

ARCHGRAF.2 : A REVISION OF ARCHGRAF
AN ARCHITECTURAL GRAPHICS PROGRAM

by

GARY WAYNE LAW

B. of Arch., Texas Tech University, 1973

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

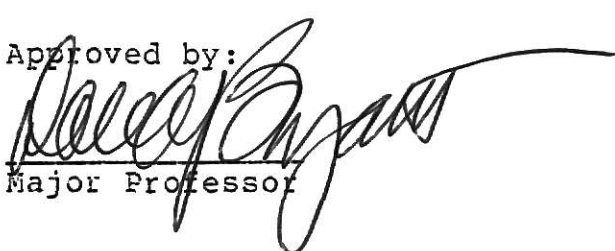
MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

Approved by:


Major Professor

SPEC
COLL
LD
2668
.R4
1981
L38
c.2

A11200 067370

TABLE OF CONTENTS

I. INTRODUCTION

Overview.....	1
Motivation.....	2
Background.....	4
Reorganization.....	9
Revision Considerations.....	11

II. REVISIONS

Chromatics Logon Procedures.....	13
Editing and Compiling.....	13
Armod Exec.....	14
Arch2 Exec.....	15
Rotation Drawing Option.....	17
Solar Drawing Option.....	18
Color Output.....	19
Group Color Change Option.....	20
Recalculating Existing Drawing.....	21
Outputting to the Plotter.....	22
Hrdcpy Fortran.....	25
Outputting to the Chromatics.....	26
Secondary Revisions.....	29

III. CONCLUSION

Conclusions.....	32
Future Work.....	37

REFERENCES.....	38
-----------------	----

IV. APPENDICES

Appendix A - Saving drawings on Floppy Disk.....	A-1
Appendix B - ASCII Equivalence Table.....	B-1
Appendix C - Variable Definitions.....	C-1
Appendix D - Source Listings	
✓ Armod Exec.....	D-1
Arch2 Exec.....	D-2
Archgraf.2 Indices.....	D-5
✓ Armain Fortran.....	D-8
✓ Arckattn Fortran.....	D-26
✓ Ardevice Fortran.....	D-27
Ardraw Fortran.....	D-35
✓ Arhidlin Fortran.....	D-55
✓ Ario Fortran.....	D-69
✓ Arparams Fortran.....	D-78
✓ Arsketch Fortran.....	D-90
Ardcpy Fortran.....	D-98
Appendix E - Creating Primitive Objects	
File Graph Object.....	E-1
Appendix F - Sample Output.....	F-1

INTRODUCTION

OVERVIEW

This report provides documentation for the reorganization and revisions to ARCHGRAF, an architectural graphics program.

The reorganization is intended to make future revisions, corrections, and modifications easier to implement and less costly in both human and computer time. Revisions included correcting various miscellaneous defects, compiling under the newer Fortran H Extended Compiler, making changes necessary to run under new system support routines, updating and consolidating plotter routines, adding interface routines for the Chromatics terminal, adding color capabilities for Chromatics and plotter output, modifying the solar and rotation drawing options, and making some options easier for the first-time or infrequent users.

Source code for the ARCHGRAF.2 files is provided as well as source code for the supporting exec and hardcopy routines. Documentation of the revisions and reorganization and access to the source code will hopefully encourage other changes, speed the familiarization process and help facilitate future modification and maintenance.

Although the user will know any version of this program only as ARCHGRAF, throughout this report the following conventions will be used:

ARCHGRAF.1 - "old" version, before implementation of changes in this report;

ARCHGRAF.2 - "new" version, as implemented utilizing modifications discussed in this report;

ARCHGRAF - used when referring to the Archgraf system in general, applies to either ARCHGRAF.1 or ARCHGRAF.2;

ARCH2 EXEC - "new" version of the ARCHGRAF.1 EXEC, in its implemented state will be renamed "ARCHGRAF".

MOTIVATION

There were several levels of motivation involved in the desire to implement a project based on the ARCHGRAF.1 program. These levels were represented by both personal considerations and possible benefits to be derived by both the architecture and computer science departments.

With a background in architecture, it was natural to be drawn to a project with architectural applications. Having employed the program as a user, it was possible to appreciate its potential for expansion and improvement and its value to the architecture and the computer science departments. The project also presented the opportunity to learn a new language, Fortran, which is used extensively for architectural applications as well as exposure to several types of hardware. Since ARCHGRAF.1 is used regularly in two classes and is the only interactive graphics program generally in use on campus, this project offered the

opportunity to do work of some lasting value rather than something which would be used or read once and then confined to a shelf.

The advantages for the architecture department were rather elementary. System support routines and system procedures had changed since the last ARCHGRAF module was created. There was no program or procedural documentation other than the user's manual. Source code was not readily available or even accessible but located in many files scattered throughout the system. Some code had to be taken from old cards or printouts as no source files existed. There was, therefore, a need to make ARCHGRAF more manageable and provide an environment which would make future modification and maintenance easier.

Similarly there are several advantages to the computer science department in maintaining a copy of this program. The computer graphics class needs some type of demonstration program showing a real world application proving there are reasons (other than illustrating computer games) for wanting to draw computer pictures. ARCHGRAF can also provide a base for experimentation and expansion while providing the experience of working within a predefined programming environment. For example, a student interested in hidden line algorithms could experiment with implementation without having to spend time writing a supporting program to draw the objects. Some of the educational advantages to be

gained from this type of "modular" programming are discussed in the February, 1981, Communications of the ACM, ACM President's Letter, "Throwaway Programs".

With a minimum of communication between departments, students from both curriculums could benefit. Architecture students would receive updated and expanded program capabilities and computer students would gain experience programming for a user other than another computer scientist.

BACKGROUND

It is not the intent here to explain all the options available in ARCHGRAF.1 as there is a separate user's manual available. Included here, however, is a general overview of the program. Since ARCHGRAF.2 is a superset of ARCHGRAF.1, revisions to the user's manual will consist primarily of additions as no user operations have been deleted.

ARCHGRAF is an interactive graphics program used primarily for creating two-dimensional representations of three dimensional architectural objects (although it is not limited to architectural objects). The program is used principally by two classes. The Regional and Community Planning Department uses it in 109-705, Planning Communications, and the Architecture Department uses it in 105-710, Topics in Architectural Design Methods. The program is used as a computer familiarization tool and as an

introduction to computer-generated graphics applications. While input can be entered from any terminal connected to the Intel AS/5 computer in Cardwell Hall, output in the form of drawings can be received only at one of the following terminals: the SAI (Science Applications, Inc.) Plasmascope in Seaton Hall, either of the two Tektronix 4054 Graphics Terminals in Seaton Hall (other Tektronix terminals may be used if available), DEC GT40 (unavailable at Kansas State University), and, with ARCHGRAF.2, the Chromatics terminal in Fairchild Hall. Output in hard copy form may be obtained from the Tektronix terminals in Seaton Hall from a Tektronix 4610 Hard Copy Unit or from any terminal by outputting to the Calcomp Plotter at Cardwell Hall. Under ARCHGRAF.2 the Chromatics terminal and the Calcomp plotter can produce color output.

The drawings are created from groupings constructed by the user and consisting of one or more previously defined primitive objects. The user may define an object through use of the Summagraphics Intelligent Digitizer and the sibling program ARCHOB (currently only available on the Tektronix 4054's). Objects may also be created from a Tektronix terminal by utilizing the "sketch" option. Objects already defined for the user and currently available are a cylinder, prism, pyramid, screen, triangle, wedge, trees, people, an icosahedron, automobile, line, cube, and square.

Groups are defined under the "Group" command by

inputting an object name, the xyz coordinate of its origin, its length (along the x axis), its width (along the y axis), its height (along the z axis), and the axis and angle of rotation if any. The groups can then be drawn in several different formats.

The types of drawing formats available are perspective, solar, rotation or elevation, stereo, or movie. The perspective option allows the user to specify the xyz coordinate of the viewer, the viewed point, and the cone of vision (useful for controlled distortion or simulating photographic lenses). These line drawings can then be used as the base or guide for more detailed architectural renderings. The solar option presents a drawing of the object from the viewpoint of the sun. This is extremely useful in the design of passive and active solar energy features. The rotation option draws a parallel projection (non-perspective 3D) picture. This is useful for plans, elevations, isometric or axiometric drawings. The stereo option creates stereo pair perspectives for viewing with a stereoscope. The movie option creates a series of perspective drawings which can be used to simulate the walk-through of an area or facilitate the production of an actual photographic movie. This option does not, however, create real time motion as in a true movie or vector-refresh graphics system.

The orientation of the coordinate system may be a

potential point of confusion. An orientation often used in computer graphics is shown in Figure 1a. The orientation used in ARCHGRAF is shown in Figure 2a. A possible explanation for this difference may lie in the origins of computer and architectural graphics. Computer graphics evolved from a vertical surface, the CRT. A histogram, for example, viewed on a CRT would be oriented as shown in Figure 1b. A draftsman, however, would traditionally view or draft a histogram from a horizontal position, on a drafting table. The orientation would then be as shown in Figure 2b. The choice is, of course, arbitrary insofar as the computer is concerned. This does demonstrate, however, the importance of understanding the user's background, environment, and needs so that the transition to "computerization" can be accomplished smoothly and quickly with a minimum of confusion.

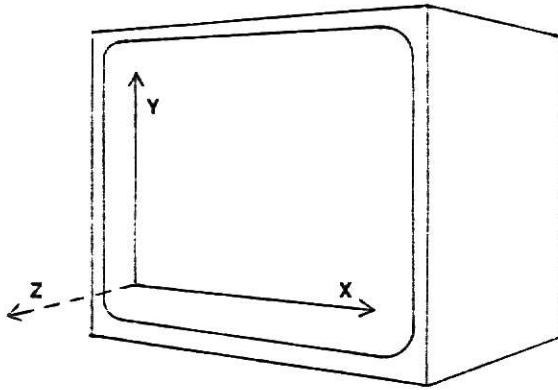


Figure 1a

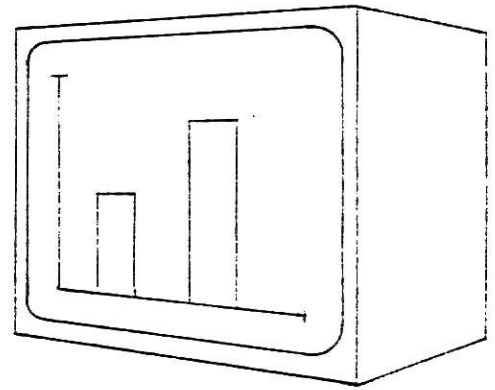


Figure 1b

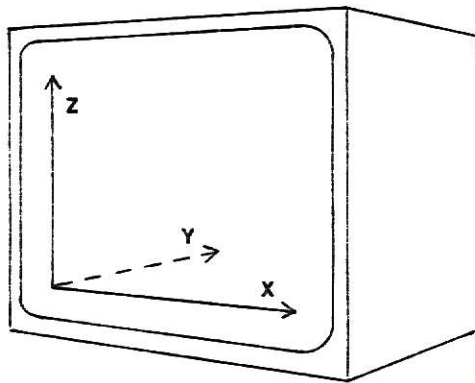


Figure 2a

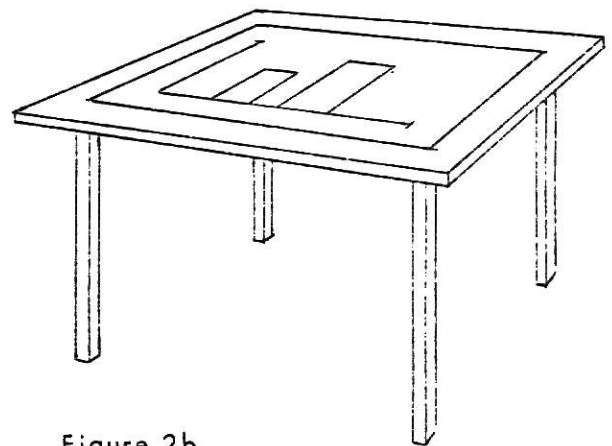


Figure 2b

REORGANIZATION

One of the problems with ARCHGRAF.1 is that it is distributed throughout numerous files stored in various locations.* Furthermore, while some files are very small, the main Fortran source code file consists of approximately 4,700 lines and requires an extremely large (and expensive) virtual machine to edit and compile. The first objective was, therefore, to break the Fortran portion of ARCHGRAF.1 into manageable pieces. The new Fortran files for ARCHGRAF.2 need to be as small as possible without creating too many files to manage. They also need to be grouped logically so that they will be relatively easy to find. This is accomplished by grouping the subroutines by function (localization principle). This also reduces the number of files involved in recompilation when specific corrections or changes are made. Each file is also made small enough to fit the default editor size to reduce the expense of editing.

* The following files are necessary to run ARCHGRAF.1:

an Archgraf Module file (the source is available only from the Archives); Archgraf Exec, which uses Addobj Exec and Hardcpy Exec; Addobj Exec uses the file Graph Object as input to Archgraf; and Hardcpy Exec uses the CMS command "OSJOB" which uses an ARCHGRAF output file called File8 Deck and calls a Proclib procedure named Hardcpy2 which calls a system maintained program called Hrdcpy2. Cklib routines and Fortran free-format routines are needed as well as two other device support files called GT40 Fortran and Plots Fortran.

The ARCHGRAF.2 program is organized into seven Fortran files and one Assembly language (Assemble) file. The GT40 and Plots Fortran files are included within one of the seven Fortran files. The files are as follows:

ARMAIN - Main program and model building subroutines;

ARDEVICE - Device interface subroutines;

ARDRAW - Drawing subroutines;

ARHIDLIN - Hidden line subroutines;

ARIO - Input/Output subroutines;

ARPARAMS - Drawing parameter subroutines (drawing types);

ARSKETCH - Sketch subroutines;

ARCKATTN ASSEMBLE - assembly subroutine.

The following files are also used:

ARMOD EXEC - Links text files and creates module file;

ARCH2 EXEC - New exec to run ARCHGRAF.2;

GRAPH OBJECT - Input file for standard objects;

HRDCPY FORTRAN - replaces system maintained HARDCPY2.

The problem with the Archgraf Exec is just the opposite of the ARCHGRAF.1 program in that it needs too many files. The Addobj and Hardcopy Execs are incorporated into the ARCH2 EXEC in conformity with the localization principle.

The file Graph Object is left unmodified since it is

alone in its function as input file for the standard objects, needs no compilation and may have object definitions easily added or deleted.

After reorganizing the program each file was compiled. Before any changes were made the program was loaded to discover system routine changes and missing files, subroutines, or libraries needed to make a new module which would run under the updated system.

