(30,20), LAR(5)

COMMON /CLFL1/ FC5, FY5, CVR5, FKC(9)

COMMON /CLFL2/ LX(140), JX(140), LY(140), JY(140), FCL(140,9)

COMMON /CLFL3/ X(140), Y(140), Z(140), XMAX(140), YMAX(140), KSC(140)

COMMON /KDC(140)

COMMON /CLFL6/ NOX, NEO(140), MAX(140)

COMMON /CLFL5/ KBAR(140,2)

COMMON /CLFL7/ MCD(140,3), XY(9), STUB(9), IFO(9), XK(9), YK(9), INX(9)

COMMON /CLFL8/ TOR(140,2), TORM(21)

COMMON /THE TFL/ THE DL(30,21), THE LL(30,21,2)

COMMON /RIRS/ IRI B, RIR(B)

COMMON /MOVE/ MOVNO, XM(21,10), PM(21,10), MSTART(10,10),

MEND(10,10), IREVE(10), NWH(10), PACT(10), MIT(10)

COMMON /ALS/ ALFC(4), ALFY(4)

COMMON /BRCS/ IBRC(10), EBRC(10)

COMMON /GEN1/ ESTI(12,4), JYPE(30,2), KANT(30,2)

COMMON /CPRS/ CPRD(140)

COMMON /CONSTS/ CONSTS(10), IE OF, ACC(3)

C

IF(I60.EQ.2) GO TO 100

IF(ACC(3).EQ.2) GO TO 12

C

READ(LU) LIG, JIG, NIG, ((P, JN(L, M, I), I=1,4), THE DL(L, P),

1(THE LL(L, M, I), I=1,2), M=1, JIG), L=1, LIG)

2,(TOR(N,1), TOR(N,2), N=1, NIG), FC1

12

CONTINUE

READ(LU) LIG, JIG, NIG, (LX(N), N=1, NIG), (JX(N), N=1, NIG), (LY(N), N=1, NIG), (JY(N), N=1, NIG)

C

IF(ILINK.EQ.3) READ(LU) KBAR

READ(LU) LVR, FC5, FY5, CVR5, FKC, LN FL, MOVNO, (X(N), N=1, NIG), (Y(N), N=1, NIG), (Z(N), N=1, NIG), (XMAX(N), N=1, NIG), (YMAX(N), N=1, NIG), (KSC(N), N=1, NIG), (KDC(N), N=1, NIG), (FCL(N, M), N=1, NIG), M=1, 9), (MCD(N, M), N=1, 3), XY, STUB, IFO, XK, YK, INX,

CPRS(N), N=1, NIG), ESTI, IRI B, RIR,

((JYPE(L, M), KANT(L, M), L=1, LIG), M=1, 2), LON, BON,

```

D((XPM(M,N,K),M=1,LIG),N=1,JIG),(LBM(M,N,K),M=1,LIG),N=1,JIG),K=1
E,9),(SPAN(M,N),M=1,LIG),(EYE(M,N),M=1,LIG),(WIDTH(M,N),M=1,LIG),
F(DEPTH(M,N),M=1,LIG),N=1,JIG),(JMX(M),M=1,LIG),(ILINE(M),M=1,LIG)
H,((IGI(M,N),M=1,LIG),N=1,JIG)
G,((IPRA(M,N),M=1,LIG),N=1,JIG)
J,(NEQ(N),MAK(N),N=1,NIG),NOY
K,ALFC,ALFY,IPRC,EBRC

```

C

```

IF(MOVNO.GT.0)READ(LU)(PACT(J),MIT(J),J=1,MOVNO),
,((XM(K,J),K=1,21),(PM(K,J),K=1,21),
1(MSTART(K,J),K=1,10),(MEND(K,J),K=1,10),J=1,MOVNO),(IREVE(J),J=1,
2MOVNO),(NUH(J),J=1,MOVNO)

```

C

RETURN

C

100 CONTINUE

C

C

RETURN

END

FILE MAP--(LO=A)

--ADDRESS--	--BLOCK--	----	PROPERTIES-----	-----TYPE-----	-----SIZE-----	----	NAME----	ADDRESS--	--BLOCK--
13B	/CONSTS/		REAL		3		IFD	666B	/CLFL7/
0B	/ALS/		REAL		4		IGI	0B	/RIGI/
4B	/ALS/		REAL		4		IFD	1	DUMMY-AR
642B	//		REAL		20		ILINE	1130B	/BMFL3/
423B	//		REAL		2		ILINK	114B	//
0B	/CONSTS/		REAL		10		INX	721B	/CLFL7/
0B	/CPROS/		REAL		140		IO9	422B	//
2B	/CLFL1/		REAL				IPRINT	113B	//
55B	//		REAL		3		IREVE	1155B	/MOVE/
1130B	/BMFL1/		REAL		600		IRIP	0B	/RIPS/
12B	/PRCS/		REAL		10		IRUN	121B	//
115B	//		REAL				ISOLV	3B	//
116B	//		REAL				ITC1	475B	//
0B	/GEN1/		REAL		4B		ITEST	120B	//
0B	/BMFL3/		REAL		600		J	1273B	
1060B	/CLFL2/		REAL		1260		JIG	4	DUMMY-AR
0B	/BMFL1/		REAL				JMX	12430B	/BMFL2/
0B	/CLFL1/		REAL				JSTOP	111B	//
3B	/CLFL1/		REAL		9		JX	214B	/CLFL2/
1B	/BMFL1/		REAL				JY	644B	/CLFL2/
1B	/CLFL1/		REAL				JYPE	68B	/GEN1/
1227B			INTEGER				K	1252B	
0B	/BRACE/		INTEGER		600		KAK	124B	//
0B	/PRCS/		INTEGER		1B		KANT	154B	/GEN1/
103B	//		INTEGER		6		KBAR	0B	/CLFL5/
75B	//		INTEGER		6		KDC	1510B	/CLFL3/
74B	//		INTEGER				KSC	1274B	/CLFL3/
60B	//		INTEGER		6		L	1223B	
64B	//		INTEGER		6		LAR	1130B	/BRACE/
12B	/CONSTS/		INTEGER				LBM	5670B	/BMFL2/
12B	//		INTEGER				LEV	2B	//
11B	//		INTEGER				LIG	3	DUMMY-AR

```

SUBROUTINE SECBM(ISHA,J,IWMX,LIG,NIG,LV2,LVM4)
COMMON MAT,MEN,LEV,ISOLV,LVIN,LAFIN,LVMAX,LNMAX,NL,IFLEV,IFFR
COMMON ZZZZ,NME,DAT(3),IDOC(6),IDOS(6),IDAT,ICRIS(6),ICRIC(6)
COMMON JSTOP,NSTOP,IPRINT,ILINK,EC1,EC5,MAXFR,ITEST,IRUN,LVU,LNU
COMMON KAK,LIST(10),ID9,CODE(2),WD(20),UF(20)
COMMON ITC1,NEZ(90)
COMMON USF(10),PON(20),LON(20)
COMMON TAKI(10),JET(10),KET(20),LET(20)
INTEGER KAK(90,2),NME(11,3)
COMMON/PL0T1/BSTIF(980,2),BLEN(980,2),MKP(980,2),KST(980,2),
1IPR(980,2),CSTIF(992,2),CLEAI(992,2),CX(992,2),CZ(992,2),
2INWAY(2),PHIC(992,2),DV(992,2),TRANS(90,2),C11(2),C22(2),
3JPR(980,2)
COMMON/ALFA/ALFA,COSA,ID3
COMMON/BMFL1/WIDTH(30,20),DEPTH(30,20)
COMMON/BMFL2/SPAN(30,20),XB(30,20,4),LBM(30,20,4),JMX(30)
COMMON/CLFL2/LX(140),JX(140),LY(140),JY(140),FCL(140,9)
COMMON/BMFL4/STPX,STPX1,STPY,STPY1,MB1(2),IFR2(2),CORD(140,2)
1,COL1,ICFL(140),IFBM,GRID1(140,2),NCOL(2),MKC(140,2)
4,CLEV(90),IBUG,STZZ1,STZZ,IFLAG1(140),IFLAG(140)
5,KKPL0T(140),SIDE(10),MARG,IPL0T(25,15),JPL,IGRID1(140)
6,IGRID(140,2),DUMMY(240)
COMMON/REIZ1/KUZ,UZ1,UZ2,UZ3,UZ4,UZ5,UZ6,UZ7,UZ8,UZ9,UZ10,UZ11,
1UZ12,UZ13,UZ14,UZ15,UZ16,UZ17,UZ18
REAL MARG(10)
INTEGER COL1(140,2)
DIMENSION CC(10)
COMMON/THETFL/DUM(1890)
DIMENSION IFB(250),XB1(250),YB1(250),MKP1(250),BMSH(21)
1,BIMSH(30,20),OB2(250),RIM(240)
EQUIVALENCE (DUM(1),IFB(1)),(DUM(251),XB1(1)),(DUM(501),YB1(1))
EQUIVALENCE (DUM(751),MKP1(1))
EQUIVALENCE (DUM(1001),BMSH(1)),(DUM(1022),BIMSH(1,1))
EQUIVALENCE (DUM(1622),OB2(1)),(DUM(1872),INUMX)
EQUIVALENCE (DUM(1),RIM(1))
IBUG=LET(11)
CALL NEWPEN(11)
SCALE=IPL0T(20,J)
GAP1=.047
IF(MAT.EQ.1)GAP1=0.
IF(SCALE.EQ.0.)SCALE=20.
FACTOR=.75
HGT=.125*FACTOR
IF(ISHA.EQ.2)GO TO 600
INU=0
IF(ISHA.EQ.3)GO TO 20
IF(IPL0T(18,J).EQ.0.AND.IPL0T(19,J).EQ.0)GO TO 30
DO 10 LM=1,LIG
CALL REIDMS(12,BMSH,21,LM)
JMX1=JMX(LM)-1
DO 8 JM=1,JMX1
BIMSH(LM,JM)=BMSH(JM)
IF(IBUG.EQ.53)PRINT 990,LM,JM,BIMSH(LM,JM),BMSH(JM)
990 FORMAT(' POSIT IC1',2I6,2(1X,A20))
8 CONTINUE
10 CONTINUE
GO TO 30