REVISION CONSIDERATIONS

There are certain principles which pertain to the development of all revisions. On the user level, options and procedures should be as simple and as self-explanatory as possible. This is desirable because most users of ARCHGRAF are first-time or infrequent users unfamiliar with both ARCHGRAF and computers in general. Explanations and options should not be too unwieldy or tutorial lest the experienced user becomes inconvenienced. On the programming level procedures or code should be both simple and as efficient as possible. Code should be easy to understand and follow since many different people will be working with the program and programming "tricks" which may cause confusion should be avoided. Structured programming techniques should be observed as much as is possible within the confines of Fortran and an existing unstructured program.

In the following chapter each revision is discussed in

detail. Only those parts of a subroutine or section of code which have been added or changed are discussed. Modifications and additions in the Fortran source code are noted by the outlining of large contiguous blocks or by the underlining of individual lines or changes.

REVISIONS

CHROMATICS LOGON PROCEDURES

An acoustical modem is used to communicate with the Ite1 AS/5 in Cardwell Hall. After connecting the modem to the correct I/O port and turning on power to the Chromatics the following sequence of keys is pressed:

User: RESET

BOOT

CRT OS

ESC H

ESC R 0 7

ESC "H" sets the terminal to half-duplex mode. ESC "R" sets the port and communication rate. The I/O port may either be "1" or "0" and the communication rate will probably depend upon the modem. "7" sets the rate to 1200 baud (see the Chromatics Operators Guide for other rate codes).

Dial the number given by Operations in Cardwell. A tone will sound when connection has been made and the telephone receiver may be placed in the modem. After the "online" prompt use standard CMS logon procedures.

EDITING AND COMPILING

Each file can be edited in the default editor but the virtual machine size must be increased to compile. This can be accomplished by typing the CMS command "RESTOR nnnnK" (1024k maximum). The RESTOR command will redefine the

virtual machine and return to CMS. All the files will currently compile under 760k (the Fortran H Extended Compiler requires 680k to operate). The file can then be compiled by typing "FORTHQ filename". File ARCKATTN ASSEMBLE may be reassembled if necessary by typing "ASSEMBLE ARCKATTN". After compilation the virtual machine can be reduced (larger machines are more expensive). If the program is going to be loaded and run a minimum of 470k will be required (type "RESTOR 470K").

ARMOD EXEC - LINKING THE TEXT FILES

If the default disk space is being used and ARMOD EXEC is going to be used to generate a module file, then all Fortran and Listing files will probably have to be erased from the "A" disk to insure sufficient space for the module file. Several useful CMS commands for keeping track of disk space, file manager space and file sizes are: "QUERY DISK A", "FMQUERY DISK", and "LISTFILE (ALLOCATE)".

ARMOD EXEC is used to link the Archgraf.2 Text files and generate the ARCHII MODULE. All necessary system or support routines are contained in the libraries listed in the Global commands except for the Fortran free-format routines which are accessed through ZDISK EXEC. When ARMOD EXEC is executed (by typing "ARMOD") the following messages will appear:

```
DUPLICATE IDENTIFIER 'CKRI';  
DUPLICATE IDENTIFIER 'CKERR';
```

```

DUPLICATE IDENTIFIER 'CHKPOS';
DUPLICATE IDENTIFIER 'CKPTR';
DUPLICATE IDENTIFIER 'CKZERO';
DUPLICATE IDENTIFIER 'CKMODE';

```

These can be ignored as they have no effect on ARCHGRAF.2.

If debugging is being done and ARMOD EXEC is used more than once, the following messages will also appear:

```

DISK Z ALREADY EXISTS;
FILE 'CKLIB TXTLIB A' ALREADY EXISTS,
SPECIFY 'REPLACE'.

```

These messages may also be ignored.

ARCH2 EXEC

In the old version of this exec, in order to use a saved group or object one had to use a "signon" with parameters. Instead of "signing on" by typing "ARCHGRAF" the user typed "ARCHGRAF O=object_filename G=group_filename". Infrequent or inexperienced users were likely to forget this procedure. The procedure has been retained for long time users, but if "O=fn" or "G=fn" is not used then the new exec will execute as follows:

User: Arch2 (Archgraf)

Comp: IF YOU WISH TO ADD YOUR OWN OBJECTS TYPE YES

U: Yes

C: ENTER OBJECT FILE NAME

U: "filename"

If the file is not found the following message is printed:

```

C: OBJECT FILE 'filename' NOT FOUND.
  ENTER OBJECT NAME OR CANCEL.

```


If an object has been saved, either by "O=" or by the above procedure the option to add the standard objects is presented.

C: IF YOU WISH TO ADD THE STANDARD OBJECTS TO YOUR OWN
OBJECT TYPE YES

If answered "YES" the standard objects (in GRAPH OBJECT) are appended to the user's object file. The ADDOBJ EXEC which appended the user's file with the GRAPH OBJECT file has been incorporated into the ARCH2 EXEC thereby eliminating the necessity for separate file storage and retrieval.

If the user has no saved object or does not wish to use it during this session then the following will result:

C: IF YOU WISH TO ADD YOUR OWN OBJECTS TYPE YES

U: No (or carriage return)

C: IF YOU WISH TO ADD SAVED GROUPS TYPE YES

The responses for adding groups are the same as for objects.

If the plotter has been used during the Archgraf session then the option is presented to either cancel or proceed with the plotter execution. The HARDCOPY EXEC has also been incorporated into the ARCH2 EXEC.

The actual plotting is submitted as an OS Batch job using the CMS Exec "OSJ". The OSJ Exec submits a job to OS and provides the necessary JCL, job, and password cards. FILE8 DECK is the Archgraf output file submitted as input to OSJ and contains JCL to locate and run the HRDCPY routine as well as the input used by HRDCPY to execute the plot. If

the plot is completed FILE8 DECK will be erased. If an error occurs in the execution of OSJ or HRDCPY, FILE8 DECK will not be erased. Depending upon the type of error, after recovery it may be possible to execute the plot from outside ARCHGRAF.2 by typing the following:

U: Osj File8 Deck (Time 0,40)

In any event the information will not be lost as was previously the case and may be used again when the error is corrected or checked for faulty input.

ROTATION DRAWING PARAMETER

ARPARAMS: Subroutine ROT

Changes in this procedure allow the user to choose rotation by axis and angle (degrees) or pick a predetermined rotation. The first message is:

C: 1 - ELEVATION
 2 - ROTATION BY AXIS AND DEGREES
 TYPE 1 OR 2

If "2" is chosen the rotation option proceeds as in ARCHGRAF.1 with the exception that a counter has been added. Rather than requesting data a fourth time and requiring a null entry, the subroutine now asks for data for only three axes.

If "1" is chosen the following menu is printed:

C: TYPE NUMBER FOR DESIRED ELEVATION
 1 - PLAN
 2 - NORTH
 3 - EAST
 4 - SOUTH
 5 - WEST

The correct data for axes and angles of rotation are assigned directly to "AN" and indirectly to "AX" which are used to calculate the transformations necessary to achieve the requested elevation. The counter is now used to determine which data has already been entered for each elevation. Other predetermined rotations can, in the same manner, easily be added to the routine.

SOLAR DRAWING OPTION

ARPARAMS: Subroutine SOLAR

In ARCHGRAF.1 the user had to provide the declination as well as the latitude and hour of day. This required access to a declination table or graph. ARCHGRAF.2 eliminates this need by calculating the declination. The previous hour-of-day entry procedure was also somewhat confusing. Noon was "0", afternoon hours positive and morning hours negative (for example, 10:00 A.M. would be "-2.0"). Hours are now entered as military time which is unambiguous. The first message is:

```
C:      ENTER 1 OR 2
        1 - BY DECLINATION
        2 - BY DATE
```

If the user answers with "1" the procedure remains unchanged from ARCHGRAF.1 with the exception of the hour data entry which is now given in military time.

If the user answers with "2" the following message is printed:

C: ENTER LATITUDE,MONTH,DAY,HOUR:
 (EX - 39.0 5 18 17)

The declination is computed by the following formula:

$$d = 23.45 \sin[((n-80)/370) \times 360]$$

where n is the Julian date. If the calculation is for the southern hemisphere the latitude should be entered as negative so that the declination will be negative.

COLOR OUTPUT

With the addition of the Chromatics terminal and color pen capabilities on the new Calcomp Plotter color output becomes a possibility.

ARDRAW: Subroutine DRAW

For the first new drawing (using the "Draw" command) ARCHGRAF.2 will ask the user to enter the device name. If the new drawing is not the first drawing then a check is made to determine whether the Chromatics or plotter is being used. If the device name entered was "UMPL","PLOT", or "CHRO" the routine will set logical "COLRET" to "true" to identify the return point and will then go to the color change option (Option 18). The entry point is different than if normally using Option 18 so that the user will be asked:

C: DO YOU WISH TO USE COLOR? (Y/N)

If answered "yes" then the routine proceeds with the group color assignments using the same code as the color change option. If answered "no" the routine returns and proceeds as

usual for a new drawing.

GROUP COLOR CHANGE OPTION (Option 18)

ARDRAW: Subroutine DRAW

If this option is selected a check is made to insure that the device being used is capable of color output. If the device is neither the plotter nor the Chromatics the following message is printed:

C: OUTPUT DEVICE MUST BE UMPL, PLOT, OR CHRO
 FOR COLOR OUTPUT

The subroutine then returns to the option prompt so that a new option may be chosen by the user.

If the device is correct the routine proceeds by setting all groups to black (white for Chromatics) so that any groups not specified will default to black (or white). The user is first reminded of which groups are to be drawn by listing the group numbers.

C: GROUPS TO BE DRAWN ARE:
 list of group numbers

Group numbers are then entered for each color as follows:

C: ENTER GROUPS TO BE BLUE:
U: Group numbers (or carriage return)
C: ENTER GROUPS TO BE GREEN:
U: Group numbers (or carriage return)
C: ENTER GROUP NUMBERS TO BE RED:
U: Group numbers (or carriage return)

If a carriage return is made no groups are set to that color. After each color entry a check is made to insure that the group numbers entered are among those to be drawn. If an invalid number is entered the following message is printed:

C: GROUP 'number' IS NOT IN LIST TO BE DRAWN

The routine then returns to the appropriate color to allow the user to re-enter the group numbers. After all colors have been processed correctly "DONE" is printed and the routine returns to the option prompt.

RECALCULATING EXISTING DRAWING (Option 12)

ARDDRAW: Subroutine DRAW

This option is also the code used to calculate a new drawing. The key to this option is the call to Subroutine BLDMAT. BLDMAT makes the final calculations for the creation of the two dimensional representation of the groups (listed in array 'M(30)') to be drawn. The results are the values in the common block labeled VECTOR with the X and Y coordinates and the drawing commands in the "X", "Y", and "ISW" buffers. BLDMAT is called for each group to be drawn and each time the results are added to the "X", "Y", and "ISW" buffers. "KS" is the buffer subscript which points to the last position used in the "X", "Y", and "ISW" buffers.

The array "ICHPN(31,2)" is used to store the data needed to correctly change group colors. The rows or subscripts represent the group number. Column One contains

the "X", "Y", and "ISW" buffer subscript (KS) which denotes the start of the group. Column Two contains the color number (IPNUM) for the group.

A check is made to insure that the first group to be assigned in "ICHPN" is active. This group then has its starting point assigned as "1" (this being the first subscript in the buffers). Each group is then confirmed as being active and then processed by BLDMAT. After each call to BLDMAT variable "KS" will contain the last buffer position so that the next group can be assigned a starting point of "KS+1". This group is then assigned its appropriate color number from array "ICOL".

This same sequence of code is also used in Subroutine MOVIE to calculate the movie drawings.

OUTPUTTING TO THE PLOTTER

ARDRAW: Subroutine PUTOUT

The new Colcomp plotter has two paper sizes with 10.9" and 32.9" widths. A check is made on the drawing's maximum width in inches (variable "XMX"). If "XMX" is greater than ten inches (and the plotter is the specified output device) the following message is printed:

```
C:      **WARNING - A DRAWING SIZE OF 'xmx' IS TOO LARGE
        FOR THE PLOTTER'S STANDARD 10 INCH PAPER WIDTH.
        DO YOU WISH TO ENTER A NEW DRAWING SIZE (YES/NO)?
```

If answered "yes" a new drawing size will be entered by the user. If answered "no" the user is asked:

C: DO YOU WISH TO USE WIDE PAPER (YES/NO)?

If answered "yes" the plot is calculated and wide paper is used. If answered "no" narrow paper is used.

If "XMX" is greater than 32 inches the following message is printed:

C: **WARNING - A DRAWING SIZE OF 'xmx' IS TOO LARGE
FOR THE PLOTTER'S WIDE PAPER WIDTH.
DO YOU WISH TO ENTER A NEW DRAWING SIZE (YES/NO)?

If answered "yes" a new drawing size can be entered by the user. If answered "no" the plot is calculated and wide paper is used. If the user specifies a drawing size larger than the paper width, one of three situations may result. If the coordinates are not too far out of the plotter's range the plotter will calculate new end points and the drawing will be clipped. This is not assured, however, and if the coordinates are too far out-of-bounds the plot will degenerate into seemingly random lines. The third possibility is that the drawing will fit the paper width. This can happen because the paper width is actually along the Y axis and the maximum "Y" distance may be less than 32 inches. The paper length along the X axis may be 50 to 100 inches but only the "X" distance is actually checked because the "Y" distance in inches at this point in the program is still unknown.

ARDEVICE: Subroutine UMPL

If the buffer subscript number in this routine equals

the buffer subscript number in "ICHPN" then a check is made to see if the corresponding group color is different from the color currently being used. If the colors are the same then the pen color does not have to be reset and variable "IPNUM" remains "0". If the new color is different, then the new color number is assigned to "IPNUM" and passed to either Entry PENDNS or PENUPS. The appropriate Entry is determined by "ISW". If "ISW" is "0" then the plotter pen is to be up (move to new location without drawing) and PENUPS is called. If "ISW" is "1" then the plotter pen is to be down (drawing or vector mode) and PENUPS is called. PLTEND is called when the last coordinates for the drawing have been reached.

ARIO: Subroutine PLDNO

Subroutine PLDNO constructs the ARCHGRAF output file "FILE8 DECK". This file consists of JCL to set up the plotter and the previously compiled and linked Fortran routine HRDCPY followed by the input for HRDCPY. It is unnecessary for each drawing to have its own JCL and is more economical to run a number of plots under the same JCL. There is, however, separate JCL for wide and for narrow paper. The JCL is added, therefore, whenever there is a change in paperwidth. The jobs are separated automatically by the sequence of "if" statements using the logicals "NAR" and "LASTNAR" (see list of variable explanations).

The procedures for both entries PENDNS or PENUPS are the same. "JPEN" is assigned the plotter pen command number (2, 3, or -3). If there is a pen color change to be made "JPEN" is also used to pass the pen color number "IPNUM". If "IPNUM" were added as a parameter the length of FILE8 DECK would be increased by 25% since "IPNUM" would also have to be passed with each set of coordinates and pen commands. By multiplying "IPNUM" by 100 and adding "JPEN" both commands can be carried by one variable and easily separated in Routine HRDCPY. The coordinates and pen commands are passed from Subroutine UMPL to PLDNO one set at a time, processed and stored in PLDNO and then written to FILE8 DECK four sets at a time. Presumably this is done to save memory space.

HRDCPY FORTRAN

Routine HRDCPY is a complete (except for input) Fortran program which is compiled and linked separately from ARCHGRAF.2 and uses the plotter commands and procedures from the updated plotter guide and newsletters. The program can be compiled and linked from CMS by using the OSJ command if job cards are not included or by RJE if job cards are included in the file. HRDCPY is stored by the JCL it contains and is activated by the JCL in FILE8 DECK. HRDCPY then reads its input (x and y coordinates and pen commands) from FILE8 DECK four sets at a time.

OUTPUTTING TO THE CHROMATICS

All graphics commands and coordinates are modified or constructed to meet the requirements and format of the receiving terminal, loaded sequentially into a buffer and transmitted to the terminal for immediate execution.

All the commands and coordinates must be assigned their ASCII equivalents as shown in the following table (also see the Appendix for the ASCII Equivalence Table for the Chromatics).

COMMAND		KEYBOARD		ASCII (decimal)
Visible Cursor	=	Mode J	=	01 74
Blind Cursor	=	Mode K	=	01 75
Plot Mode On	=	Mode G	=	01 71
Plot Mode Off	=	Cancel Mode	=	21
Concatenated Vector	=	(	=	40
Home Cursor	=	Home	=	28
Set Background On	=	Mode M	=	01 77
Set Background Off	=	Mode N	=	01 78
Select Color	=	Mode C	=	01 67
Clear Screen	=	Erase Page	=	12
Colors:				
Black	=	0	=	48
Blue	=	1	=	49
Green	=	2	=	50
Cyan	=	3	=	51
Red	=	4	=	52
Magenta	=	5	=	53
Yellow	=	6	=	54
White	=	7	=	55

The first step in processing these commands and coordinates is taken in Subroutine CHRO.

ARDEVICE: Subroutine CHRO

To set the terminal to an initial state Entry MATOUT is

called when CHRO is first entered. The commands to set the background to black, clear the screen, disable the cursor and put the terminal in plot mode are entered into the buffer (array IBUFF). A check is made to insure that there is sufficient buffer space for the next set of instructions. The coordinates are set to zero (ASCII 48). Each set of X and Y coordinates and the accompanying "ISW" plot command are then processed.

If the buffer subscript equals the buffer number in "ICHPN" then Entry NEWCOL is called (using the color number from "ICHPN") and the color change commands are put into the buffer. Only four color possibilities have been implemented in order to match the plotter's four color limitation. Addition of the other Chromatics colors could be accomplished by simply following the established procedures in Subroutines DRAW and CHRO.

The first step in coordinate conversion is made by multiplying each coordinate by the scale factor (SF) and adding the offset (XL or YL) which is usually zero. This result is then multiplied by the number of pixels per inch for each axis in order to set the screen proportions and scale correctly. These modified X and Y coordinates and the plot command are then passed to Subroutine MATIC.

ARDEVICE: Subroutine MATIC

If the terminal is not in plot mode the Cursor Disable

and Plot Mode commands are entered into the buffer. This may be necessary, for example, if the buffer is output before completion of the drawing. If "IP" (ISW) is "0" the graphics command is to move the cursor to a new position without drawing. This is accomplished by entering the concatenated vector command which terminates the current series of vectors, if any, moving to the location given by the coordinates which follow and then continuing to connect each coordinate pair until another command is given. If "IP" is "1" the graphics command is to draw to this coordinate. No command entry is needed since the concatenated vector command will continue to draw until cancelled.

The Chromatics terminal expects a three digit coordinate so that each coordinate must be separated into three digits and each digit entered into the buffer in the proper order. After each coordinate is divided into the three digits, high, medium, and low (least significant is low), each digit must be converted to its ASCII equivalent. This can be derived by adding or offsetting each digit by "48" (see ASCII Equivalence Table in Appendix). The X coordinate is entered into the buffer first with the most significant digit first and the least significant digit last.

After both X and Y coordinates have been entered into the buffer a check is made to insure that there is

sufficient buffer space for the next series of instructions. If "IP" is "-1" or there is insufficient buffer space the buffer will be output to the terminal. If "IP" is "-1" signifying the end of the drawing instructions, the plot mode cancel, home and visible cursor commands are entered into the buffer. The foreground is set to white so that the text will not change colors with each drawing. Subroutine USTRAN which actually transmits the buffer is a routine located in CKLIB TXTLIB.

SECONDARY REVISIONS

ARMAIN: Block Data, Subroutine CHGRP and GINPUT

The "LINE" object has been added to the body of ARCHGRAF.2. To add the Line object to a group the X, Y, and Z coordinates of both end points are entered rather than the usual origin, length, width, and height parameters. This object may cause problems if hidden line removal is desired. The code for this change had been written by another student but never added to the program. This modification therefore consisted of simple editing and little attempt was made to test or debug this feature. To distinguish from other changes, lines of code pertaining to the Line object are marked by "L"'s in the line numbers. As this feature only affects the program when the line object is specified any problems can be avoided by simply not using the Line

object.

ARMAIN: Subroutine OINPUT

Variables "KT" and "LT" are checked to insure they will not exceed the buffer or array range. They are both incremented by "1" in Subroutine NAOBJ. If either is not less than 1000, a warning message is printed and the "Object" command is cancelled.

Variable "JT" is the object counter and is also incremented in Subroutine NAOBJ. It is checked to insure that it does not exceed the bounds of the array "NAME" where it is used as part of a linked list. If "JT" is not less than 30, the maximum number of objects, a warning message is printed and the "Object" command is cancelled.

ARIO: Subroutine DINI

Device sizes were updated for the new plotter and corrected for the SAI terminal. "XM" is the maximum size in inches of the terminal window (or paper) along the X axis, while "YM" is the maximum size in inches along the Y axis.

ARDRAW: Subroutine BLDMAT

The changes in BLDMAT were necessary for compilation under the Fortran H Extended Compiler. These were the only severe errors encountered. All other changes due to compilation under the newer compiler were for warnings and

are not specifically listed in this report.

ARDRAW: Subroutine DRAW

New, more descriptive device names were added. The old names (shown first) were retained for long time users.

Tektronix terminals	- 'STOR' or 'TEK'
GT40 terminal	- 'GT40'
Calcomp Plotter	- 'UMPL' or 'PLOT'
SAI Plasmascope	- 'NEW' or 'SAI'
Chromatics terminal	- 'CHRO'

ARDRAW: Subroutine DRAW

Option 6 (Output current drawing on another device) did not work when used for drawings of over 1000 lines. Once the intended program path was determined only a few additions were necessary to make the correction.

CONCLUSIONS

Numerous problems encountered while attempting to reconstruct ARCHGRAF made the reconstruction and reorganization the most time consuming and difficult portion of this project. Finding source code files for the various routines was an exercise in persistence and perseverance. While the personnel of the Computing Center were very helpful there was very little organization. Most system routines are stored only as text or module files and there is no central directory or location from which sources may be found. Source files are often kept on individual accounts so that by talking to a number of people someone is eventually found who may remember under which account the file is stored. After a futile search through the system files, a Hardcopy2 Fortran card deck was finally found on top of a cabinet under some old listings which were to be discarded. Although the old Fortran free-format routines would not work under the updated system, the new free-format routines had been available only to the systems programmers until my inquiry resulted in their being made accessible through "ZDISK". Furthermore, CMS system changes during this time necessitated procedural alterations at three different times.

The reasons for wanting to consolidate the scattered files and some of the programs which had been maintained by

the systems personnel partially resulted from my experiences in attempting to find these files. The first is the fear that the routines will be accidentally lost. The second reason is to make correcting and updating easier by insuring that the source files are readily accessible. The third reason is that by having the source files readily accessible the routines will not have to be maintained by the systems personnel. Updating routines used primarily for ARCHGRAF receives a low priority for system programmer time because ARCHGRAF is not used by a high percentage of people relative to the total university (as are, for example, various grader or statistical packages). By retaining the source and utilizing the consultants and system programmers to help ascertain needed corrections (high priority is given to those physically present) and making these corrections oneself, updates can be implemented in much less time than by submitting through channels and waiting for the system programmer to make the corrections. If time is not a factor or if the corrections necessary are too difficult or otherwise require a system programmer, then at least there is a readily available source copy for them to use.

Some problems were inherent within the program. ARCHGRAF.1 was written in 1969, before structured programming techniques were widely known and portions of the program are extremely hard to follow. The use of many subroutines is positive and brings much of the structure to

ARCHGRAF, but many of these subroutines are not particularly well structured. Much of this problem is due to the Fortran language which makes structuring difficult because of its reliance on the "GO TO" statement. The program could be greatly improved if written in a language such as Pascal or PL/I which contain data structures and commands which Fortran does not possess. A Record data structure could consolidate many of the arrays into logical groups and eliminate the need for many of the currently used subscript pointers which are more obscure. The "IF-THEN-ELSE" or "IF-THEN BEGIN-ELSE BEGIN" statement would be extremely valuable and eliminate the need for most of the current "GO TO" statements necessitated by Fortran's limited "IF" statement. Understandability could be increased by alphabetic label names rather than Fortran's limited numeric labels. The numeric labeling could be greatly improved, however, by simply re-editing so that the labels are in ascending order rather than pseudo-randomly distributed.

The treatment of many variable names was also a source of confusion and an impediment to rapid comprehension. With the lack of documentation to provide explanation many variables, particularly the logicals, seemed quite cryptic. The Fortran imposed limitation of variable names of no more than six characters may moderately contribute to this problem. Retaining throughout the program the same name for a particular variable would greatly improve cognition.

Variables passed as parameters change names needlessly and Common variables also often have different names in different subroutines. In a large program which has a considerable number of variables these superfluous names only exacerbate the problem.

The above summary of difficulties encountered is a circuitous means of describing some of the perceptions gained from this project. By having to deal with an existing program and its attendant faults a clearer or more forceful insight may be gained of our own programming and design techniques and how to improve these techniques.

Although ARCHGRAF is not necessarily an example of good programming technique nor state-of-the-art computer graphics it is probably a good example of the type of programming many students are going to have to work with at some time in the future since maintenance plays such a large part in most computer installations. Furthermore, problems encountered while working with someone else's program are different from those dealt with when programming from scratch and can be stressed most strongly by personal experience. Ideas discussed in class such as the need for adequate documentation (and what comprises adequate documentation) cannot be truly appreciated until one works on a large program without this documentation. The long term advantages of thorough planning, including ease of maintenance, and the usefulness of software engineering

techniques such as module or functional specifications also become more apparent. Had ARCHGRAF.1 been a currently written, well documented program in a structured language its educational benefit would have been much reduced.

FUTURE WORK

There are a number of projects which could be undertaken that could be worthwhile both in terms of improvement to ARCHGRAF and to student experience. Following is a list of some possible projects.

Explore new methods to improve user interface and input; write interface between the Chromatics disk and ARCHGRAF so that data can be stored and passed back to ARCHGRAF; fix or replace the hidden line routine; write interface routines for the computer science department digitizer and plotter for input and output through the Chromatics; write routines to calculate data for primitive object creation in an interactive environment; and change the Stereo drawing parameter to take advantage of the color output capabilities.

Many of the changes necessary to update ARCHGRAF would not be worth the effort relative to the educational value. ARCHGRAF could be completely rewritten in Pascal in less time than it would take to completely decypher and rework ARCHGRAF in Fortran. ARCHGRAF is a good starting point, however, and can serve at the very least as an inspiration for related projects.

REFERENCES

Sturgul, John R. and Merchant, Michael J. Applied Fortran IV Programming. Wadsworth Publishing Company, Inc., Belmont, California, 1976.

Chromatics Preliminary Operator's Manual (Revised),
Chromatics, Inc., Atlanta, Georgia, November 1978.

Chromatics Disk Software Reference Manual,
Chromatics, Inc., Atlanta, Georgia, 1978.

Denning, Peter J. ACM President's Letter, "Throwaway Programs". Communications of the ACM, Vol. 24, No. 2, February 1981.

SAVING DRAWINGS ON FLOPPY DISK

A drawing from the Chromatics can be saved on a floppy disk to be redrawn later at the terminal by using the Create buffer and Redraw function.

- 1) Load a formatted diskette into one of the Chromatics disk drives.
- 2) After the last option number, usually 12 or 13, has been entered but before the carriage return, press the CREATE key.
- 3) After the drawing is complete press the CREATE key again to cancel the Create Mode.
- 4) Press the DISK OS key.
- 5) After the "*DOS" prompt type "BUFF filename.BUF". This will store the drawing on the diskette.
- 6) You may then return to continue the program by pressing the CRT OS key.
- 7) The last drawing in the create buffer can be drawn by pressing the REDRAW key.
- 8) To retrieve a drawing from the diskette press the DISK OS key.
- 9) After the "*DOS" prompt type "DRAW filename.BUF". This will load the drawing commands back into the Create buffer and overlay any drawing previously in the create buffer.

- 10) Press the CRT OS key.
- 11) Press ERASE PAGE.
- 12) Press the REDRAW key.

ASCII CODE ASSIGNMENT

B-1

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NOTE: ENTRIES WITH X INDICATE UNUSED ANSI ASCII CODES.

NOTE: A7 ON IMPLIES SPECIAL GRAPHIC CHARACTERS.

NOTE: ENTRIES WITH X INDICATE UNUSED ANSI ASCII CODES.