```

```

20 CONTINUE
  CALL ENCO(LIG)
30 CONTINUE
  IF(IPL0T(16,J).EQ.0)GO TO 700
  DO 320 IW=1,IWMX
    MB=MB1(IW)
    IFR=IFR2(IW)
    IW1=INVER(IW)
    DO 320 I=1,IFR
      KA=COL1(I,IW)
      IF(I.LT.IFR)KB=COL1(I+1,IW)-1
      IF(I.EQ.IFR)KB=MB
      DO 320 K=KA,KB
        LN=(K-1)*LVN4+LV2
        MK=MKC(K,IW)
        MK2=MKB(LN,IW)
        IF(ISTIF(LN,IW).LE.1.E-15.AND.IPL0T(9,J).EQ.0)GO TO 320
        IF(IW.EQ.1)MARK=MOD(MK2,10000)
        IF(IW.EQ.2)MARK=MK2/10000
        LI1=MARK/100
        IS1=MARK-LI1*100
        DBM1=0.
        RATIO=.5
        IF(MAT.EQ.2)GO TO 40
        IF(DEPTH(LI1,IS1).EQ.0.)GO TO 40
        RATIO=WIDTH(LI1,IS1)/DEPTH(LI1,IS1)
40 CONTINUE
        IF(BUG.EQ.53)
          1PRINT 852,K,MK,MK2,IW,MARK,LI1,IS1,IFBB(LI1)
852 FORMAT(10 I0 1*,8I8)
        IF(IFBB(LI1).EQ.2)GO TO 300
        DO 280 KN=1,8
          IJL=JLB*5(LI1,IS1,KN)
          IF(BUG.EQ.53)PRINT 852,KN,LI1,IS1,IJL,IFBB(LI1),INU
          IF(IJL.EQ.0)GO TO 300
          LI2=IJL/100
          IS2=IJL-LI2*100
          DBM=FXB*5(LI1,IS1,KN)/12.
          DBM1=DBM1+DBM
          IF(WIDTH(LI2,IS2).LE.1.)GO TO 280
          IF(SPAN(LI2,IS2).EQ.0.)GO TO 280
          INU=INU+1
          MKB1(INU)=IJL+IW1*10000
          CALL UNP53(KST(LN,IW),A1,A2,B)
          HGT=.125*FACTOR
          BM=SPAN(LI2,IS2)/(12.*SCALE)
          BM1=SPAN(LI1,IS1)/(12.*SCALE)
          GAP2=0.
          IF(MAT.EQ.2)GO TO 60
          GAP1=WIDTH(LI1,IS1)/(24.*SCALE)+.023
          GAP2=WIDTH(LI2,IS2)/(24.*SCALE)
          IF(BUG.EQ.53)WRITE(6,('+'IOSECB1+' .5I5,2F10.3)')
          1MARK,LI1,IS1,LI2,IS2,GAP1*120./UZ10,GAP2*120./UZ10
60 CONTINUE
        IF(IW.EQ.2)GO TO 85

```

C MAIN BEAM IN XDIRECTION

C

333 FORMAT(10A1)

DECODE(10,333,PIMSH(LI1,IS1))CC

DO 70 IC=1,10

IF(CC(IC).EQ.1M)GO TO 80

70 CONTINUE

80 CONTINUE

IC1=IC-1

SLENG=IC1*HGT

XR1(INU)=DBM1*COXA/SCALE+CORD(MK,IW)/SCALE

YR1(INU)=CORD(MK,IW1)/SCALE+DRM1*ALFA/(SCALE)

C1=XR1(INU)

C2=YR1(INU)+GAP1

C5=C1

C6=RM+YR1(INU)

BLENG1=CORD(MK,IW)/SCALE+RM1*.5-.5*SLENG-HGT/3.

BLENG2=BLENG1+SLENG+2.*HGT/3.

IF(IPL0T(18,J).EQ.0)GO TO 90

IF(BSTIF(LN,IM).EQ.10.)GO TO 90

IF(BSTIF(LN,IM).LE.1.E-15)GO TO 90

IF(RATIC.6E.5..AND.MAT.EQ.1)GO TO 90

IF(C1.LT.BLENG1)GO TO 90

IF(C1.GT.BLENG2)GO TO 90

C2=C2+HGT+.047

GO TO 90

C

C MAIN BEAM IN YDIRECTION

C

85 CONTINUE

XB1(INU)=CORD(MK,IW1)/SCALE-DBM1*ALFA/SCALE

YB1(INU)=DRM1*COXA/(SCALE)+CORD(MK,IW)/SCALE

BLENG=3.*HGT

IF(LI1.GT.9)BLENG=4.*HGT

C1=XB1(INU)+GAP1

C2=YB1(INU)

C5=XB1(INU)+BM

C6=YB1(INU)

BLENG1=CORD(MK,IW)/SCALE+RM1*.5-BLENG*.5-HGT/3.

BLENG2=BLENG1+BLENG+2.*HGT/3.

IF(IPL0T(14,J).EQ.0)GO TO 90

IF(C2.LT.BLENG1)GO TO 90

IF(C2.GT.BLENG2)GO TO 90

C1=C1+HGT+.047

90 CONTINUE

C

C FIND RIGHT END OF SEC. BM

C CHECK IF IT IS SUPP. IN MAIN BM OR IN SECOND BM.

C

IJL1=IJL+1

DO 160 I4=1,LIG

I3=LIG-I4+1

JMX2=JMX(I3)-1

IF(JMX2.LT.0)GO TO 160

DO 140 I2=1,JMX2

CALL UNSQZB(2,IFBB(I3),I2,IRI,IER)

DO 120 KKN=1,8

IJL2=JLPM5(I3,I2,KKN)


```

      IF(IJL2.EQ.0)GO TO 140
      IF(IJL1.NE.IJL2)GO TO 120
      IF(IPI.NE.1)GO TO 130
      DO 100 N1=1,N16
      HGT1=HGT
      LX1=LX(N1)
      JX1=JX(N1)
      IF(IW.EQ.2)LX1=LY(N1)
      IF(IW.EQ.2)JX1=JY(N1)
      IF(IPUG.EQ.53)PRINT 858,N1,*3,I2,LX1,JX1
858  FORMAT(* I07 *,9I8)
      IF(LX1.NE.I3)GO TO 100
      IF(JX1.NE.I2)GO TO 100
      IF(N1.LT.MK)GO TO 140
      BLENG=3.*HGT
      IF(LX1.GT.9)BLENG=4.*HGT
      IF(IW.EQ.1)GO TO 98
      DECODE(10,333,RIMSH(I3,I2))CC
      DO 95 IC=1,10
      IF(CC(IC).EQ.1H)GO TO 96
95  CONTINUE
96  CONTINUE
      IC1=IC-1
      BLENG=IC1*HGT1
98  CONTINUE
      GO TO 137
100 CONTINUE
      PRINT 840
840  FORMAT(3H***,*MESSAGE 1 SEC. BEAMS *,3H***,* ERROR*)
      STOP
120 CONTINUE
      GO TO 140
130 CONTINUE
      HGT1=.75*HGT
      BLENG=3.*HGT1
      IF(I3.GT.9)BLENG=4.*HGT1
      IF(IW.EQ.1)GO TO 138
      DECODE(10,333,RIMSH(I3,I2))CC
      DO 132 IC=1,10
      IF(CC(IC).EQ.1H)GO TO 134
132 CONTINUE
134 CONTINUE
      IC1=IC-1
      BLENG=IC1*HGT1
      GO TO 138
137 CONTINUE
      BLENG3=(CORD(N1,IW)+SPAN(I3,I2)*.5/12.)/SCALE-.5*BLENG-HGT1/3.
      GO TO 139
138 CONTINUE
      BLENG3=(CORD(MK,IW)+SPAN(I3,I2)*.5/12.)/SCALE-.5*BLENG-HGT1/3.
139 CONTINUE
      BLENG4=BLENG3+BLENG+2.*HGT1/3.
      GO TO 180
140 CONTINUE
160 CONTINUE
180 CONTINUE
      IF(MAT.EQ.2)GO TO 190