The following list contains all of the new variables in ARCHGRAF.2 as well as a number of variables from ARCHGRAF.1 which have been deduced and are used in the subroutines affected by ARCHGRAF.2. The following conventions apply to the Common Variable list:

Common label/List of variable names - (Subroutines in which
names are used)
Variable name - description

The following conventions apply to the Individual Variable list:

Variable name - description

COMMON VARIABLES

B/XST,XLOC,YLOC - (DRAW)

SF, XL , YL - (CHRO,UMPL)

XS,XLOC,YLOC - (PUTOUT)

XST - Scale factor

XLOC - X offset

YLOC - Y offset

COLOUR/ICHPN(31,2) - (MAIN,DRAW,CHRO,UMPL,MOVIE)

ROWS: Group Number

COLUMN 1: Marks beginning of group by NMP at time of
addition of group to drawing buffers

COLUMN 2: Color number for group:
1 = Black (plotter) or white (Chromatics)
2 = Blue (plotter) or cyan (Chromatics)
3 = Green
4 = Red

JCLFLG/JCLON,NAR,LASTNAR - (MAIN,PUTOUT,PLDNO)

JCLON - True: first plot job, need JCL
False: Not first job

NAR - True: Use narrow paper
False: Use wide paper

LASTNAR - True: Last JCL was for narrow paper
False: Last JCL was for wide paper

JOHN/INIT

INIT - True: Initialization is needed
False: Do not initialize

MISCD/IDT,IDEV,SF,FRA - (DRAW,DINI,PUTOUT)

IDT - Drawing type number:
1 = Rotation
2 = Perspective
3 = Stereo
4 = Solar
5 & 6 = Movie

IDEV - Device number:
1 = STOR or TEK
2 = GT40
3 = UMPL or PLOT
4 = NEW or SAI
5 = CHRO

FRA - True: Clear screen
False: Overlay screen

SIZE/XM(6),YM(6) - (DINI,PUTOUT)

XM - ROWS: Device number

COLUMN: Device window size in inches along X axis

YM - ROWS: Device number

COLUMN: Device window size in inches along Y axis

TABLES/NAME(30,5),GROUP(30,3)

NAME - ROWS: Object number

COLUMN 1: Object name

COLUMN 2: KT+1

COLUMN 3: LT+1

COLUMN 4: Number of points in object

COLUMN 5: Number of lines in object

GROUP - ROWS: Group number

COLUMN 3: 0 = Inactive
1 = Active

VECTOR/DX(1000),DY(1000),ISW(1000),NMP -

(MAIN,CHRO,UMPL,PUTOUT)

X(1000), Y(1000), IS(1000),KS - (DRAW)

DX(1000),DY(1000),ISB(1000),NSTOR - (BLDMAT)

XT(1000),YT(1000),ISG(1000),NSTOR - (MOVIE)

DX - Buffer for X coordinates

DY - Buffer for Y coordinates

ISW - Drawing commands:

1 = draw (vector) mode

0 = locate (move but do not draw) mode

-1 = end of drawing

NMP - pointer to last position in DX, DY, and ISW buffers

INDIVIDUAL VARIABLES

ANS - Answer from user

AN - Angle in degrees

AX,AXX - Axis

CNTR - Counter

COLRET - True: Return to location in program for new drawing

ELV - Elevation drawing type

GC - Group color number (entry by user)

HR - hour in military time

IBUFF(2031) - Buffer for transmitting coordinates and
commands

to Chromatics, SAI, and GT40 terminals

ICOL(30) - Contains group numbers of groups to be drawn

IDEVT - temporary device number (see IDEV)

IPEN(4) - Pen commands for plotter:

2 = pen down (draw)

3 = pen up (move)

-3 = end of plot

102 = change to black pen, pen down

103 = change to black pen, pen up

etc

a0b

pen command

pen color

IPNUM - Color number:

1 = Black (plotter) or white (Chromatics)

2 = Blue (plotter) or cyan (Chromatics)

3 = Green

4 = Red

IPT - Counter

IXHC - High X coordinate (most significant digit)

IXMC - Middle X coordinate (middle digit)

IXLC - Low X coordinate (least significant digit)
IYHC - High Y coordinate (most significant digit)
IYMC - Middle Y coordinate (middle digit)
IYLC - Low Y coordinate (least significant digit)
JPEN - plotter pen command
LEN - Pointer to last location in buffer IBUFF
M(30) - Contains, in order of user entry, group numbers
 of groups to be drawn
MON(12,2) - ROWS: Months (January = 1)
 COLUMN 1: Number of days per month
 COLUMN 2: Julian date of last day of previous
month
NOUT - Total number of groups to be drawn
OP - Option number (1 through 18)
TEMP - Temporary storage
XX - Drawing size (maximum direction in inches along X
axis)

SOURCE LISTING

ARMOD EXEC

&CONTROL OFF NOMSG
EXEC ZDISK
&CP BYPASS ON
FMR CKLIB TXTLIB (U VM999
GLOBAL TXTLIB USLIB1 CKLIB FORTLIB CMSLIB
GLOBAL MACLIB OSMACRO TSOMAC
LOAD ARMAIN ARCKATTN ARDEVICE ARDRAW ARHIDLIN ARI0 ARPARAMS ARSKETCH
GENMOD ARCHII MODULE A1 (NOMAP

SOURCE LISTING

ARCH2 EXEC

```

&CONTROL OFF NCMSG
&ERROR &GOTO -NOG
STATE GRAPH OBJECT
&GOTO -OKG
-NOG &ERROR &CONTINUE
FMR GRAPH OBJECT (U VM4B9
-OKG &ERROR &CONTINUE
&O = F
&GL = F
ERASE NEW GROUP
&TYPE
&IF &INDEX EQ 0 &GOTO -OBMAN
&IF &1 EQ $ARCHOB &GOTO -SKIPIT
&A = &SUBSTR &1 1 2
&IF &A NE O= &GOTO -TRY
&1 = &SUBSTR &1 3 6
&GOTO -OBCHK
-TRY &IF &A NE G= &GOTO -ERROR
&2 = &SUBSTR &1 3 6
&GOTO -GCHK
-GROUP &A = &SUBSTR &2 1 2
&IF &A NE G= &GOTO -START
&2 = &SUBSTR &2 3 6
&GOTO -GCHK
-OBMAN &TYPE IF YOU WISH TO ADD YOUR OWN OBJECTS TYPE YES
&READ VARS &ANS
&TYPE &ANS
&IF .&ANS NE .YES &GOTO -GRMAN
&TYPE
&TYPE ENTER OBJECT FILE NAME
&READ VARS &NAME
&1 = &SUBSTR &NAME 1 6
&GOTO -OBCHK
-GRMAN &TYPE IF YOU WISH TO ADD SAVED GROUPS TYPE YES
&READ VARS &ANS
&IF .&ANS NE .YES &GOTO -OBJ
&TYPE
&TYPE ENTER GROUP FILE NAME
&READ VARS &NAME
&2 = &SUBSTR &NAME 1 6
&GOTO -GCHK
-OBCHK &ERROR &GOTO -NOOBJ
STATE &1 OBJECT
&O = T
&ERROR &CONTINUE
-GTEST &IF &INDEX GE 2 &GOTO -GROUP
&IF &INDEX EQ 0 &GOTO -GRMAN
&2 = DUMMY
&GOTO -OKOBJ

```

```
-GCHK &ERROR &GOTO -BADG
STATE &2 GROUP
&GL = T
&ERROR &CONTINUE
&IF &O NE T &GOTO -GRAPH
&GOTO -START
-NOOBJ &TYPE OBJECT FILE &1 NOT FOUND
&TYPE ENTER OBJECT FILE NAME OR CANCEL
&READ VARS &G
&IF &G EQ CANCEL &1 = GRAPH
&IF &G EQ CANCEL &GOTO -GTEST
&1 = &G
&GOTO -OBCHK
-BADG &CONTINUE
&TYPE GROUP FILE &2 NOT FOUND
&TYPE ENTER GROUP FILE NAME OR CANCEL
&READ VARS &G
&IF &G EQ CANCEL &GOTO -START
&2 = &G
&GOTO -GCHK
-ERROR &TYPE ERROR IN ARCHGRAF COMMAND
&TYPE SEE CURRENT WRITEUP AND TRY AGAIN
&EXIT
-OBJ &IF &O NE T &GOTO -GRAPH
&2 = DUMMY
&GOTO -START
-GRAPH &1 = GRAPH
&IF &GL EQ T &GOTO -START
&2 = DUMMY
&GOTO -START
-SKIPIT &1 = &2
&IF &INDEX LE 2 &GOTO -AG
&2 = &3
&GOTO -START
-AG &2 = DUMMY
-OKOBJ &CONTINUE
-START &TYPE ***** ARCHII LOADING *****
&TYPE ***** WAIT A MINUTE PLEASE *****
&ERROR &GOTO -ARCH
STATE ARCHII MODULE
&GOTO -FILES
-ARCH &ERROR &CONTINUE
&CONTROL ERROR MSG
&CONTROL OFF NOMSG
-FILES &CONTROL ERROR MSG
&ERROR &CONTINUE
FMSET C OFF
&IF &1 EQ GRAPH &GOTO -FILDEF
&TYPE IF YOU WISH TO ADD THE STANDARD OBJECTS TO
&TYPE YOUR OWN OBJECT TYPE Y.
&READ VARS &A
&IF .&A EQ .Y &GOTO -ADDOB
&GOTO -FILDEF
-ADDOB &CONTROL OFF NOMSG
```

```
&BEGSTACK
BOT
DEL
&END
&STACK GETFILE &2 OBJECT
&STACK FILE
EDIT &1 OBJECT
-FILDEF &CONTINUE
FILEDEF 1 DISK NEW GROUP ( RECFM F LRECL 80 BLKSIZE 80 )
FILEDEF 2 DISK NEW OBJECT ( RECFM F LRECL 80 BLKSIZE 80 )
FILEDEF 4 TERMINAL ( RECFM F LRECL 100 BLKSIZE 100 )
FILEDEF 5 DISK &1 OBJECT ( RECFM F LRECL 80 BLKSIZE 80 )
FILEDEF 6 TERMINAL ( RECFM F LRECL 132 BLKSIZE 132 )
FILEDEF 7 DISK &2 GROUP ( RECFM F LRECL 80 BLKSIZE 80 )
FILEDEF 8 DISK FILE8 DECK ( RECFM F LRECL 80 BLKSIZE 80 )
FILEDEF 9 TERMINAL
CP TERM BYPASS ON
CP SET MSG OFF
&TYPE 'escape character for erase page'
ARCHII
&ERROR &GOTO -COPY
&CONTROL OFF NOMSG
STATE FILE8 DECK
&TYPE DO YOU WANT A HARDCOPY (YES/NO)?
&READ VARS &ANS
&IF &ANS EQ &30 &GOTO -COPY
&IF &ANS NE YES &GOTO -COPY
&CONTROL ERROR
&ERROR &GOTO -NOERAS
EXEC OSJ FILE8 DECK ( TIME 0,40 )
ERASE FILE8 DECK
-NOERAS &CONTINUE
&CONTROL CMS
&ERROR &GOTO -FIN
-COPY &CONTROL OFF NOMSG
&ERROR &GOTO -FIN
STATE NEW GROUP
&ERROR &GOTO -RENAME
STATE &2 GROUP
-PRO2
&TYPE DO YOU WANT TO REPLACE FILE " &2 " ?
&ERROR &GOTO -NEWNAME
&READ VARS &ANS
&IF &ANS EQ NO &GOTO -NEWNAME
&IF &ANS NE YES &GOTO -PRO2
ERASE &2 GROUP
&GOTO -RENAME
-NEWNAME &TYPE TYPE NEW NAME OR CANCEL.
&READ VARS &ANS
&ERROR &GOTO -FIN
&IF &ANS EQ &30 &GOTO -NEWNAME
&IF &ANS NE CANCEL &GOTO -NE1
ERASE NEW GROUP
&GOTO -FIN
```

```
-NE1 RENAME NEW GROUP A1 &ANS GROUP A1
&GOTO -FIN
-RENAME &CONTINUE
&TYPE ENTER FILE NAME FOR SAVED GROUP
&READ VARS &B
RENAME NEW GROUP A1 &B GROUP A1
&2 = &B
-FIN &ERROR &GOTO -EXIT
STATE &2 GROUP
FMS
-EXIT CP SET MSG ON
&CONTROL CMS MSG
```

INDEX OF SUBROUTINES AND ENTRIES - ORDER OF OCCURANCE

NAME	PAGE		
ARMAIN		ARDEVICE	ARHIDLIN
MAIN.....8	COMP.....27	CELATE....55	ARPARAMS
ACTIVE.....10	DIST.....27	ENTRA.....62	EYESP.....78
BINTER.....11	CHRO.....27	HIDLI.....65	MOVIE.....78
BLANK.....12	MATIC.....27	HVB.....66	SAME.....79
CHGRP.....12	MATOUT...29	RSCA.....66	PERS.....81
PRINT.....13	NEWCOL...30	INOUT.....67	PERS2....82
CMS.....16	GT40.....31	KDET.....67	PERS1....83
COMCAN.....16	NEWD.....31	PRCVM.....67	ROT.....84
GINPUT.....16	NEWIN....31	VERDIS....68	SOLAR.....86
GINPT.....16	GT40IN...31		STERO.....87
GRIN.....17	BUFF3.....31	ARIO	ARSKETCH
GT.....18	BUFF3A...33	CKABTN....69	BUILD.....90
NAOBJ.....18	SAICDE...33	CKMA.....69	CLOCK.....92
NAMES.....19	SAIDAT...33	CKVA.....69	FINDPT....93
NEWG.....19	UMPL.....34	DINI.....69	GRID.....93
NEWO.....19		DEVICE...70	HITE.....94
GINPUT.....20	ARDRAW	ERROR.....70	LNADD.....94
OTA.....21	BLDMAT...35	ERR.....70	PEXPND....94
OTA1.....22	COMB.....37	ERRORS...70	PLSAVE....96
OTT.....22	DRAW.....38	FREAD.....70	SCREEN....96
OTS.....22	DRAW2....38	APARM....71	SKETCH....97
CTR.....22	LOUT.....45	IPARM....71	
ODEL.....22	MATF.....46	RPARM....71	
REPT.....22	MAT1.....47	JOY.....72	
SAVE.....23	MAT2.....47	PLDNO....75	
RDSV.....24	PUTOUT...47	PENDMS...76	
USER.....24	RAST.....50	PENUPS...76	
VALID.....24	RTEST....50	PLTEND...76	
	SSAVE.....51	SYSTEM....77	
ARCKATTN	SWAP.....52		
CKATTN....26	TRAT.....52		
ATTNEX....26	VBLD.....53		
	VBL.....53		

* Entry names are indented

INDEX OF SUBROUTINES AND ENTRIES - ALPHABETICAL

NAME	PAGE		
ACTIVE.....10		GRID.....93	PENUPS.....76
APARM.....71		GRIN.....17	PERS.....81
ATTNEX.....26		GT.....18	PERS1.....83
BINTER.....11		GT40.....31	PERS2.....82
BLANK.....12		GT40IN....31	PEXPND.....94
BLDMAT.....35		HIDLI.....65	PLDNC.....75
BLOCK DATA...9		HITE.....94	PLSAVE.....96
BUFF3.....31		HVB.....66	PLTEND....76
BUFF3A....32		INOUT.....67	PRINT.....13
BUILD.....90		IPARMS....71	PROVM.....67
CELATE.....55		JOY.....72	PUTOUT.....47
CHGRP.....12		KDET.....67	RAST.....50
CHRO.....27		LNADD.....94	RDSV.....24
CKABTN.....69		LOUT.....45	REPT.....22
CKATTN.....26		MAIN.....8	ROT.....84
CKMA.....69		MAT1.....47	RPARM.....71
CKVA.....69		MAT2.....47	RSCA.....66
CLOCK.....92		MATF.....46	RTEST.....50
CMS.....16		MATIC.....28	SAICDE....33
COMB.....37		MATOUT....29	SAIDAT....33
COMCAN.....16		MOVIE.....78	SAME.....79
COMP.....27		NAMES.....19	SAVE.....22
DEVICE....70		NAOBJ.....18	SCREEN.....96
DINI.....69		NEWCOL....30	SKETCH.....97
DIST.....27		NEWD.....31	SOLAR.....86
DRAW.....38		NEWG.....19	SSAVE.....51
DRAW2....38		NEWIN....31	STERO.....87
ENTRA.....62		NEWO.....19	SWAP.....52
ERR.....70		ODEL.....22	SYSTEM.....77
ERROR.....70		OINPUT.....20	TRAT.....52
ERRORS....70		OTA.....21	UMPL.....34
EYESP.....78		OTA1.....22	USER.....24
FINDPT.....93		OTR.....22	VALID.....24
FREAD.....70		OTS.....22	VBL.....53
GINPT.....16		OTT.....22	VBLD.....53
GINPUT.....16		PENDNS....76	VERDIS.....68