```

GAP1=WIDTH(I3,I2)/(24.*SCALE)+.023

GAP2=WIDTH(LI2,IS2)/(24.*SCALE)

190 CONTINUE

IF(IW.EQ.2)GO TO 220

C6=C6-GAP1

IF(IPL0T(9,J).EQ.0.AND.PSTIF(LN,IW).LE.1.E-15)GO TO 240

IF(IPL0T(9,J).EQ.0.AND.WIDTH(I3,I2).LT.2..AND.MAT.EQ.1)GO TO 240

IF(IPL0T(14,J).EQ.0.AND.IBI.EQ.1)GO TO 240

IF(IPL0T(18,J).EQ.0.AND.IBI.NE.1)GO TO 240

IF(C5.LT.BLENG3)GO TO 240

IF(C5.GT.BLENG4)GO TO 240

C6=C6-MGT1-.047

GO TO 240

220 CONTINUE

C

C Y DIRECTION (MAIN BEAM)

C

IF(DEPTH(I3,I2).EQ.133.)GO TO 240

IF(DEPTH(I3,I2).LE.0.)GO TO 240

C5=C5-GAP1

IF(RATIO.GE.5..AND.MAT.EQ.1)GO TO 240

IF(WIDTH(I3,I2).LT.2..AND.MAT.EQ.1)GO TO 240

IF(IPL0T(18,J).EQ.0.AND.IBI.EQ.1)GO TO 240

IF(IPL0T(19,J).EQ.0.AND.IBI.NE.1)GO TO 240

IF(C6.LT.BLENG3)GO TO 240

IF(C6.GT.BLENG4)GO TO 240

C5=C5-MGT1-.047

240 CONTINUE

IF(IRUG.EQ.53)

1PRINT R60,MARK,IW,IJL,INU,C1,C2,C5,C6,BLENG1,BLENG2,BLENG3,BLENG4

860 FORMAT(' I98',4I6,8F8.3)

IF(MAT.EQ.2)GO TO 260

IF(IRUG.EQ.53)WRITE(6,('(I0SECB2',5I5,2F10.3)')

1MARK,LI2,IS2,I3,I2,GAP1*120./UZ10,GAP2*120./UZ10

C11(1)=C1-GAP2

C11(2)=C5-GAP2

C22(1)=C2

C22(2)=C6

IF(IW.EQ.1)GO TO 245

C11(1)=C1

C11(2)=C5

C22(1)=C2-GAP2

C22(2)=C6-GAP2

245 CONTINUE

CALL DASH(C11,C22,2,.1,.1)

IF(IW.EQ.2)GO TO 247

C11(1)=C1+GAP2

C11(2)=C5+GAP2

GO TO 250

247 CONTINUE

C22(1)=C2+GAP2

C22(2)=C6+GAP2

250 CONTINUE

CALL DASH(C11,C22,2,.1,.1)

GO TO 265

260 CONTINUE

CALL PLOT(C1,C2,3)

```

      CALL PL0T(05,C6,2)
265 CONTINUE
      CALL SQZBITS(2,IFBB(LI2),IS2,2,IER)
      OR2(INU)=C2
      PIM(INU)=C6-C2
      IF(IV.EQ.1)GO TO 270
      OR2(INU)=C1
      PIM(INU)=C5-C1
270 CONTINUE
280 CONTINUE
300 CONTINUE
320 CONTINUE
      DO 520 I4=1,LIG
      MKK=MKK1+0
      DO 500 I1=1,LIG
      I13=LIG-I1+1
      JMX1=JMX(I13)-1
      IF(JMX1.LT.0)GO TO 500
      DO 480 I12=1,JMX1
      CALL UNSQZB(2,IFBB(I13),I12,IPI,IER)
      IF(IBUG.EQ.53)
1PRINT 855,I1,I13,LIG,JMX1,I12,IPI,INU
      IF(IBM.EQ.1)GO TO 500
      IF(IBM.EQ.0)GO TO 480
      HGT1=.75*HGT
      DBM1=0
      DO 460 KN=1,K
      IF(KN.EQ.1)KNN=1
      IJL=JLBM5(I13,I12,KN)
C
C FIND SECOND. BEAM SUPPORTED IN BM.(SECONDARY) MKB2
C DIRECTION IN2 OF BM MKB2
C
      IF(IPUG.EQ.53)
1PRINT 855,I1,I13,JMX1,I12,IPI,IJL
855 FORMAT(' I0 4',8IP)
      IF(IJL.EQ.0)GO TO 480
      LI1=IJL/100
      IS1=IJL-LI1*100
      CALL SQZBITS(2,IFBB(I13),I12,2,IER)
      CALL UNSQZB(2,IFBB(LI1),IS1,IPI,IER)
      IF(BUG.EQ.53)PRINT 855,IJL,IBM,LI1,IS1,KNN,I13,I12,IBM
      IF(IBM.NE.0.AND.KNN.EQ.1)GO TO 480
      DBM=FXBM5(I13,I12,KN)/12.
      DBM1=DBM+DBM1
      MKK1=I13+100+I12
      IF(MKK.NE.MKK1)KK1=0
      IF(SPAN(LI1,IS1).LE.8.)KNN=1
      IF(SPAN(LI1,IS1).EQ.0.)GO TO 460
      IF(WIDTH(LI1,IS1).LE.1.)GO TO 460
      IF(KK1.NE.0)GO TO 380
      MKK=I13+100+I12
      DO 360 KK=1,INU
      MKB2=MOD(MKB1(KK),10000)
      IN2=MKB1(KK)/10000
      IF(BUG.EQ.53)PRINT 856,KK,INU,MKK,MKB2,MKB1(KK),IJL
856 FORMAT(' I0 6',8IB)

```

```

      IF(MKB2.EQ.MMK)GO TO 380
380  CONTINUE
      PRINT RFO
      850 FORMAT( 3H***, ' MESSAGE SEC. BEAMS ', 3H***, ' ERROR ')
      STOP
390  CONTINUE
C
C  MKB2 PM SUPPORTING NEW SECOND. PM
C  IM2=1 XDIRECTION SUPPORTING PM.
C

```

```

      KK1=MK
      INU=INU+1
      BM=SPAN(LI1,IS1)/(12.*SCALE)
      BM1=SPAN(II3,II2)/(12.*SCALE)
      RATIO=.5
      IF(MAT.EQ.2)GO TO 383
      IF(DEPTH(II3,II2).EQ.0.)GO TO 383
      RATIO=WIDTH(II3,II2)/DEPTH(II3,II2)
383  CONTINUE
      IM3=INVER(IM2)
      GAP2=0.
      IF(MAT.EQ.2)GO TO 382
      GAP1=WIDTH(II3,II2)/(24.*SCALE)+.023
      GAP2=WIDTH(LI1,IS1)/(SCALE*24.)
382  CONTINUE
      IF(IM3.EQ.1)GO TO 390

```

```

C
C  NEW SECOND. BEAM YDIRECTION
C
      DECODE(10,333,BIMSH(II3,II2))CC
      DO 385 IC=1,10
      IF(ICC(IC).EQ.1P)GO TO 386

```

```

385  CONTINUE
386  CONTINUE
      IC1=IC-1
      BLENG=IC1*HGT1
      XB1(INU)=XB1(KK)+DBM1/SCALE
      YB1(INU)=YB1(KK)
      IF(SPAN(LI1,IS1).EQ.0.)GO TO 460
      IF(WIDTH(LI1,IS1).LE.1.)GO TO 460
      C1=XF1(INU)
      C2=YB1(INU)+GAP1
      C5=XB1(INU)
      C6=YB1(INU)+BM
      BLENG1=XB1(KK)+BM1+.5-.5*BLENG-HGT1/3.
      BLENG2=BLENG1+BLENG+2.*HGT1/3.
      IF(RATIO.GE.5..AND.MAT.EQ.1)GO TO 410
      IF(WIDTH(II3,II2).LT.2..AND.MAT.EQ.1)GO TO 410
      IF(DEPTH(II3,II2).EQ.133.)GO TO 410
      IF(IPL0T(19,J).EQ.0)GO TO 410
      IF(C1.LT.BLENG1)GO TO 410
      IF(C1.GT.BLENG2)GO TO 410
      C2=C2+HGT1+.047
      GO TO 410

```

```

C
C  NEW SECONDARY BEAM IN XDIRECTION
C

```

300 CONTINUE

XR1(INU)=XR1(KK)

YR1(INU)=YR1(KK)+DBM1/SCALE

IF (SPAN(I1,I3).EQ.0.)GO TO 460

IF (WIDTH(I1,I3).LE.1.)GO TO 460

C1=YR1(INU)+GAP1

C2=YR1(INU)

C5=XR1(INU)+PM

C6=YR1(TAU)

BLENG=3.*HGT1

IF (I13.GT.9)BLENG=4.*HGT1

PLENG1=YR1(KK)+PM1+.5+.5*BLENG-HGT1/3.

PLENG2=PLENG1+BLENG+2.*HGT1/3.

IF (IPL0T(I5,J).EQ.0)GO TO 410

IF (IPL0T(I,J).EQ.0.AND.WIDTH(I13,I12).LT.2..AND.MAT.EQ.1)GO TO 410

IF (C2.LT.PLENG1)GO TO 410

IF (C2.GT.PLENG2)GO TO 410

C1=C1+HGT1+.047

410 CONTINUE

C

C FIND RIGHT SUPPORT (I1J1) OF BEAM I1J

C

I1J1=I1J+1

DO 430 I6=1,L16

I3=L16-I6+1

JMX2=JMX(I3)-1

IF (JMX2.LT.0)GO TO 430

DO 428 I2=1,JMX2

CALL UNSQZB(2,IFRB(I3),I2,IPI,IER)

DO 426 KKN=1,8

I1J2=JLRM5(I3,I2,KKN)

IF (I1J2.EQ.0)GO TO 428

IF (I1J1.NE.I1J2)GO TO 426

BM1=SPAN(I3,I2)/(12.*SCALE)

RATIO=.5

IF (MAT.EQ.2)GO TO 415

IF (DEPTH(I3,I2).EQ.0.)GO TO 415

RATIO=WIDTH(I3,I2)/DEPTH(I3,I2)

415 CONTINUE

IF (IPI.NE.1)GO TO 423

DO 422 N1=1,N16

HGT1=HGT

LX1=LX(N1)

JX1=JX(N1)

IF (IW2.EQ.2)LY1=LY(N1)

IF (IW2.EQ.2)JY1=JY(N1)

IF (IBUG.EQ.53)PRINT 858,N1,I3,I2,LX1,JX1,MKK,IW2,I13,I12

IF (MAT.EQ.2)GO TO 417

GAP1=WIDTH(I3,I2)/(24.*SCALE)+.823

417 CONTINUE

IF (LX1.NE.I3)GO TO 422

IF (JX1.NE.I2)GO TO 422

IF (IW3.EQ.2)GO TO 422

DECODE(18,333,BIMSH(I3,I2))CC

DO 418 IC=1,10

IF (CC(IC).EQ.1M)GO TO 428

418 CONTINUE

420 CONTINUE

IC1=IC-1

PLENG=IC1*HGT1

PLENG3=CORD(N1,IM2)/SCALE+PM1+.5-.5*PLENG-HGT1/3.

PLENG4=PLENG3+PLENG+2.*HGT1/3.

GO TO 432

C

C NEW BEAM IN YDIRECTION RIGHT END CCORDS.

C

422 CONTINUE

PLENG=3.*HGT1

IF(I3.GT.9)PLENG=4.*HGT1

PLENG3=CORD(N1,IM2)/SCALE+BM1+.5-.5*PLENG-HGT1/3.

PLENG4=PLENG3+PLENG+2.*HGT1/3.

GO TO 434

C

C NEW BEAM SUPPORTED ON A SECOND. BEAM

C

C XDIRECTION

C

423 CONTINUE

HGT1=.75*HGT

IF(IM3.EQ.2)GO TO 427

DECODE(10,333,BIMSH(I3,I2))CC

DO 424,IC=1,10

IF(CC(IC).EQ.1H)GO TO 425

424 CONTINUE

425 CONTINUE

IC1=IC-1

BLENG=IC1*HGT1

PLENG3=YB1(KK)+PM1+.5-.5*BLENG-HGT1/3.

PLENG4=PLENG3+BLENG+2.*HGT1/3.

GO TO 432

C

C NEW SECOND. BEAM SUPPORTED ON A SECOND BEAM

C

C YDIRECTION

C

427 CONTINUE

BLENG=3.*HGT1

IF(I3.GT.9)BLENG=4.*HGT1

PLENG3=YB1(KK)+PM1+.5-.5*BLENG-HGT1/3.

BLENG4=BLENG3+BLENG+2.*HGT1/3.

GO TO 434

426 CONTINUE

IF(MAT.EQ.2)GO TO 412

GAP1=WIDTH(I3,I2)/(24.*SCALE)+.023

412 CONTINUE

428 CONTINUE

430 CONTINUE

C

C XDIRECTION NEW BEAM RIGHT SUPP.

C

432 CONTINUE

C5=C5-GAP1

IF(RATIO.GE.5..AND.MAT.EQ.1)GO TO 439

IF(WIDTH(I3,I2).LT.2..AND.MAT.EQ.1)GO TO 438

IF(IPLOT(18,J).EQ.0.AND.IBI.EQ.1)GO TO 438

IF(IPLOT(19,J).EQ.0.AND.IBI.NE.1)GO TO 438

```
IF(C6.LT.BLENG3)GO TO 434
IF(C6.GT.BLENG4)GO TO 435
C5=C5-HGT1-.047
GO TO 438

C
C YDIRECTION NEW BEAM RIGHT SUPPORT
C
434 CONTINUE
C6=C6-GAP1
IF(IPLT(9,J).EQ.0.AND.WIDTH(I1,I2).LT.2..AND.MAT.EQ.1)GO TO 436
IF(IPLT(14,J).EQ.0.AND.IBI.EQ.0)GO TO 436
IF(IPLT(15,J).EQ.0.AND.IBI.NE.1)GO TO 438
IF(C5.LT.BLENG3)GO TO 438
IF(C5.GT.BLENG4)GO TO 439
C6=C6-HGT1-.047
438 CONTINUE
IF(IPUG.EQ.53)
1PRINT 854, KK, II, II3, II2, JMW1, IBI, IW2, INU, IJL1, IJL2,
1C1, C2, C5, C6, RM, BLENG1, BLENG2, BLENG3, BLENG4
854 FORMAT(* IO 3*, IO14, 9F8.2)
MKB1(INU)=IJL+IW3*10000
IF(MAT.EQ.2)GO TO 450
C11(1)=C1-GAP2
C11(2)=C5-GAP2
C22(1)=C2
C22(2)=C6
IF(IPUG.EQ.53)WRITE(6,*(=IOSECB3*, 6I5, 2F10.3)*)
1IW3, MKK1, LI1, IS1, II3, II2, GAP1*120./UZ10, GAF2*120./UZ10
IF(IW3.EQ.2)GO TO 442
C11(1)=C1
C11(2)=C5
C22(1)=C2-GAP2
C22(2)=C6-GAP2
442 CONTINUE
CALL DASH(C11, C22, 2, .1, .1)
IF(IW3.EQ.1)GO TO 446
C11(1)=C1+GAP2
C11(2)=C5+GAP2
GO TO 448
446 CONTINUE
C22(1)=C2+GAP2
C22(2)=C6+GAP2
448 CONTINUE
CALL DASH(C11, C22, 2, .1, .1)
GO TO 452
450 CONTINUE
CALL PLOT(C1, C2, 3)
CALL PLOT(C5, C6, 2)
452 CONTINUE
CALL S07BITS(2, IERR(11), IS1, 2, IER)
BIN(INU)=C6-C2
OB2(INU)=C2
IF(IW3.EQ.2)GO TO 440
OB2(INU)=C1
BIN(INU)=C5-C1
440 CONTINUE
460 CONTINUE
```

```

490 CONTINUE
500 CONTINUE
    DO 510 I3=1,LIG
        JMX1=JMX(I3)-1
        DO 510 I2=1,JMX1
            CALL UNSQZR(2,IFBR(I3),I2,YPI,ICR)
            IF(YPI.EQ.0)GO TO 520
510 CONTINUE
        GO TO 530
520 CONTINUE
530 CONTINUE
    INUMX=INU
    GO TO 700
400 CONTINUE

C
    CALL NEWPEN(1)
C SECONDARY BEAM MARKS
C
    HGT1=.75*HGT
    DO 640 IMX=1,INUMX
        IFBR=0
        MKK=MOD(MKB1(IMX),10000)
        IW2=MKB1(IMX)/10000
        LI1=MKK/100
        IS1=MKK-LI1*100
        IF(IMX.EQ.INUMX)GO TO 605
        MKK1=MOD(MKB1(IMX+1),10000)
        IW3=MKB1(IMX+1)/10000
        LI2=MKK1/100
        IS2=MKK1-LI2*100
605 CONTINUE
        SLENG1=3.*HGT1
        IF(LI1.GT.9)SLENG1=4.*HGT1
        GAP1=0.
        IF(MAT.EQ.1)GAP1=WIDTH(LI1,IS1)/(24.*SCALE)
        POS1=RIM(IMX)*.5
        C3=OP2(IMX)+POS1
        C4=YB1(IMX)-GAP1
        IF(IMX.EQ.INUMX)GO TO 608
        C5=YB1(IMX+1)-YB1(IMX)-2.*(HGT1+.047+GAP1)
608 CONTINUE
        IF(IW2.EQ.1)GO TO 620
        C3=XB1(IMX)+GAP1
        C4=OB2(IMX)+POS1
        IF(IMX.EQ.INUMX)GO TO 625
        C5=XB1(IMX+1)-XB1(IMX)-2.*(HGT1+.047+GAP1)
620 CONTINUE
        IF(IPUG.EQ.53)WRITE(6,*(+I0SEB4+*,6I4,6F10.3)*
        *MKK,IW2,LI1,IS1,IMX,INUMX,XB1(IMX),XB1(IMX+1),YB1(IMX)
        *,YB1(IMX+1),GAP1,C5
        GAP=HGT1+.047+GAP1
        IF(IMX.EQ.INUMX)GO TO 625
        IF(ABS(OB2(IMX+1)-OB2(IMX)).GT.GAP)GO TO 625
        IF(C5.LE.0..AND.IW3.EQ.IW2)GO TO 640
625 CONTINUE
        IF(SLENG1.GT.RIM(IMX))IFBR=2
        IF(.75*SLENG1.GT.BIP(IMX))IFBR=3