* Entry names are indented

C		00001000
C		00002000
C	-----	00003000
C	MAIN PROGRAM - COMMAND PARSER.	00004000
C	-----	00005000
	COMMON /MISC/JT,KT,LT,L,N,B	00006000
	INTEGER CINDX(24)/1,3,5,7,9,11,13,15,17,19,	00007000
1	21,23,25,27,28,29.	00008000
2	35,31,33,36,37,38,40/	00009000
	INTEGER IT,B	00010000
	REAL COM(40)	00011000
	DATA COM/'STOP','QUIT','SYST','O ','OBJE','G ','GROU','D ','	00012000
	1'DRAW','C ','CHAN','P ','PRIN','A ','ACTI','N ','NAME',	00013000
	1'S ','SAVE','RE ','REP ','R ','REPT','REPE','READ','U ',	00014000
	1'USER','BATC','BLAN','EXPL','COMM','CMS ','CMS ','H ','HELP',	00015000
	1' ','NEWG','NEWO','BUIL','SKET'/	00016000
C		00017000
C	_PRINT THE LOGO	00018000
	PRINT 60	00019000
	60 FORMAT(8X,'*****'/	00020000
	2 8X,'* THE ARCHITECTURAL GRAPHICS SYSTEM */	00021000
	3 8X,'*****'/)	00022000
C		00023000
C	_PRINT THE PREFIX CHARACTER AND	00024000
	20 PRINT 30	00025000
	30 FORMAT('?')	00026000
C	_READ THE COMMAND.	00027000
	CALL CKATTN(3,&20)	00028000
100	CALL FREAD(4,&100,&50)	00029000
	CALL APARM(COMM,' ',4,&100)	00030000
	DO 101 I=1,40	00031000
	IF(COMM.EQ.COM(I))GO TO 14	00032000
101	CONTINUE	00033000
	CALL ERRORS(0,0,'BAD COMM',&20)	00034000
C		00035000
C	_BRANCH TO THE PROPER ROUTINE.	00036000
14	GO TO(1,1,1,2,2,3,3,4,4,5,5,6,6,8,8,9,9,10,10,12,12,13,12,12,13,	00037000
	115,15,16,17,18,18,19,19,21,21,50,51,52,40,40),I	00038000
1	CALL SYSTEM	00039000
2	CALL OINPUT(&20)	00040000
3	IF(I.EQ.6) CALL GINPT(&20)	00041000
	CALL GINPUT(&20)	00042000
4	CALL DRAW(&20)	00043000
5	CALL CHGRP(0,&20)	00044000
6	CALL PRINT(&20)	00045000
8	CALL ACTIVE(&20)	00046000
9	CALL NAMES(&20)	00047000
10	CALL SAVE(&20)	00048000
12	CALL REPT(&20)	00049000
13	CALL RDSV(&20)	00050000
15	CALL USER(&20)	00051000
16	CALL ERR(.TRUE.,&20)	00052000
17	CALL BLANK(&20)	00053000
40	CALL BUILD(&20)	00054000
50	CALL DRAW2(&20)	00055000

```

51 CALL NEWG(&20)                                00056000
52 CALL NEWO(&20)                                00057000
19  CALL CMS                                      00058000
    GO TO 20                                      00059000
C                                                00060000
C_"COMMAND" COMMAND.                            00061000
18  DO 1005 IT=1,24                              00062000
    ITZ=IT+1                                       00063000
    PRINT 1001, COM(CINDX(IT)),IT,ITZ             00064000
1001 FORMAT(4X,A4,(' ',I2,' ',I2,' '))           00065000
1005 CONTINUE                                     00066000
    GO TO 20                                      00067000
C                                                00068000
C_"HELP" COMMAND.                              00069000
21  PRINT 1002                                    00070000
1002 FORMAT(' FOR A LISTING OF COMMANDS TYPE "COMM". THE NUMBERS FOLLOW
1ING',                                           00072000
1/,' THE COMMAND NAMES REFER TO LINES IN THE FILE ARCH:GINFO', 00073000
1/,' WHERE THE COMMAND DESCRIPTION BEGINS. TO GET COMMAND',    00074000
1/,' DESCRIPTIONS, TYPE "CMS", THEN $COPY ARCH:GINFO(N1,N2)', 00075000
1/,' WHERE N1,& N2 ARE THE COMMAND NOS. WHEN YOU WISH',        00076000
1/,' TO RETURN TO THE PROGRAM TYPE "$RES".'//) 00077000
    GO TO 20                                      00078000
    END                                           00079000
C-----00080000
C ARCH:GRAPHIC SOURCE PROGRAM.                  00081000
C-----00082000
C                                                00083000
    BLOCK DATA                                00084000
    COMMON /TABLES/NAME,GROUP,IOBJ,ROBJ/MISC/JT,KT,LT,L,N,B 00085000
1/DATA/X,Y,Z,LINE/TABLE2/IG,RG,NGT             00086000
1/MATRIX/DRAT,MATG,MATO,MAT                    00087000
    COMMON/VECTOR/DX,DY,ISW,NMP,GOOD,FIRST       00088000
    COMMON/PDATA/ D,OPT,RX1,RX2,RY1,RY2         00089000
    COMMON /PTS/ PT,BAD,BACK                    00090000
    COMMON/PLANE/NPP,NPLA,IVEC,MYLINE            00091000
    INTEGER NPLA(30,2)/1,6,2,27*0,37,1,31,27*0/,MYLINE(1000,2),
LL192LLL
1IVEC(800)/4,1,4,3,2,4,2,3,7,6,4,6,7,8,5,4,5,8,4,1,4,3,4,8,7,4,1,2,
00093000
26,5,2,1,3*2,1,764*0/,NPP/37/                 LL194LLL
    LOGICAL PT,BAD                              00095000
    LOGICAL*1 BACK/F/                           00096000
    REAL RX1/-4.6/,RX2/4.6/,RY1/-4.6/,RY2/4.6/ 00097000
    DIMENSION DX(1000),DY(1000),ISW(1000)       00098000
    LOGICAL FIRST/.FALSE./                      00099000
    LOGICAL*4 GOOD/T/                            00100000
    INTEGER NAME(30,5)/'SQRX','CUBE','LINE',27*0,2*1,9,27*0,4,8,2
LL1101LLL
1,27*0,2*1,13,27*0,4,12,1,27*0/,LINE(1000,2)/1,2,3,4,1,5,6,7,2*8
LL1102LLL
2,3,6,1,987*0,2,3,4,1,5,6,7,8,5,4,7,2,2,987*0/,
LL1103LLL
3GROUP(30,3)/90*0/,
00104000
4IOBJ(300,3)/900*0/,
00105000
5IG(100)/100*0/,
00106000
6 JT/3/,KT/10/,LT/13/,L/0/,N/0/,NGT/0/         LL107LLL
    REAL X(1000)/0.,2*1.,2*0.,2*1.,2*0.,1.,990*0./,
LL1108LLL
1Y(1000)/4*0.,4*1.,0.,1.,990*0./,
LL1109LLL
2Z(1000)/2*0.,2*1.,2*0.,2*1.,0.,1.,990*0./,
LL1110LLL
3ROBJ(300,8)/2400*0./,
00111000

```


4B/ ' ' /	00112000
5, RG(100,5)/500*0./	00113000
REAL DRAT(4,4)/16*0./,MATG(4,4)/16*0./,MATO(4,4)/16*0./	00114000
1, MAT(4,4)/16*0./	00115000
DIMENSION CKRINF(9,12)	00116000
COMMON/CKRI/ BASE,CKRINF,ICKRAC	00117000
COMMON/CKERR/ IERROR	00118000
COMMON/CKCPOS/ ICX, ICY	00119000
COMMON/CKPTR/ IP	00120000
COMMON/CKZERO/ NZERO	00121000
COMMON/CKMODE/ MODE, ISKIP	00122000
COMMON/COLOUR/ ICHPN(31,2), ICOL(30)	00123000
COMMON/ JCLFLG/ JCLON, NAR, LASTNAR	00124000
LOGICAL JCLON/.TRUE./	00125000
LOGICAL NAR, LASTNAR	00125.50
DATA IP, NZERO/1,5/	00126000
DATA MODE, ISKIP/0,1/	00127000
DATA BASE, ICKRAC/0.,1/	00128000
DATA CKRINF /0.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00129000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00130000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00131000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00132000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00133000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00134000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00135000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00136000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00137000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00138000
10.,0.,1023.,0.,1023.,0.,0.,1.,1.,	00139000
10.,0.,1023.,0.,1023.,0.,0.,1.,1./	00140000
DATA IERROR, ICX, ICY/0,0,767/	00141000
END	00142000
C	00143000
C	00144000
C-----	00145000
C SUBROUTINE ACTIVE - TO CHANGE THE STATUS OF A GROUP	00146000
C VIA THE "ACTIVE" COMMAND.	00147000
C-----	00148000
SUBROUTINE ACTIVE(*)	00149000
COMMON /TABLES/NAME, GROUP /MISC/III, J, K, L, N, B	00150000
INTEGER ST, NAME(30,5), GROUP(30,3)	00151000
INTEGER A/'ACTI'/, I/'INAC'/'	00152000
1 PRINT 100	00153000
100 FORMAT('ENTER GROUP NUMBER AND STATUS(EG. 2, ACTIVE):')	00154000
5 CALL FREAD(4, &5, &6)	00155000
6 CALL IPARM(LG, 0, &5)	00156000
CALL APARM(ST, ' ', 4, &5)	00157000
IF(LG.LE.L)GO TO 10	00158000
CALL ERROR(8, L, LG.0., B)	00159000
RETURN 1	00160000
10 IF(LG.LE.0)CALL COMCAN(&33)	00161000
IF(ST.EQ.A)GO TO 12	00162000
IF(ST.EQ.I)GO TO 14	00163000
CALL ERROR(3, LG, 0, 0., B)	00164000
GO TO 1	00165000
12 GROUP(LG, 3)=1	00166000

```

      PRINT 102                                00167000
102  FORMAT(' DONE')                          00168000
      RETURN 1                                00169000
14   GROUP(LG,3)=0                            00170000
      PRINT 102                                00171000
33  RETURN 1                                  00172000
      END                                      00173000
C                                           00174000
C                                           00175000
C-----                                00176000
C SUBROUTINE BINTER - BUILD INTERFACE TO ARCH:GRAPHIC. COPIES 00177000
C                      THE OBJECT INTO THE DATA STRUCTURE. 00178000
C-----                                00179000
      SUBROUTINE BINTER(NEXT,OBNAME)           00180000
      COMMON/STOR/LB,LE,PX,PY,PZ,NL,NP,PTEMP,PLANE 00181000
      COMMON/PLANE/NPP,NPLA(30,2),IVEC(800),MYLINE(1000.2) 00182000
      COMMON/MISC/JT,KT,LT/DATE/X,Y,Z,LINE      00183000
      REAL X(1000),Y(1000),Z(1000),PX(350),PY(350),PZ(350) 00184000
      INTEGER LB(500),LE(500),PTEMP(500),LINE(1000,2),CBNAME 00185000
      LOGICAL PLANE(50)                        00186000
      K1=KT+1                                  00187000
      L1=LT+1                                  00188000
      CALL NACBJ(OBNAME,NP,NL)                 00189000
C                                           00190000
C_TRANSFER POINTS                            00191000
      DO 10 I=K1,KT                           00192000
      X(I)=PX(I-K1+1)                          00193000
      Y(I)=PY(I-K1+1)                          00194000
      Z(I)=PZ(I-K1+1)                          00195000
10  CONTINUE                                  00196000
C                                           00197000
C_TRANSFER LINES                             00198000
      DO 11 I=L1,LT                           00199000
      LINE(I,1)=LB(I-L1+1)                     00200000
      LINE(I,2)=LE(I-L1+1)                     00201000
11  CONTINUE                                  00202000
C                                           00203000
C_TRANSFER PLANES                            00204000
      NPLA(JT,1)=0                             00205000
      NPLA(JT,2)=NPP                           00206000
      NPLANE=2                                  00207000
      N=1                                       00208000
12  NUM=PTEMP(NPLANE)+1                       00209000
      IF(.NOT.PLANE(N))GO TO 14                 00210000
      DO 13 I=1,NUM                            00211000
13  IVEC(NPP+I-1)=PTEMP(NPLANE+I-1)           00212000
      NPP=NPP+NUM                              00213000
      NPLA(JT,1)=NPLA(JT,1)+1                 00214000
14  IF(N.EQ.PTEMP(1))RETURN                   00215000
      N=N+1                                    00216000
      NPLANE=NPLANE+NUM                       00217000
      GO TO 12                                 00218000
      END                                      00219000
C                                           00220000
C                                           00221000
C-----                                00222000

```

```

C SUBROUTINE BLANK - TO BLANK REPEATS OF GROUPS VIA THE "BLANK" COMMAND.00223000
C-----00224000
      SUBROUTINE BLANK(*)00225000
      INTEGER IBV(100)00226000
      COMMON/TABLE2/IG(100),RG(100,5),NG00227000
      INTEGER GR(20),ANS/'OK '/,A00228000
1      PRINT 9800229000
      98 FORMAT('ENTER GROUP NUMBERS TO BE BLANKED:')00230000
      700 CALL FREAD(4,&700,&701)00231000
      701 DO 720 I=1,2000232000
            CALL IPARM(GR(I),0,&700)00233000
            IF ((GR(I) .GE. 0) .AND. (GR(I) .LE. NG)) GO TO 72000234000
            CALL ERRORS(8,GR(I),'GROUP # ',&1)00235000
      720 CONTINUE00236000
            DO 2 I=1,2000237000
            IF(GR(I).EQ.0) GO TO 300238000
2      CONTINUE00239000
3      I=I-100240000
            IF(I.LE.0) CALL COMCAN(&33)00241000
            PRINT 97,(GR(J),J=1,I)00242000
      97 FORMAT('WISH TO BLANK ALL REPITITIONS'/00243000
            2 'OF GROUPS',20I3)00244000
            PRINT 9600245000
      96 FORMAT('OK(OK)?')00246000
      8 CALL FREAD(4,&8,&9)00247000
      9 CALL APARM(A,' ',&8)00248000
            IF(A.NE.ANS) CALL COMCAN(&33)00249000
            DO 50 J=1,I00250000
            JJ=000251000
            DO 10 K=1,NG00252000
            IF(IG(K).EQ.GR(J)) GO TO 1000253000
            JJ=JJ+100254000
            IBV(JJ)=K00255000
      10 CONTINUE00256000
            NG=JJ00257000
            IF(NG.EQ.0) GO TO 6000258000
            DO 50 II=1,NG00259000
            IG(II)=IG(IBV(II))00260000
            DO 50 KK=1,500261000
      50 RG(II,KK)=RG(IBV(II),KK)00262000
      60 CONTINUE00263000
            PRINT 9400264000
      94 FORMAT(' DONE')00265000
      33 RETURN 100266000
            END00267000
C00268000
C00269000
C-----00270000
C SUBROUTINE CHGRP - TO EDIT GROUPS VIA THE "CHANGE" COMMAND.00271000
C-----00272000
      SUBROUTINE CHGRP(IX,*)00273000
      LOGICAL VALID,SW00274000
      DATA LINE/'LINE'/00275000
      INTEGER*2 KAX00276000
      INTEGER GNAM(30)/'GR1 ','GR2 ','GR3 ','GR4 ','GR5 ','GR6 ','GR7 ',00277000
      1'GR8 ','GR9 ','GR10','GR11','GR12','GR13','GR14','GR15',00278000

```

1	'GR16','GR17','GR18','GR19','GR20','GR21','GR22','GR23',	00279000
2	'GR24','GR25','GR26','GR27','GR28','GR29','GR30'/	00280000
	INTEGER#2 AXA(7) / 'Z','X','Y','Z','X','Y',' ', ' ', AX	00281000
	COMMON/TABLES/NAME, GROUP, IOBJ, ROBJ /MISC/JT,KT,LT,L,N,B	00282000
	INTEGER NAME(30,5), GROUP(30,3), IOBJ(300,3), PR	00283000
	REAL ROBJ(300,8), LEN	00284000
C		00285000
C		00286000
1	PRINT 98	00287000
98	FORMAT('ENTER GROUP NUMBER AND PRINT OPTION(EG. 3,1,):')	00288000
	IX=0	00289000
	PR=0	00290000
	GO TO 5	00291000
C		00292000
C_PRINT	GROUP SUMMARY ON SPRINT VIA THE "PRINT" COMMAND.	00293000
	ENTRY PRINT(*)	00294000
	SW=.FALSE.	00295000
	PR=1	00296000
	IX=1	00297000
95	PRINT 96	00298000
96	FORMAT('ENTER GROUP NUMBER:')	00299000
5	CALL FREAD(4,&5,&6)	00300000
6	CALL IPARM(LR,0,&5)	00301000
	CALL IPARM(PR,0,&5)	00302000
	IF(LR.EQ.0 .AND. SW)RETURN 1	00303000
	IF(LR.LE.0)CALL COMCAN(&33)	00304000
	SW=.TRUE.	00305000
	IF(IX.GT.0)PR=1	00306000
	IF(LR.LE.L)GO TO 10	00307000
	CALL ERROR(8,L,LR,1.,B)	00308000
	IXX=IX+1	00309000
	GO TO (1,95),IXX	00310000
10	IF(PR.LE.0)GO TO 30	00311000
	PRINT 101,LR, GROUP(LR,3)	00312000
101	FORMAT('// GROUP # ',2I3/' OB NAME',7X,'XO',7X,'YO',7X,'ZO',	00313000
1	4X,'WIDTH',3X,'LENGTH',3X,'HEIGHT AXIS ANGLE')	00314000
	K=GROUP(LR,1)	00315000
	IN=GROUP(LR,2)	00316000
	DO 18 I=1,IN	00317000
	IF(IOBJ(K,1).GT.0) GO TO 17	00318000
	PRINT 99,I	00319000
99	FORMAT('OBJECT ',I2,' HAS BEEN DELETED.')	00320000
	GO TO 18	00321000
17	KNAM=IOBJ(K,1)	00322000
	IF(KNAM.LE.100) GO TO 1001	00323000
	KNAM=KNAM-100	00324000
	NAMX=GNAM(KNAM)	00325000
	GO TO 1002	00326000
1001	NAMX=NAME(KNAM,1)	00327000
1002	KAXX=ROBJ(K,7)	00328000
	KAX=AXA(KAXX+4)	00329000
	PRINT 102,I,NAMX,(ROBJ(K,J),J=1,6),KAX,ROBJ(K,8)	00330000
102	FORMAT(I3,1X,A4,6F9.2,1X,A2,F7.1)	00331000
18	K=IOBJ(K,2)	00332000
	IF(IX.GT..5)GO TO 95	00333000
C		00334000

C_CHANGE OPTIONS.	00335000
PRINT 103	00336000
103 FORMAT(/,' OPTION=0 TO RETURN'/	00337000
1 ' =1 TO CHANGE ORIGIN'/	00338000
2 ' =2 TO CHANGE SIZE'/	00339000
3 ' =3 TO CHANGE ROTATION'/	00340000
4 ' =4 TO REPLACE OBJECT'/	00341000
5 ' =5 TO ADD OBJECT'/	00342000
6 ' =6 TO DELETE OBJECT'/)	00343000
30 PRINT 104	00344000
104 FORMAT('ENTER OPTION AND OBJECT NUMBER(EG. 3,2,):')	00345000
40 CALL FREAD(4,&40,&41)	00346000
41 CALL IPARM(J,0,&40)	00347000
CALL IPARM(I,0,&40)	00348000
C	00349000
C_BAD OPTION?	00350000
IF(J.LE.0)RETURN 1	00351000
IF(J.LE.6)GO TO 32	00352000
CALL ERROR(9,J,6,0.,B)	00353000
GO TO 30	00354000
32 GO TO (21,22,23,24,25,26),J	00355000
C	00356000
C_CHANGE ORIGIN.	00357000
21 PRINT 105	00358000
105 FORMAT ('ENTER XO, YO, ZO')	00359000
50 CALL FREAD(4,&50,&51)	00360000
51 CALL RPARM(XO,0.0,&50)	00361000
CALL RPARM(YO,0.0,&50)	00362000
CALL RPARM(ZO,0.0,&50)	00363000
CALL OTT(LR,I,XO,YC,ZO)	00364000
GO TO 30	00365000
C	00366000
C_CHANGE SIZE.	00367000
22 PRINT 107	00368000
107 FORMAT('ENTER WIDTH, LENGTH, AND HEIGHT:')	00369000
55 CALL FREAD(4,&55,&56)	00370000
56 CALL RPARM(W,1.0,&55)	00371000
CALL RPARM(LEN,1.0,&55)	00372000
CALL RPARM(H,1.0,&55)	00373000
CALL OTS(LR,I,W,LEN,H)	00374000
GO TO 30	00375000
C	00376000
C_CHANGE ROTATION.	00377000
23 PRINT 108	00378000
108 FORMAT('ENTER AXIS,ROTATION:')	00379000
60 CALL FREAD(4,&60,&61)	00380000
61 CALL APARM(AX,' ',2,&60)	00381000
CALL RPARM(ROT,0.0,&60)	00382000
199 DO 1007 IAX=1,7	00383000
IF(AX.EQ.AXA(IAX)) GO TO 1008	00384000
1007 CONTINUE	00385000
CALL ERRORS(7,0,'BAD AXIS',&30)	00386000
1008 RAX=MOD(IAX-1,3)	00387000
GO TO (200,200,200,111,111),J	00388000
200 CALL OTR(LR,I,RAX,ROT)	00389000
GO TO 30	00390000

16	PRINT 15	LL391LLL
15	FORMAT('INVALID PARAMETERS ON LINE')	LL392LLL
C		00393000
C_REPLACE	OBJECT.	00394000
24	PRINT 109	00395000
109	FORMAT('ENTER NAME XO YO ZO WIDTH LENGTH HEIGHT AXIS ANGLE:')	00396000
65	CALL FREAD(4,&65,&66)	00397000
66	CALL APARM(NAM,' ',4,&65)	00398000
	CALL RPARM(XO,0.0,&65)	00399000
	CALL RPARM(YO,0.0,&65)	00400000
	CALL RPARM(ZO,0.0,&65)	00401000
	CALL RPARM(W,0.0,&65)	00402000
	CALL RPARM(LEN,0.0,&65)	00403000
	CALL RPARM(H,0.0,&65)	00404000
	CALL APARM(AX,' ',2,&65)	00405000
	CALL RPARM(ROT,0.0,&65)	00406000
	IF(NAM.NE.LINE) GO TO 43	LL407LLL
	IF(W.EQ.DX.AND.LEN.EQ.DY.AND.H.EQ.DZ) GO TO 16	LL408LLL
	W=W-DX	LL409LLL
	LEN=LEN-DY	LL410LLL
	H=H-DZ	LL411LLL
43	IF(W.NE.0..OR.H.NE.0..OR.LEN.NE.0.) GO TO 199	00412000
	W=1.	00413000
	LEN=1.	00414000
	H=1.	00415000
	GO TO 199	00416000
111	DO 210 IN=1,JT	00417000
	IF(NAM.EQ.NAME(IN,1))GO TO 201	00418000
210	CONTINUE	00419000
	DO 1003 KN=1,30	00420000
	IF(NAM.EQ.GNAM(KN)) GO TO 1004	00421000
1003	CONTINUE	00422000
	CALL ERRORS(6,0,'BAD NAME',&30)	00423000
1004	IF(VALID(LR,KN)) GO TO 1005	00424000
	GO TO 24	00425000
1005	IN=KN+100	00426000
201	JX=J-3	00427000
	GO TO (400,401),JX	00428000
400	CALL OTA1(I,R,I,IN,XO,YO,ZO,W,LEN,H,RAX,ROT)	00429000
	GO TO 30	00430000
401	CALL OTA(LR,I,IN,XO,YO,ZO,W,LEN,H,RAX,ROT)	00431000
	GO TO 30	00432000
C		00433000
C_ADD	OBJECT.	00434000
25	I=GROUP(LR,2)+1	00435000
	GO TO 24	00436000
C		00437000
C_DELETE	OBJECT.	00438000
26	CALL ODEL(LR,I)	00439000
	PRINT 300	00440000
300	FORMAT(' DONE')	00441000
	GO TO 30	00442000
33	RETURN 1	00443000
	END	00444000
C		00445000
C		00446000

SUBROUTINE CMS	00447000
PAUSE	00448000
RETURN	00449000
END	00450000
C	00451000
C	00452000
C-----	00453000
C SUBROUTINE COMCAN - TO PRINT "COMMAND CANCELLED" .	00454000
C-----	00455000
SUBROUTINE COMCAN(*)	00456000
PRINT 100	00457000
100 FORMAT(' COMMAND CANCELLED')	00458000
RETURN 1	00459000
END	00460000
C	00461000
C	00462000
C-----	00463000
C SUBROUTINE GINPUT - TO READ GROUP INFO FROM A TERMINAL	00464000
C VIA THE "GROUP" COMMAND.	00465000
C-----	00466000
SUBROUTINE GINPUT(*)	00467000
LOGICAL VALID	00468000
DATA LINE/'LINE'/	LL469LLL
INTEGER GNAM(30)/'GR1 ','GR2 ','GR3 ','GR4 ','GR5 ','GR6 ','GR7 ','GR8 ','GR9 ','GR10','GR11','GR12','GR13','GR14','GR15',	00470000
1'GR16','GR17','GR18','GR19','GR20','GR21','GR22','GR23',	00471000
2'GR24','GR25','GR26','GR27','GR28','GR29','GR30'/	00472000
LOGICAL FIR/T/	00473000
COMMON /TABLES/NAME,GROUP /MISC/JT,KT,LT,L,N,B	00474000
INTEGER NAME(30,5),GRUP(30,3)	00475000
REAL AXIS(3),LEN	00476000
INTEGER POFF	00477000
LOGICAL HEAD,ECHO	00478000
INTEGER*2 AXA(7)/'Z','X','Y','Z ','X ','Y ',' '/,AX	00479000
DATA AXIS,NEXT,LAST/' Z',' X',' Y','NEXT','LAST'/	00480000
DATA POFF/'*POF'/	00481000
HEAD=.TRUE.	00482000
GO TO 1000	00483000
C	00484000
C	00485000
ENTRY GINPT(*)	00486000
HEAD=.FALSE.	00487000
ECHO=.FALSE.	00488000
FIR=.FALSE.	00489000
1000 IF(FIR) ECHO=.TRUE.	00490000
FIR=.FALSE.	00491000
IS=1	00492000
1 L=L+1	00493000
GROUP(L,1)=N+1	00494000
GRUP(L,3)=1	00495000
IF(IS.EQ.2) GO TO 12	00496000
PRINT 100,L	00497000
100 FORMAT(' GROUP #',I3)	00498000
IF(HEAD) PRINT 109	00499000
109 FORMAT(' NAME',7X,'XO',7X,'YO',7X,'ZO',4X,'WIDTH',3X,'LENGTH ',	00500000
1'HEIGHT',1X,'AXIS',1X,'ROTATION')	00501000
	00502000