```



```

      IF (IPUG.EQ.53) PRINT 997,IMX,MKK,IM2,IFBM,C3,C4,XR1(IMX)
      1,OR2(IMX),RIM(IMX),PCS1,SLENG1
      997 FORMAT(' SECBM',4I6,RF8.2)
      IF (IFBM.NE.3) GO TO 630
      C3=XR1(IMX)+GAP1
      C4=OR2(IMX)+RIM(IMX)+.5
      IF (IM2.EQ.2) GO TO 630
      C4=XR1(IMX)-GAP1
      C3=OR2(IMX)+RIM(IMX)+.5
      630 CONTINUE
      CALL LABEL515,C3,C4,KA,KP,LVM4,IM2,MKK,SCALE)
      640 CONTINUE
      700 CONTINUE
      CALL NEWPEA(1)
      RETURN
      END

```

FILE MAP--(LC=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
0B	/ALFA/		REAL		DBM	3221B	
3222B		*S*	REAL		DRM1	3213B	
3223B		*S*	REAL		DEPTH	1130B	/BMFL1/
3224B		*S*	REAL		DUM	0B	/THETFL/
5050B	/BMFL4/	EQV	REAL	240	DUMMY	5050B	/BMFL4/
1775B	/THETFL/	EQV	REAL	693	DV	46412B	/PLOT1/
3650B	/PLOT1/		REAL	1960	EC1	115B	//
3241B			REAL		EC5	116B	//
3237B			REAL		FACTOR	3162B	
3240B			REAL		FCL	1060B	/CLFL2/
3262B			REAL		GAP	3320B	
3263B			REAL		GAP1	3161B	
3225B			REAL		GAP2	3227B	
1750B	/THETFL/	EQV	REAL	21	GRID1	1305B	/BMFL4/
3226B			REAL		HGT	3163B	
642B	//		REAL	20	HGT1	3257B	
0B	/PLOT1/		REAL	1960	I	3177B	
3146B			REAL	10	IRI	3251B	
27010B	/PLOT1/		REAL	1984	IRI1	3275B	
2367B	/BMFL4/		REAL	90	IBR	17240B	/PLOT1/
423B	//		REAL	2	IBUG	2521B	/BMFL4/
448B	/BMFL4/		INTEGER	280	IC	3230B	
10B	/BMFL4/		REAL	28B	ICFL	1070B	/BMFL4/
1B	/ALFA/		REAL		ICRIC	103B	//
23110B	/PLOT1/		REAL	1984	ICRIS	75B	//
32710B	/PLOT1/		REAL	1984	IC1	3231B	
36610B	/PLOT1/		REAL	1984	IDAT	74B	//
3233B			REAL		IDOC	60B	//
52576B	/PLOT1/		REAL	2	IDQS	64B	//
3234B			REAL		ID3	2B	/ALFA/
52600B	/PLOT1/		REAL	2	IER	3252B	
3316B			REAL		IFBB	0B	/THETFL/
3317B			REAL		IFBM	1304B	/BMFL4/
3235B			REAL		IFFRM	12B	//
3236B			REAL		IFLAG	2740B	/BMFL4/
55B	//		REAL	3	IFLAG1	2524B	/BMFL4/

```

SUBROUTINE SIZE5 (IS, J, IW, C1V, C2V, LVNC, SCALE)
COMMON MAT, MEM, LEV, ISOLV, LVFIN, LVFIN, LVMAX, LVMAX, NL, IFLEV, IFFRM
COMMON ZZZZ, NMF, DAT(3), IDOC(6), IDOS(6), IDAT, ICRIS(6), ICRIC(6)
COMMON JSTOP, NSTOP, IPRINT, ILINK, EC1, EC5, MAXFR, ITEST, IRUN, LVU, LNU
COMMON KAK, LIST(10), IOF, CODE(2), MD(25), UF(20)
COMMON ITC1, AEZ(90)
COMMON USE(10), RON(20), LON(20)
COMMON TAKI(10), JET(10), KET(20), LET(20)
INTEGER KAK(90,2), NME(11,3)
COMMON/PLT1/ PSTIF(980,2), PLEN(980,2), MKP(990,2), KST(980,2),
1 IPR(980,2), CSTIF(992,2), CLEAI(992,2), CX(992,2), CZ(992,2),
2 INWAY(2), PHIC(992,2), DV(992,2), TRANS(90,2), C11(2), C22(2),
3 JRR(980,2)
COMMON/ALFA/ALFA, COSA, ID3
COMMON/DAIND2/MSOP2, MIND2(91)
COMMON /DADAT2/LIND2(171), CJP(18), CLDAT(10,6)
COMMON/SPLICE/NSPL(140)
COMMON/RMFL2/SPAN(30,20), XBM(30,20,4), LRM(30,20,4), JMX(30)
COMMON/CLFL2/LY(140), JX(140), LY(140), JY(140), FCL(140,9)
COMMON/RMFL4/STPX, STPY, STPY1, MB1(2), IFR2(2), CORD(140,2)
1, COL1, ICFL(140), IFRM, GRID1(140,2), NCOL(2), MKC(140,2)
4, CLEV(90), IBUG, STZZ1, STZZ, IFLAG1(140), IFLAG(140)
5, KKPLT(140), SIDE(10), MARG, IPLOT(25,15), JPL, IGRID1(140)
6, IGRID(140,2), DUMMY(240)
COMMON/RMFL1/WIDTH(30,20), DEPTH(30,20)
REAL MARG(10)
INTEGER COL1(140,2)
DIMENSION CC(10), LEVE(21)
COMMON/THETFL/DUM(1890)
DIMENSION IFRB(250), XB1(250), YB1(250), MKB1(250), BMSH(21)
1, BMSH(30,20), OB2(250), BIM(240)
EQUIVALENCE (DUM(1), IFRB(1)), (DUM(251), XB1(1)), (DUM(501), YB1(1))
EQUIVALENCE (DUM(751), MKB1(1))
EQUIVALENCE (DUM(1001), BMSH(1)), (DUM(1022), BIMSH(1,1))
EQUIVALENCE (DUM(1622), OP2(1)), (DUM(1872), INUMX)
EQUIVALENCE (DUMMY(1), BIM(1))
IFRM=0
FACTCR=.75
HGT=.125*FACTOR
IF (IS.EQ.2) 60 TO 150
IF (IS.EQ.3) 60 TO 500
C
C MAIN BEAM SHAPES
C
RM=1./(BLEM(LVNC, IW)*SCALE)
IF (IW.EQ.1) MARK=MOD(MKB(LVNC, IW), 10000)
IF (IW.EQ.2) MARK=MKB(LVNC, IW)/10000
LI1=MARK/100
IS1=MARK-LI1*100
WIDT1=0.
IF (MAT.EQ.1) WIDT1=WIDTH(LI1, IS1)/(24.*SCALE)
DECODE(10, 333, BIMSH(LI1, IS1)) CC
DO 100 IC=1, 10
IF (CC+IC).EQ.14 160 TO 110
100 CONTINUE
110 CONTINUE
IC1=IC-1