12	NT=GROUP(L,2)+1	00503000
139	IF(IS.EQ.1)PRINT 101,NT	00504000
101	FORMAT(I3)	00505000
	GO TO (2,3),IS	00506000
2	CONTINUE	00507000
700	CALL FREAD(4,&700,&701)	00508000
701	CALL APARM(NAM,' ',4,&700)	00509000
	CALL RPARM(DX,0.0,&700)	00510000
	CALL RPARM(DY,0.0,&700)	00511000
	CALL RPARM(DZ,0.0,&700)	00512000
	CALL RPARM(W,0.0,&700)	00513000
	CALL RPARM(LEN,0.0,&700)	00514000
	CALL RPARM(H,0.0,&700)	00515000
	CALL APARM(AX,' ',2,&700)	00516000
	CALL RPARM(ROT,0.0,&700)	00517000
	HEAD=.FALSE.	00518000
	GO TO 3	00519000
C		00520000
C		00521000
	ENTRY GRIN(IADD,NAM,DX,DY,DZ,W,LEN,H,AX,ROT,*)	00522000
	IS=2	00523000
	IF(IADD.GT.0)GO TO 1	00524000
	GO TO 12	00525000
16	PRINT 15	LL526LLL
15	FORMAT('INVALID PARAMETERS ON LINE')	LL527LLL
	GO TO 139	LL528LLL
C		00529000
C_SPECIAL NAME?		00530000
3	IF(NAM.EQ.NEXT)GO TO 1	0053100
	IF(NAM.EQ.LAST)GO TO 20	00532000
	IF(NAM.EQ.POFF) GO TO 103	00533000
	IF(NAM.NE.LINE)GO TO 43	LL534LLL
	IF(W.EQ.DX.AND.LEN.EQ.DY.AND.H.EQ.DZ) GO TO 16	LL535LLL
	W=W-DX	LL536LLL
	LEN=LEN-DY	LL537LLL
	H=H-DZ	LL538LLL
43	IF(W.NE.0..OR.H.NE.0..OR.LEN.NE.0.) GO TO 13	LL539LLL
	W=1.	00540000
	H=1.	00541000
	LEN=1.	00542000
C		00543000
C_CHECK THE AXIS.		00544000
13	DO 200 IAX=1,7	00545000
	IF(AX.EQ.AXA(IA)) GO TO 201	00546000
200	CONTINUE	00547000
	GO TO 25	00548000
201	IA=MOD(IA-1,3)	00549000
	IF(IA.GT.3) GO TO 25	00550000
C		00551000
C_EXISTING OBJECT?		00552000
9	DO 10 I=1,JT	00553000
	IF(NAM.EQ.NAME(I,1))GO TO 14	00554000
10	CONTINUE	00555000
C		00556000
C_EXISTING GROUP?		00557000
	DO 210 I=1,30	00558000


```

      IF(NAM.EQ.GNAM(I)) GO TO 212                                00559000
210  CONTINUE                                                    00560000
C                                                                    00561000
C_INVALID OBJECT NAME.                                           00562000
      CALL ERROR(6,L,NT,0.,NAM)                                    00563000
      GO TO 700                                                    00564000
212  IF(VALID(L,I)) GO TO 213                                     00565000
      GO TO 700                                                    00566000
213  I=I+100                                                      00567000
C                                                                    00568000
C_ADD THE OBJECT OR GROUP.                                         00569000
14   IF(ECHO.AND.IS.EQ.1)PRINT 98,NAM,DX,DY,DZ,W,LEN,H,AXIS(IA+1),ROT 00570000
      AXX=IA                                                       00571000
98   FORMAT (1X,A4,6F9.2,1X,A4,1X,F8.2)                          00572000
      CALL OTA (L,NT,I,DX,DY,DZ,W,LEN,H,AXX,ROT)                 00573000
      IF(IS.EQ.1)GO TO 12                                          00574000
      RETURN 1                                                     00575000
20   IF(NT.EQ.1)GROUP(L,1)=0                                       00576000
      RETURN 1                                                     00577000
25   CALL ERRORS(7,L,'BAD AXIS',&2)                               00578000
103  ECHO=.NOT.ECHO                                                00579000
      GO TO 139                                                    00580000
      END                                                          00581000
C                                                                    00582000
C                                                                    00583000
C-----                                                           00584000
C SUBROUTINE GT - TO SAVE GROUP REPEAT INFO IN THE DATA STRUCTURE. 00585000
C          (CALLED FROM SUBROUTINE REPT).                          00586000
C-----                                                           00587000
      SUBROUTINE GT(NO,X,Y,Z,A,R,*,*)                             00588000
      COMMON /TABLE2/IG(100),RG(100,5),NGT /MISC/JT,KT,LT,L,N,B 00589000
      IF(NO.LE.L)GO TO 9                                           00590000
      CALL ERROR(8,L,NO,2.,B)                                       00591000
      RETURN 1                                                     00592000
9     IF(NO.LE.0) RETURN 2                                          00593000
      IAX=A+.5                                                      00594000
      IF(IAX.LE.2.AND.IAX.GE.0)GO TO 12                            00595000
      CALL ERROR(7,NO,0,AX,B)                                       00596000
      RETURN 1                                                     00597000
12    NGT=NGT+1                                                    00598000
      IG(NGT)=NO                                                    00599000
      RG(NGT,1)=X                                                    00600000
      RG(NGT,2)=Y                                                    00601000
      RG(NGT,3)=Z                                                    00602000
      RG(NGT,4)=A                                                    00603000
      RG(NGT,5)=R                                                    00604000
      RETURN                                                         00605000
      END                                                           00606000
C                                                                    00607000
C                                                                    00608000
C-----                                                           00609000
C SUBROUTINE NAOBJ - TO ENTER OBJECT NAME IN DATA STRUCTURE.    00610000
C-----                                                           00611000
      SUBROUTINE NAOBJ(NA,NP,LI)                                    00612000
      COMMON /TABLES/NAME /MISC/JT,KT,LT                          00613000
      INTEGER NAME(30,5)                                           00614000

```

JT=JT+1	00615000
NAME(JT,1)=NA	00616000
NAME(JT,2)=KT+1	00617000
NAME(JT,4)=LT+1	00618000
NAME(JT,3)=NP	00619000
NAME(JT,5)=LI	00620000
KT=KT+NP	00621000
LT=LT+LI	00622000
RETURN	00623000
C	00624000
C_PRINT OBJECT NAMES ON SPRINT.	00625000
ENTRY NAMES(*)	00626000
PRINT 99	00627000
99 FORMAT(' NAMES OF OBJECTS:')	00628000
PRINT 100, (NAME(I,1),I=1,JT)	00629000
100 FORMAT(4(5X,A4))	00630000
RETURN 1	00631000
END	00632000
C	00633000
C	00634000
C-----	00635000
C SUBROUTINES NEWG & NEWO - TO DESTROY GROUPS AND OBJECTS.	00636000
C VIA THE "NEWGROUP" AND	00637000
C "NEWOBJECT" COMMANDS.	00638000
C-----	00639000
SUBROUTINE NEWG(*)	00640000
DATA IS/'SQRX'/,IC/'CUBE'/,OK/'OK '/	00641000
COMMON /TABLES/ NA(30,5),NGR(30,3),IO(300.3)/MISC/JT,KT,LT,L,N	00642000
COMMON /TABLE2/ IG(100),RG(100,5),NGT	00643000
COMMON/PLANE/NPP,NPLA(30,2),IVEC(800),MYLINE(1000.2)	00644000
PRINT 99	00645000
99 FORMAT('WISH TO DESTROY ALL GROUPS(OK)?')	00646000
700 CALL FREAD(4,&700,&701)	00647000
701 CALL APARM(OKAY,'NO ',4,&700)	00648000
IF(OKAY.NE.OK) CALL COMCAN(&33)	00649000
DO 10 I=1,L	00650000
10 NGR(I,2)=0	00651000
DO 12 I=1,N	00652000
12 IO(I,1)=0	00653000
L=0	00654000
N=0	00655000
NGT=0	00656000
PRINT 100	00657000
100 FORMAT(' DONE')	00658000
RETURN 1	00659000
C	00660000
C	00661000
ENTRY NEWO(*)	00662000
PRINT 97	00663000
97 FORMAT('WISH TO DESTROY ALL OBJECTS(OK)?')	00664000
710 CALL FREAD(4,&710,&711)	00665000
711 CALL APARM(OKAY,'NO ',4,&710)	00666000
IF(OKAY.NE.OK) CALL COMCAN(&33)	00667000
JT=2	00668000
KT=10	00669000
LT=12	00670000

NPP=31	00671000
NA(1,1)=IS	00672000
NA(2,1)=IC	00673000
PRINT 100	00674000
33 RETURN 1	00675000
END	00676000
C	00677000
C	00678000
C-----	00679000
C SUBROUTINE OINPUT - TO READ IN OBJECT INFORMATION FROM FILE	00680000
C VIA THE "OBJECT" COMMAND.	00681000
C-----	00682000
SUBROUTINE OINPUT(*)	00683000
COMMON /MISC/JT,KT,LT,L,N /DATA/X,Y,Z,LINE/TABLES/CODE	00684000
*,NQ(30,3),IOBJ(300,3)	00685000
INTEGER LINE(1000,2),CODE(30,5),LAST/'LAST'/	00686000
COMMON/PLANE/NPP,NPLA(30,2),IVEC(800),MYLINE(1000,2)	00687000
INTEGER YES/'YES '/,ANS,XX/'XXXX'/	00688000
REAL X(1000),Y(1000),Z(1000)	00689000
1 READ(5,900,END=98) NAME,NPTS,LINES,NPLANE	00690000
IF (NAME .EQ. LAST) GO TO 40	00691000
C	00692000
IF(KT .LT. 1000 .OR. LT .LT. 1000) GO TO 3	00693000
PRINT 2,NAME	00694000
2 FORMAT ('BUFFER FULL - OBJECT ',A4,' NOT ENTERED')	00695000
CALL COMCAN(&33)	00696000
3 IF(JT .LT. 30) GO TO 5	00697000
PRINT 4,NAME	00698000
4 FORMAT ('OBJECT LIST FULL - OBJECT ',A4,' NOT ENTERED')	00699000
CALL COMCAN(&33)	00700000
5 CONTINUE	00701000
C	00702000
C_EXISTING OBJECT?	00703000
DO 10 I=1,JT	00704000
IF(NAME.EQ.CODE(I,1)) GO TO 12	00705000
10 CONTINUE	00706000
C	00707000
C_ECHO VALID NAME.	00708000
PRINT 36,NAME	00709000
36 FORMAT(3X,A4)	00710000
11 K1=KT+1	00711000
L1=LT+1	00712000
CALL NAOBJ(NAME,NPTS,LINES)	00713000
READ(5,910,END=98)(X(I),Y(I),Z(I),I=K1,KT)	00714000
READ(5,920,END=98)(LINE(I,1),LINE(I,2),I=L1,LT)	00715000
C	00716000
C_ADD PLANE INFORMATION	00717000
NPLA(JT,1)=NPLANE	00718000
NPLA(JT,2)=NPP	00719000
IF (NPLANE .LE. 0) GO TO 1	00720000
DO 420 I1=1,NPLANE	00721000
READ(5,920,END=98) N1,(IVEC(I1+NPP),I1=1,N1)	00722000
IVEC(NPP)=N1	00723000
420 NPP=NPP+N1+1	00724000
GO TO 1	00725000
C	00726000

C_AN OBJECT ALREADY EXISTS WITH THAT NAME.	00727000
12 PRINT 200,NAME	00728000
200 FORMAT('OBJECT ',A4,' ALREADY EXISTS.')	00729000
2 'DO YOU WISH TO REPLACE IT(YES,NO)?'	00730000
700 CALL FREAD(4,&700,&701)	00731000
701 CALL APARM(ANS,'NO ',4,&700)	00732000
IF (ANS .EQ. YES) GO TO 19	00733000
READ(5,910,END=98) (DUM,DUM,DUM,I=1,NPTS)	00734000
READ(5,920,END=98) (DUM,DUM,I=1,LINES)	00735000
IF (NPLANE .LE. 0) GO TO 90	00736000
DO 80 I=1,NPLANE	00737000
80 READ(5,920) J1, (DUM,J=1,J1)	00738000
103 FORMAT(I4)	00739000
90 CALL COMCAN(&33)	00740000
C	00741000
C_DELETE EXISTING OBJECT.	00742000
19 CODE(I,1)=XX	00743000
DO 20 ICT=1,N	00744000
20 IF(IOBJ(ICT,1).EQ.I) IOBJ(ICT,1)=JT+1	00745000
GO TO 11	00746000
C	00747000
C_"LAST" HAS BEEN FOUND.	00748000
40 PRINT 35	00749000
35 FORMAT(' DONE')	00750000
33 RETURN 1	00751000
98 CALL ERRORS(19,0,'END ON 5',&34)	00752000
900 FORMAT(A4,3I4)	00753000
910 FORMAT(3F10.3)	00754000
920 FORMAT(20I4)	00755000
34 RETURN 1	00756000
END	00757000
C	00758000
C	00759000
C-----	00760000
C SUBROUTINE OTA - TO ADD A GROUP TO THE DATA STRUCTURE.	00761000
C-----	00762000
SUBROUTINE OTA(GRO,OBJ,IN,DX,DY,DZ,W,LEN,H,AX,ROT)	00763000
INTEGER GRO, OBJ,NAME(30,5),IOBJ(300,3),GROUP(30,3)	00764000
REAL LEN, ROBJ(300,8)	00765000
COMMON/TABLES/NAME,GRUP,IOBJ,ROBJ/MISC/JT,KT,LT,L,N,B	00766000
IO=2	00767000
4 ISW=1	00768000
5 LOBJ=GROUP(GRO,1)	00769000
IQ=OBJ-IO	00770000
IF(IQ.LE.0)GO TO 11	00771000
DO 10 I=1,IQ	00772000
10 LOBJ=IOBJ(LOBJ,2)	00773000
11 IF(OBJ-1-GROUP(GRO,2)) 13,12,14	00774000
12 IF(ISW.NE.1)CALL ERROR(1,GRO,OBJ,0.,B)	00775000
IOBJ(LOBJ,2)=N+1	00776000
LOBJ=N+1	00777000
GO TO 15	00778000
14 CALL ERROR(2,GRO,OBJ,0.,B)	00779000
RETURN	00780000
13 CONTINUE	00781000
15 GO TO(100,200,300,400,700),ISW	00782000

C		00783000
C_UPDATE THE GROUP'S OBJECT POINTER.		00784000
ENTRY OTA1(GRO,CBJ,IN,DX,DY,DZ,W,LEN,H,AX,ROT)		00785000
IO=1		00786000
GO TO 4		00787000
100 IF(IOBJ(LOBJ,1).NE.0)GO TO 110		00788000
N=N+1		00789000
GROUP(GRO,2)=GROUP(GRO,2)+1		00790000
IOBJ(LOBJ,2)=N+1		00791000
110 IOBJ(LOBJ,1)=IN		00792000
GO TO 200		00793000
C		00794000
C_ENTER THE TRANSLATION; DX,DY,DZ		00795000
ENTRY OTT(GRO,OBJ,DX,DY,DZ)		00796000
IO=1		00797000
ISW=2		00798000
GO TO 5		00799000
200 ROBJ(LOBJ,1)=DX		00800000
ROBJ(LOBJ,2)=DY		00801000
ROBJ(LOBJ,3)=DZ		00802000
GO TO(300,500,500,700),ISW		00803000
C		00804000
C_ENTER THE SCALE; W,LEN,H		00805000
ENTRY OTS (GRO,OBJ,W,LEN,H)		00806000
IO=1		00807000
ISW=3		00808000
GO TO 5		00809000
300 ROBJ(LOBJ,4)=W		00810000
ROBJ(LOBJ,5)=LEN		00811000
ROBJ(LOBJ,6)=H		00812000
GO TO(400,500,500,500,700),ISW		00813000
C		00814000
C_ENTER THE ROTATION; AX,ROT		00815000
ENTRY OTR(GRO,OBJ,AX,ROT)		00816000
IO=1		00817000
ISW=4		00818000
GO TO 5		00819000
400 ROBJ(LOBJ,7)=AX		00820000
ROBJ(LOBJ,8)=ROT		00821000
500 RETURN		00822000
C		00823000
C_DELETE AN OBJECT FROM A GROUP.		00824000
ENTRY ODEL(GRO,OBJ)		00825000
IO=1		00826000
ISW=5		00827000
GO TO 5		00828000
700 IOBJ(LOBJ,1)=-1		00829000
RETURN		00830000
END		00831000
C		00832000
C		00833000
C-----		00834000
C SUBROUTINE REPT - TO REPEAT GROUPS VIA THE "REPEAT" COMMAND.		00835000
C-----		00836000
SUBROUTINE REPT(*)		00837000
INTEGER*2 AXA(7)/'Z','X','Y','Z ','X ','Y ',' ' '//,AX		00838000