```

```

      SLENG=IC1*HGT
      IF (SLENG.GT.RM) IFRM=2
      IF (.75*SLENG.GT.RM) IFRM=3
      IF (IFRM.EQ.3) GO TO 140
      C1=C1W+SLENG*.5*ALFA-.047*CO SA
      C2=C2W-.047*ALFA-SLENG*.5*CO SA
      WIDT2=WIDT1*25.4*120.
      IF (IPUG.EQ.53) WRITE(6,*(I,SIZE51*,4I5,5F10.3)*)
      *MARK,IW,LI1,IS1,WIDT2,ALFA,CO SA,C1W,C2W
      IF (IW.EQ.2.AND.IS.EQ.1) GO TO 120
      C1=C1W-.047*ALFA-SLENG*.5*CO SA
      C2=C2W-SLENG*.5*ALFA+.047*CO SA
120  CONTINUE
      HGT1=HGT
      IF (IFRM.EQ.2) HGT1=.75*HGT
      ANGLE=0.
      IF (IS.EQ.4) GO TO 130
      ANGLE=ASIN(ALFA)*57.29578
      IF (IW.EQ.2) ANGLE=ANGLE+90.
130  CONTINUE
      CC1=BIMSH(LI1,IS1)
      CALL SYMBOL(C1,C2,HGT1,CC1,ANGLE,IC1)
      GO TO 470
140  CONTINUE
      C1=C1W+.5*HGT
      C2=C2W
      IF (IW.EQ.2) GO TO 145
      C1=C1W
      C2=C2W+.5*HGT1
145  CONTINUE
      CALL SYMBOL(C1,C2,HGT,***,ANGLE,1)
      GO TO 470
333  FORMAT(10A1)
C
C   SECONDARY BEAM SHAPES
C
150  CONTINUE
      DO 460 IMX=1,INUMX
      IFRM=0
      MKK=MOD(MKB1(IMX),10000)
      IW2=MKB1(IMX)/10000
      LI1=MKK/100
      IS1=MKK-LI1*100
      WIDT1=0.
      IF (MAT.EQ.1) WIDT1=WIDTM(LI1,IS1)/(24.*SCALE)
      DECODE(10,333,BIMSH(LI1,IS1))CC
      DO 170 IC=1,10
      IF (CC(IC).EQ.1W) GO TO 140
170  CONTINUE
180  CONTINUE
      IC1=IC-1
      HGT1=.75*HGT
      SLENG1=IC1*HGT1
      IF (SLENG1.GT.BIM(IMX)) IFRM=2
      IF (.75*SLENG1.GT.BIM(IMX)) IFRM=3
      POS1=(BIM(IMX)-SLENG1)*.45
      POS2=.047

```

```

      C3=OP2(IMX)+POS1
      C4=YF1(IMX)+POS2
      C4=C4+WIDTH1
      ANGLE=0.
      IF(IM2.EQ.1)GO TO 190
      C3=XP1(IMX)-POS2
      C4=OP2(IMX)+POS1
      C3=C3-WIDTH1
      ANGLE=90.
190  CONTINUE
      IF(IFRM.EQ.3)GO TO 210
      IF(IFBM.EQ.2)HGT1=.75+HGT1
      CC1=PISSH(LI1,IS1)
      CALL SYMBOL(C3,C4,HGT1,CC1,ANGLE,IC1)
      GO TO 450
210  CONTINUE
      C3=XP1(IMX)-.047
      C4=OP2(IMX)+.5*PIIM(IMX)-.5+HGT1
      IF(IM2.EQ.2)GO TO 220
      C3=OP2(IMX)+PIIM(IMX)+.5-.5+HGT1
      C4=YF1(IMX)+.047
220  CONTINUE
      CALL SYMBOL(C3,C4,HGT1,*,*,ANGLE,1)
450  CONTINUE
460  CONTINUE
470  CONTINUE
      GO TO 1000
500  CONTINUE
C
C   COLUMN PROFILES
C
      LDASD=0
      KA=C1M
      KB=C2M
      LVM4=LVNC
C
C   FIND HIGHEST LEVEL PER COLUMN STACK
C
      N=0
      DO 520 K=KA,KB
      N=N+1
      DO 510 L=2,LVM4
      L1=LVM4-L+2
      LVN=(K-1)*LVM4+L1
      LEVE(N)=L1
      IF(ICX(LVN,IM).EQ.999..AND.CZ(LVN,IM).EQ.999.)GO TO 510
      GO TO 520
510  CONTINUE
520  CONTINUE
      DO 590 L=2,LVM4
      L1=LVM4-L+2
      N=0
      DO 580 K=KA,KB
      N=N+1
      LVN=(K-1)*LVM4+L1
      MK=MKC(K,IM)
      CJP(2)=6M

```

```

      IF(L1.GT.LEVE(N))GO TO 540
      IF(L1.EQ.LDASD)GO TO 530
      CALL STINDX(12,MIND2,91,0)
      IF(MIND2(L1).LE.0) GO TO 532
      CALL READMS(12,LIND2,171,L1-1)
      CALL STINDX(12,LIND2,171,0)
      IF(LIND2(MK+1).LE.0) GO TO 532
      LDASD=L1
530 CONTINUE
      CALL READMS(12,CJP,2,MK+30)
C
532 CONTINUE
      IF(L1.EQ.LEVE(N))GO TO 540
      IF(NSPL(MK).EQ.0)GO TO 540
      IF(NSPL(MK).GT.0)GO TO 535
      NSPL1=IAPS(NSPL(MK))
      MULT=MOD(L1-1,NSPL1)
      IF(MULT.GT.0)GO TO 540
      GO TO 540
535 CONTINUE
      CALL UNSQZB(1,NSPL(MK),L1-1,IRI,IER)
      IF(IRI.EQ.0)GO TO 540
540 CONTINUE
      IF(IPUG.EQ.53)PRINT 999,MK,L1-1,IRI,NSPL(MK),CJP(2)
999 FORMAT(' SIZE5-1',3I8,120,1X,A10)
      COL=0.
      C1=CX(LVN,IV)/SCALE
      C2=CZ(LVN,IV)/SCALE
      IF(CLENI(LVN,IV).EQ.0.)GO TO 540
      COL=1./CLENI(LVN,IV)
      IF(CSTIF(LVN,IV).LE.1.E-20.OR.COL.EQ..9)GO TO 540
      IF(LEVE(N).EQ.L1)GO TO 550
      C1X=C1-.047
      C17=C2+2./SCALE
      CALL PLOT(C1X,C17,3)
      C2X=C1+.047
      CALL PLOT(C2X,C17,2)
550 CONTINUE
      DECODE(10,333,CJP(2))CC
      DO 555 IC=1,10
      IF(CC(IC).EQ.1)GO TO 560
555 CONTINUE
560 CONTINUE
      IC1=IC-1
      ANGLE=90.
      CLENG=IC1*HGT
      DIMEN=ABS(CZ(LVN,IV)-CZ(LVN-1,IV))/SCALE
      IF(CLENG.GE.DIMEN)CLENG=IC1*.75*HGT
      CC1=CJP(2)
      C1X=C1-.047
      C17=C2-COL/(2.*SCALE)-CLENG*.5
      IF(CLENG.GT.DIMEN)GO TO 570
      CALL SYMBOL(C1X,C17,HGT,CC1,ANGLE,IC1)
      GO TO 580
570 CONTINUE
      CALL SYMBOL(C1X,C17,HGT,***,ANGLE,1)
580 CONTINUE

```

590 CONTINUE

C

C

1000 CONTINUE

C

RETURN

END

LE MAP--(LO=A)

-ADDRESS--BLOCK----		PROPERTIES-----	TYPE-----	SIZE	-NAME---ADDRESS--BLOCK---		
0B	/ALFA/		REAL		GRID1	1305P	/BMFL4/
1071P			REAL		HGT	1054P	
5050B	/BMFL4/	EQV	REAL	240	HGT1	1070P	
1775B	/THETFL/	EQV	REAL	600	IP1	1122B	
3650P	/PLOT1/		REAL	1960	IBR	17240B	/PLOT1/
1055P			REAL		IEUG	2521P	/BMFL4/
1750B	/THETFL/	EQV	REAL	21	IC	1062B	
642B	//		REAL	20	ICFL	1070B	/BMFL4/
0B	/PLOT1/		REAL	1960	ICRIC	103B	//
1014B			REAL	10	ICRIS	75B	//
1072B			REAL		IC1	1063B	
253B	/DADAT2/		REAL	1B	ICAT	74B	//
275P	/DADAT2/		REAL	60	IDOC	60B	//
1130P			REAL		IDOS	66B	//
27010P	/PLOT1/		REAL	1984	ID3	2B	/ALFA/
2367P	/BMFL4/		REAL	90	IER	1123B	
423B	//		REAL	2	IFBB	0B	/THETFL
1124P			REAL		IFBM	1304B	/BMFL4/
440B	/BMFL4/		INTEGER	2B0	IFFRM	12B	//
10B	/BMFL4/		REAL	2B0	IFLAG	2740B	/BMFL4/
1B	/ALFA/		REAL		IFLAG1	2524B	/BMFL4/
23110B	/PLOT1/		REAL	1984	IFLEV	11B	//
32710B	/PLOT1/		REAL	1984	IFR2	6B	/BMFL4/
36610B	/PLOT1/		REAL	1984	IGRID	4420B	/BMFL4/
1065B			REAL		IGRID1	4204B	/BMFL4/
4	DUMMY-ARG		REAL		ILINK	114P	//
1125P			REAL		IMX	1073B	
1126P			REAL		INUMX	3517P	/THETFL
52576P	/PLOT1/		REAL	2	INWAY	42510B	/PLOT1/
1066P			REAL		IO9	422P	//
5	DUMMY-ARG		REAL		IPLOT	3414P	/BMFL4/
1127B			REAL		IPRINT	113B	//
52600P	/PLOT1/		REAL	2	IRUN	121B	//
1102B			REAL		IS	1	DUMMY-A
1103B			REAL		ISOLV	3B	//
55B	//		REAL	3	IS1	1060B	
1130B	/BMFL1/		REAL	600	ITC1	475B	//
1131B			REAL		ITEST	120B	//
0B	/THETFL/	EQV	REAL	1890	IW	3	DUMMY-A
5050B	/BMFL4/	EQV	REAL	240	IW2	1076B	
46412B	/PLOT1/		REAL	1984	J	2	DUMMY-A
115B	//		REAL		JBR	52602B	/PLOT1/
116B	//		REAL		JET	724B	//
1053B			REAL		JMX	12430B	/BMFL2/
1060B	/CLFL2/		REAL	1260	JPL	4203B	/BMFL4/

SUBROUTINE UNP53(I,A1,A2,P)
COMMON/ALFA/SINA,COSA,ID3

SINA=0.

COSA=1.

C AAARBBCCC

I1=I/1000

I2=I1*1000

R=(I-I2)*.01

I3=I1*1000

I4=I3*1000

A2=(I1-I4)*.01

I5=I3/1000

I6=I5*1000

A1=(I3-I6)*.01

IF(I5.EQ.0)GO TO 10

SINA=-((I5-10000)*.0001

IF(I5.LT.10000)SINA=I5*.0001

Y=ASIN(SINA)

COSA=COS(Y)

10 CONTINUE

RETURN

END

FILE MAP--(LO=A)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
2	DUMMY-ARG		REAL	I2		45B	
3	DUMMY-ARG		REAL	I3		46B	
4	DUMMY-ARG		REAL	I4		47B	
1B	/ALFA/		REAL	I5		50B	
1	DUMMY-ARG		INTEGER	I6		51B	
2B	/ALFA/		INTEGER	SINA		0B	/ALFA/
44B			INTEGER	Y		52B	

URES--(LO=A)

-----TYPE-----ARGS-----CLASS-----

GENERIC	1	INTRINSIC
GENERIC	1	INTRINSIC

IENT LABELS--(LO=A)

ADDRESS-----PROPERTIES-----DEF

37B

20

POINTS--(LO=A)

ADDRESS--ARGS---

3B

```

SUBROUTINE SYMP5( IEND, K1, SCALE, Y10, Y10, K, IV)
COMMON MAT, MEM, LEV, ISOLV, LVFIN, LVFIN, LVMAX, LVMAX, NL, IFLEV, IFFRM
COMMON ZZZZ, NME, DAT(3), IDOC(6), IDCS(6), IDAT, ICRIS(6), ICRIC(6)
COMMON JSTOP, NSTOP, IPRINT, ILINK, EC1, EC5, MAXFR, ITEST, IRUN, LVU, LNU
COMMON KAK, LIST(10), TOR, CONF(2), MO(20), UF(20)
COMMON ITC1, NEZ(90)
COMMON USE(10), BON(20), LON(20)
COMMON TAKI(10), JET(10), KET(20), LET(20)
INTEGER KAK(90,2), NME(11,3)

```

```

COMMON/ PLOT1/ PSTIF(980,2), PLEN(980,2), MVE(980,2), KST(980,2),
1IPR(990,2), CSTIF(992,2), CLENI(992,2), CX(992,2), CZ(992,2),
2INWAY(2), PHIC(992,2), DV(992,2), TRANS(90,2), C11(2), C22(2),
3JBR(990,2)

```

```

COMMON/ BMFL4/ STPX, STPX1, STPY, STPY1, MB1(2), IFR2(2), CORD(140,2), COL1
1, ICFL(140), IFPM, GRID1(140,2), NCCL(2), MKC(140,2)
4, CLEV(9), IPUE, STZZ1, STZZ, IFLAG1(140), IFLAG(140)
5, KKPLT(140), SIDE(10), MARG, PLOT(25,15), JPL, IGRID1(140)
6, IGRID(140,2)
REAL MARG(10)

```

```

INTEGER COL1(140,2)
COMMON/ ALFA/ SINA, COSA, ID3
COMMON /BMFL1/ WIDTH(30,20), DEPTH(30,20)
COMMON /BMFL2/ SPAN(30,20), XEM(30,20,4), LBM(30,20,4), JMX(30)
COMMON /BMFL3/ EYE(30,20), ILINE(30)
COMMON/ PMLN1/ FC1, FY1

```

```

COMMON /RIGI/ IGI(30,20)
COMMON /BRACE/ IBRA(30,20), LAR(5)
COMMON/ CLFL1/ FC5, FY5, CVR5, FKC(9)
COMMON/ CLFL2/ LX(140), JX(140), LY(140), JY(140), FCL(140,9)
COMMON/ CLFL3/ X(140), Y(140), Z(140), XMAX(140), YMAX(140), KSC(140)
*, KDC(140)

```

```

COMMON/ CLFL6/ NOX, NEO(140), MAK(140)
COMMON/ CLFL5/ KBAR(140,2)
COMMON /CLFL7/ MCD(140,3), XY(9), STUB(9), IFO(9), XK(9), YK(9), INX(9)
COMMON/ CLFL8/ TOR(140,2), TCRM(21)
COMMON/ RIBS/ RIR9, RIR(8)
COMMON/ MOVE/ MOVNO, XM(21,10), FM(21,10), MSTART(10,10),

```

```

MEVD(10,10), IREVE(10), NWH(10), PACT(10), MIT(10)
COMMON/ ALS/ ALFC(4), ALFY(4)
COMMON/ BRCS/ IBRC(10), EBRC(10)
COMMON/ GEN1/ ESTI(12,4), JYPE(30,2), KANT(30,2)
COMMON/ CPR0S/ CPR0(140)
COMMON/ CONSTS/ CONSTS(10), IEOF, ACC(3)
COMMON/ ALFAC/ ALFAC(140)
DATA XSIZE/16., YSIZE/24./
IF( IEND.EQ.3) GO TO 50

```

```

C
C IEND =1 LEFT END CONNECTION
C 2 RIGHT END CONNECTION
C 3 TO DRAW A COLUMN
C 4 COLUMN ORIENTATION IN FLOOR PLAN( BETA 90 OR 0)
C

```

```

HGT=YSIZE/(SCALE*100.)
HEI=.125
RADE=.063
ANGLE1=90.

```


ANGLE2=270.

ANGLE3=-90.

C1=X10+HEI/3.5

C3=Y10

IF(IEND.EQ.4)C1=X10+HEI*COSA+COSA/3.5

IF(IEND.EQ.4)C3=Y10+HEI*COSA+SINA/3.5

C2=Y10+RAD

IF(IV.EQ.2.AND.IEND.EQ.4)C1=X10-HEI*COSA*SINA/3.5

IF(IV.EQ.2.AND.IEND.EQ.4)C3=Y10+HEI*COSA*COSA/3.5

ALFA=ASIN(SINA)*57.29578

IF(IV.EQ.1.AND.IEND.EQ.4)ANGLE3=-90.+ALFA

IF(IV.EQ.2.AND.IEND.EQ.4)ANGLE3=ALFA

IF(IEND.EQ.4)GO TO 20

IF(IEND.EQ.1)GO TO 10

ANGLE1=90.

ANGLE2=-90.

ANGLE3=90.

IF(IV.EQ.1.AND.IEND.EQ.5)ANGLE3=90+ALFA

IF(IV.EQ.2.AND.IEND.EQ.5)ANGLE3=-190.+ALFA

C1=X10-HEI/3.5

C3=Y10

IF(IEND.EQ.5)C1=X10-HEI*COSA*COSA/3.5

IF(IEND.EQ.5)C3=Y10-HEI*COSA*SINA/3.5

C2=X10-RAD

IF(IV.EQ.2.AND.IEND.EQ.5)C1=X10+HEI*COSA*SINA/3.5

IF(IV.EQ.2.AND.IEND.EQ.5)C3=Y10-HEI*COSA*COSA/3.5

10 CONTINUE

IF(K1.EQ.1)GO TO 20

C

C K1=0 HINGED CONNECTION *****

CALL ARC(C2,Y10,RAD,ANGLE1,ANGLE2)

GO TO 30

20 CONTINUE

C

C K1=1 RIGID CONNECTION

C

FACTOR=.60

HEI=FACTOR*.125

CALL SYMBOL(C1,C3,HEI,2,ANGLE3,-1)

HEI=.5*HEI

CALL SYMBOL(C1,C3,HEI,2,ANGLE3,-1)

30 CONTINUE

RETURN

50 CONTINUE

C

C KSC(MK)=1 SO X(MK)

C KSC(MK)=2 ROUND X(MK)

C 3 RECT X(MK) Y(MK)

C 9 DUMMY DRAW AN X

C 10 PILASTER

C 11 BETA 0 WEB PARALLEL TO X FCL(MK,4) FCL(MK,5)

C 12 BETA 90 WEB PERPENDICULAR TO X FCL(MK,5) FCL(MK,4)

C

C FACT =1.5 MAGNIFICATION FACTOR

C

FACT=1.5

IF(MAT.EQ.1)FACT=1.

```

      MK=MKC(K,IW)
      ALC=ALFAC(MK)
      COSAL=COS(ALC)
      SINAL=SIN(ALC)
      IF (IPUG.EQ.53) PRINT 999,MK,IFLAG(MK),KSC(MK)
090  FORMAT(' ** DEBUG SYMBF ***.416)
      IF (IFLAG(MK).NE.0) GO TO 100
      IF (KSC(MK).EQ.0) GO TO 100
      KMK=KSC(MK)
      IF (KMK.EQ.1) GO TO 90
      IF (KMK.EQ.2) GO TO 85
      IF (KMK.EQ.3) GO TO 90
      IF (KSC(MK).EQ.9) GO TO 70
      IF (KMK.EQ.10) GO TO 75
      IF (KSC(MK).EQ.11) GO TO 60
C
C   KSC(MK) EQ 12
C
      FLAN=FACT*FCL(MK,5)/(12.*SCALE)
      DEP=FACT*FCL(MK,4)/(12.*SCALE)
      X1=X10+DEP*.5*SINAL-FLAN*.5*COSAL
      Y1=Y10-DEP*.5*COSAL-FLAN*.5*SINAL
      X2=X1+FLAN*COSAL
      Y2=Y1+FLAN*SINAL
      X4=X10-DEP*.5*SINAL+FLAN*.5*COSAL
      Y4=Y10+DEP*.5*COSAL+FLAN*.5*SINAL
      X3=X4-FLAN*COSAL
      Y3=Y4-FLAN*SINAL
      X5=X10+DEP*.5*SINAL
      Y5=Y10-DEP*.5*COSAL
      X6=X10-DEP*.5*SINAL
      Y6=Y10+DEP*.5*COSAL
      CALL PLOT(X1,Y1,3)
      CALL PLOT(X2,Y2,2)
      CALL PLOT(X5,Y5,3)
      CALL PLOT(X6,Y6,2)
      CALL PLOT(X3,Y3,3)
      CALL PLOT(X4,Y4,2)
      IF (IPUG.EQ.53.AND.MK.EQ.1)
1  PRINT 998,MK,KSC(MK),X10,Y10,DEP,FLAN,COSAL,SINAL,ALFAC(MK),
2  X1,Y1,X2,Y2
998  FORMAT(' **DEBUG SYMBF***.213,11F6,2)
      IFLAG(MK)=1
      GO TO 100
C
C   KSC(MK) EQ 11
C
      60  CONTINUE
      DEP=FACT*FCL(MK,4)/(12.*SCALE)
      FLAN=FACT*FCL(MK,5)/(12.*SCALE)
      X1=X10-DEP*.5*COSAL-FLAN*.5*SINAL
      Y1=Y10-DEP*.5*SINAL+FLAN*.5*COSAL
      X2=X1+FLAN*SINAL
      Y2=Y1-FLAN*COSAL
      X4=X10+DEP*.5*COSAL+FLAN*.5*SINAL
      Y4=Y10+DEP*.5*SINAL-FLAN*.5*COSAL
      X3=X4-FLAN*SINAL
      Y3=Y4+FLAN*COSAL

```

```

X5=X10-DEP*.5*COSAL
Y5=Y10-DEP*.5*SINAL
X6=X10+DEP*.5*COSAL
Y6=Y10+DEP*.5*SINAL
CALL PLOT(X1,Y1,3)
CALL PLOT(X2,Y2,2)
CALL PLOT(X5,Y5,3)
CALL PLOT(X6,Y6,2)
CALL PLOT(X3,Y3,3)
CALL PLOT(X4,Y4,2)
IF (IPUE.EQ.5).AND.(MK.EQ.6)
  PRINT 998,MK,KSC(MK),X10,Y10,DEP,FLAN,COSAL,SINAL,ALFAC(MK),
  2X1,Y1,X2,Y2
  IFLAG(MK)=1
  GO TO 100

```

```

C
C DUMMY
C

```

```

70 CONTINUE
HGT=YSIZE/(100.*SCALE)
CALL SYMBOL(C1,C2,4*HGT,2,0,-1)
IFLAG(MK)=1
GO TO 100
75 CONTINUE

```

```

C
C PILASTERS
C

```

```

IF (Y(MK).EQ.0) GO TO 85
GO TO 90

```

```

C
C ROUND CONCRETE COLUMN X(MK)
C

```

```

85 CONTINUE
RAD=FACT*X(MK)/(24.*SCALE)
CALL ARC(X10,Y10,RAD,0.,360.)
IFLAG(MK)=1
GO TO 100

```

```

C
C RECT. CONCRETE COLUMN X(MK) Y(MK)
C SO CONCRETE COLUMN X(MK) Y(MK)
C

```

```

90 CONTINUE
FLAN=FACT*Y(MK)/(12.*SCALE)
DEP=FACT*X(MK)/(12.*SCALE)
X1=X10-DEP*.5*COSAL-FLAN*.5*SINAL
Y1=Y10-DEP*.5*SINAL+FLAN*.5*COSAL
X2=X1+FLAN*SINAL
Y2=Y1-FLAN*COSAL
X4=X10+DEP*.5*COSAL+FLAN*.5*SINAL
Y4=Y10+DEP*.5*SINAL-FLAN*.5*COSAL
X3=X4-FLAN*SINAL
Y3=Y4+FLAN*COSAL
X5=X10-DEP*.5*COSAL
Y5=Y10-DEP*.5*SINAL
X6=X10+DEP*.5*COSAL
Y6=Y10+DEP*.5*SINAL
CALL PLOT(X1,Y1,3)

```

CALL PLOT(X2,Y2,2)

CALL PLOT(X4,Y4,2)

CALL PLOT(X3,Y3,2)

CALL PLOT(Y1,Y1,2)

IFLAG(MV)=1

GO TO 100

C
C DONE RETURNC
100 CONTINUE
END

LE MAP--(LOEA)

ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK
13E	/CONSTS/		REAL	3	FCL	1060P	/CLFL2/
726P			REAL		FC1	0P	/BMLA1/
722P			REAL		FC5	0P	/CLFL1/
09	/ALFAC/		REAL	140	FKC	3B	/CLFL1/
0P	/ALS/		REAL	4	FLAN	732P	
4P	/ALS/		REAL	4	FY1	1P	/BMLN1/
714P			REAL		FY5	1P	/CLFL1/
715P			REAL		GRID1	1305P	/BMFL4/
716P			REAL		HEI	712P	
3650B	/PLOT1/		REAL	1960	HGT	711P	
642B	//		REAL	20	IBR	17240P	/PLOT1/
0P	/PLOT1/		REAL	1960	IBRA	0P	/BRACE/
27010P	/PLOT1/		REAL	1984	IBRC	0P	/BRCS/
2367B	/BMFL4/		REAL	99	IBUG	2521P	/BMFL4/
423P	//		REAL	2	ICFL	1070B	/BMFL4/
440B	/BMFL4/		INTEGER	280	ICRIC	103P	//
0P	/CONSTS/		REAL	10	ICRIS	75P	//
10P	/BMFL4/		REAL	280	IDAT	74P	//
1P	/ALFA/		REAL		IDOC	60B	//
727B			REAL		IDOS	66B	//
0P	/CPPOS/		REAL	140	ID3	2P	/ALFA/
23110P	/PLOT1/		REAL	1984	IEND	1	DUMMY-AR
2P	/CLFL1/		REAL		IEOF	12B	/CONSTS/
32710B	/PLOT1/		REAL	1984	IFRM	1304P	/BMFL4/
36610B	/PLOT1/		REAL	1984	IFFRM	12B	//
717B			REAL		IFLAF	2740B	/BMFL4/
52576B	/PLOT1/		REAL	2	IFLAG1	2524B	/BMFL4/
721B			REAL		IFLEV	11B	//
52600B	/PLOT1/		REAL	2	IFO	666B	/CLFL7/
720B			REAL		IFR2	6B	/BMFL4/
55B	//		REAL	3	IGI	0B	/RIGI/
733B			REAL		IGRID	4420B	/BMFL4/
1130B	/BMFL1/		REAL	600	IGRID1	4204B	/BMFL4/
46412B	/PLOT1/		REAL	1984	ILINE	1130B	/BMFL3/
12B	/BRCS/		REAL	10	ILINK	114B	//
115B	//		REAL		INWAY	42510P	/PLOT1/
116B	//		REAL		INX	721B	/CLFL7/
0P	/GEN1/		REAL	4P	IO9	422B	//
0P	/BMFL3/		REAL	600	IPL0T	3414B	/BMFL4/
724B			REAL		IPRINT	113B	//
723B			REAL		IREVE	1155B	/MOVE/

END

1 2 0 6 8 6

FIN