ENTRY RDSV(*)	00895000
READ(7,900,END=99) L,N,NGT	00896000
READ (7,900,END=99)((GRCUP(I,J),J=1,3),I=1,L)	00897000
READ (7,910,END=99)((ICBJ(I,J),J=1,3),(ROBJ(I,K),K=1,8)),I=1,N)	00898000
READ(7,920,END=99)((IG(I),(RG(I,J),J=1,5)),I=1,NGT)	00899000
PRINT 930	00900000
33 RETURN 1	00901000
99 CALL ERRORS(19,0,'END ON 7',&34)	00902000
900 FORMAT(3I4)	00903000
910 FORMAT(3I4,8F8.3)	00904000
920 FORMAT(I4,5F8.3)	00905000
930 FORMAT(' DONE')	00906000
34 RETURN 1	00907000
END	00908000
C	00909000
C	00910000
C-----	00911000
C SUBROUTINE USER - JUST A BUNCH OF DUMMY ROUTINES.	00912000
C-----	00913000
SUBROUTINE USER(*)	00914000
PRINT 100	00915000
100 FORMAT('THERE IS NO USER ROUTINE TO EXECUTE.')	00916000
RETURN 1	00917000
END	00918000
C	00919000
C	00920000
C-----	00921000
C LOGICAL FUNCTION VALID - TO TEST FOR ILLEGAL GROUP ENTRIES.	00922000
C-----	00923000
LOGICAL FUNCTION VALID(IGO,IGI)	00924000
INTEGER IGS(40),T(40,2),NT/0/	00925000
IGOF=IGO	00926000
K=0	00927000
IF(IGO.EQ.IGI) GO TO 50	00928000
IF(NT.EQ.0) GO TO 80	00929000
10 DO 20 I=1,NT	00930000
IF(IGO.NE.T(I,2)) GO TO 20	00931000
IF(T(I,1).EQ.IGI) GO TO 50	00932000
K=K+1	00933000
IGS(K)=T(I,1)	00934000
20 CONTINUE	00935000
IF(K.EQ.0) GO TO 80	00936000
IGO=IGS(K)	00937000
K=K-1	00938000
GO TO 10	00939000
50 CALL ERRORS(28,IGI,'LCOPING ',&90)	00940000
80 NT=NT+1	00941000
T(NT,1)=IGOF	00942000
T(NT,2)=IGI	00943000
VALID=.TRUE.	00944000
RETURN	00945000
90 VALID=.FALSE.	00946000
RETURN	00947000
END	00948000
50 CALL ERRORS(28,IGI,'LOOPING ',&90)	00940000
80 NT=NT+1	00941000

	T(NT,1)=IGOF	00942000
	T(NT,2)=IGI	00943000
	VALID=.TRUE.	00944000
	RETURN	00945000
90	VALID=.FALSE.	00946000
	RETURN	00947000
	END	00948000

CKATTN	CSECT		00001000
	ENTRY	ATTNEX	00002000
	SAVE	(14,12),T,*	00003000
	LR	12,15	00004000
	USING	CKATTN,12	00005000
	LR	15,13	00006000
	LA	13,SAVEAREA	00007000
	ST	13,8(,15)	00008000
	ST	15,4(,13)	00009000
*			00010000
	ST	15,SAVEADDR	00011000
	LM	2,3,0(1)	00012000
	L	5,0(2)	00013000
	M	4,=F'100'	00014000
	ST	4,WAITTIME	00015000
	MVC	ALTSAVE(72),0(15)	00016000
	STAX	ATTNEX	00017000
*			00018000
	L	13,4(,13)	00019000
	RETURN	(14,12),T,RC=0	00020000
*			00021000
*	EXIT ROUTINE		00022000
*			00023000
ATTNEX	SAVE	(14,12),T	00024000
	LR	12,15	00025000
	USING	ATTNEX,12	00026000
	LR	15,13	00027000
	LA	13,SAVEAREA	00028000
	ST	13,8(,15)	00029000
	ST	15,4(,13)	00030000
*			00031000
	STIMER	WAIT,BINTVL=WAITTIME	00032000
	L	2,SAVEADDR	00033000
	MVC	0(72,2),ALTSAVE	00034000
*			00035000
	LR	13,2	00036000
	RETURN	(14,12),T,RC=4	00037000
*			00038000
*	WORKING STORAGE		00039000
*			00040000
WAITTIME	DS	F	00041000
SAVEADDR	DS	F	00042000
ALTSAVE	DS	18F	00043000
SAVEAREA	DS	18F	00044000
		END	00045000

C		00001000
C		00002000
C	-----	00003000
C	SUBROUTINE COMP - TO OUTPUT ON A COMPUTEK OR A TEKTRONIX DISPLAY.	00004000
C	-----	00005000
	SUBROUTINE COMP(INT,FRA,*)	00006000
	COMMON/B/SF,XL,YL /PTS/PT,BAD	00007000
	COMMON/MISCD/IDT	00008000
	COMMON /VECTOR/DX(1000),DY(1000),ISW(1000),NMP	00009000
	LOGICAL FRA,INT,PT,BAD	00010000
	IF(.NOT.INT) GO TO 10	00011000
	IF(IDT.NE.5 .AND. IDT.NE.6)GO TO 3	00012000
1	CALL FREAD(4,&1,&2)	00013000
2	CALL RPARM(ST,0.0,&1)	00014000
3	CALL CKDER	00015000
10	DO 20 I=1,NMP	00016000
	IX=(DX(I)*SF+XL)/.008	00017000
	IY=(DY(I)*SF+YL)/.008	00018000
	IF(ISW(I).EQ.0) GO TO 21	00019000
	CALL CKDM(IX,IY,1,0)	00020000
	GO TO 20	00021000
21	CALL CKDM(IX,IY,0,0)	00022000
20	CONTINUE	00023000
	IF(.NOT.BAD)CALL CKMA(0.,750.,1)	00024000
	IF(.NOT. FRA)RETURN 1	00025000
	CALL CKTRAN	00026000
	RETURN 1	00027000
	END	00028000
C		00029000
C		00030000
C	-----	00031000
C	FUNCTION DIST - THE DISTANCE FROM POINT X,Y	00032000
C	TO LINE X1,Y1,X2,Y2 .	00033000
C	-----	00034000
	FUNCTION DIST(X,Y,X1,Y1,X2,Y2)	00035000
	IF((X1-X2).EQ.0.0)X2=X2+.01	00036000
	S=(Y1 Y2)/(X1-X2)	00037000
	B=Y1-S*X1	00038000
	DIST=(ABS(S*X-Y+B))/(SQRT(S*S+1.))	00039000
	RETURN	00040000
	END	00041000
C		00042000
C		00043000
C	-----	00044000
C	SUBROUTINE CHRO - TO OUTPUT TO THE CHROMATICS	00045000
C	-----	00046000
C		00047000
	SUBROUTINE CHRO(INIT,FRA,*)	00048000
	LOGICAL INIT,FRA	00049000
	COMMON/B/SF,XL,YL/PTS/PT,BAD	00050000
	LOGICAL BAD	00051000
	COMMON/VECTOR/DX(1000),DY(1000),ISW(1000),NMP	00052000
	COMMON/COLOUR/ICHPN(31,2)	00053000
	IF(.NOT.INIT) GO TO 10	00054000
	IF(IDT.NE.5.AND.IDT.NE.6) GO TO 3	00055000

1	CALL FREAD(4,&1,&2)	00056000
2	CALL RPARM(ST,0.0,&1)	00057000
3	CALL MATOUT	00058000
10	K=1	00059000
	DO 20 I=1,NMP	00060000
	IF(I.NE.ICHPN(K,1)) GO TO 30	00061000
	IF (IPNUM.EQ.ICHPN(K,2)) GO TO 30	00062000
	IPNUM=ICHPN(K,2)	00063000
	K=K+1	00064000
	CALL NEWCOL(IPNUM)	00065000
30	IX=(DX(I)*SF+XL)*35	00066000
	IY=(DY(I)*SF+YL)*50	00067000
	CALL MATIC(IX,IY,ISW(I))	00068000
20	CONTINUE	00069000
	IF(.NOT.BAD) CALL MATIC(0,0,-1)	00070000
	RETURN 1	00071000
	END	00072000
C		00073000
C	-----	00074000
	SUBROUTINE MATIC(IX,IY,IP)	00075000
C	-----	00076000
C		00077000
	DIMENSION IBUFF(2031)	00078000
	DATA LEN/0/,MODE/-2/,IXHC/48/,IXMC/48/,IXLC/48/,IYHC/48/,IYMC/48/,	00079000
	1IYLC/48/	00080000
	IF(MODE.NE.-2) GO TO 5	00081000
C	DISABLE CURSOR	00082000
	IBUFF(LEN+1)=01	00083000
	IBUFF(LEN+2)=75	00084000
	LEN=LEN+2	00085000
C	PUT IN PLOT MODE	00086000
	IBUFF(LEN+1)=01	00087000
	IBUFF(LEN+2)=71	00088000
	LEN=LEN+2	00089000
	MODE=0	00090000
5	CONTINUE	00091000
	IF (IP.EQ.1) GO TO 30	00092000
	IF (IP.EQ.-1) GO TO 40	00093000
C	SET POSITION MODE (CONCATENATED VECTOR)	00094000
	LEN=LEN+1	00095000
	IBUFF(LEN)=40	00096000
	MODE=1	00097000
30	CONTINUE	00098000
C	CONVERT X COORDINATES TO THREE DIGITS FOR CHROMATICS	00099000
	IXHD=IX/100	00100000
	IXMD=(IX/10)-(IXHD*10)	00101000
	IXLD=IX-((IXMD*10)+(IXHD*100))	00102000
C	OFFSET X COORDINATES	00103000
	IXHC=IXHD+48	00104000
	IXMC=IXMD+48	00105000
	IXLC=IXLD+48	00106000
C	CONVERT Y COORDINATES TO THREE DIGITS	00107000
	IYHD=IY/100	00108000
	IYMD=(IY/10)-(IYHD*10)	00109000
	IYLD=IY-((IYMD*10)+(IYHD*100))	00110000
C	OFFSET Y COORDINATES	00111000

	IYHC=IYHD+48	00112000
	IYMC=IYMD+48	00113000
	IYLC=IYLD+48	00114000
C	DRAW OR MOVE TO SPECIFIED POSITION	00115000
	IBUFF(LEN+1)=IXHC	00116000
	IBUFF(LEN+2)=IXMC	00117000
	IBUFF(LEN+3)=IXLC	00118000
	IBUFF(LEN+4)=IYHC	00119000
	IBUFF(LEN+5)=IYMC	00120000
	IBUFF(LEN+6)=IYLC	00121000
	LEN=LEN+6	00122000
C	INSURE BUFFER SPACE FOR NEXT INSTRUCTION	00123000
	IF ((LEN .LE. 2031-14) .AND. (IP .NE. -1)) RETURN	00124000
40	CONTINUE	00125000
C	OUTPUT BUFFER AND SET BACK TO ALPHA MODE	00126000
C	PLOT MODE OFF	00127000
	IBUFF(LEN+1)=21	00128000
C	SET FOREGROUND TO WHITE	00129000
	IBUFF(LEN+2)=01	00130000
	IBUFF(LEN+3)=67	00131000
	IBUFF(LEN+4)=55	00132000
C	HOME CURSOR	00133000
	IBUFF(LEN+5)=28	00134000
C	MAKE CURSOR VISIBLE	00135000
	IBUFF(LEN+6)=01	00136000
	IBUFF(LEN+7)=74	00137000
	LEN=LEN+7	00138000
	MODE=-2	00139000
	CALL USTRAN(LEN,IBUFF)	00140000
	LEN=0	00141000
	RETURN	00142000
C	ENTRY MATOUT	00143000
	IF(MODE.EQ.-2) GO TO 200	00144000
C	SET ALPHA MODE/ENABLE CURSOR	00145000
	IBUFF(LEN+1)=21	00146000
	IBUFF(LEN+2)=01	00147000
	IBUFF(LEN+3)=74	00148000
	LEN=LEN+3	00149000
	MODE=-2	00150000
C		00151000
200	CONTINUE	00152000
C	SET BACKGROUND MODE ON	00153000
	IBUFF(LEN+1)=01	00154000
	IBUFF(LEN+2)=77	00155000
	LEN=LEN+2	00156000
C	SET BACKGROUND COLOR TO BLACK	00157000
	IBUFF(LEN+1)=01	00158000
	IBUFF(LEN+2)=67	00159000
	IBUFF(LEN+3)=48	00160000
	LEN=LEN+3	00161000
C	SET BACKGROUND MODE OFF	00162000
	IBUFF(LEN+1)=01	00163000
	IBUFF(LEN+2)=78	00164000
	LEN=LEN+2	00165000
C	CLEAR SCREEN	00166000
		00167000

	LEN=LEN+1	00168000
	IBUFF(LEN)=12	00169000
C	DISABLE CURSOR	00170000
	IBUFF(LEN+1)=01	00171000
	IBUFF(LEN+2)=75	00172000
	LEN=LEN+2	00173000
C	PUT IN PLOT MODE	00174000
	IBUFF(LEN+1)=01	00175000
	IBUFF(LEN+2)=71	00176000
	LEN=LEN+2	00177000
	MODE=0	00178000
C	INSURE BUFFER SPACE FOR NEXT MOVE/DRAW INSTRUCTION	00179000
C	NEED 1 TO SET VECTOR SUBMODE	00180000
C	NEED 7 FOR MOVE	00181000
C	NEED 6 FOR DRAW	00182000
C	NEED 7 TO ENABLE CURSOR AND CANCEL PLOT MODE	00183000
	IF (LEN.LT.2031-24) GO TO 210	00184000
	CALL USTRAN(LEN,IBUFF)	00185000
	LEN=0	00186000
210	CONTINUE	00187000
	IXHC=48	00188000
	IXMC=48	00189000
	IXLC=48	00190000
	IYHC=48	00191000
	IYMC=48	00192000
	IYLC=48	00193000
	RETURN	00194000
C		00195000
	ENTRY NEWCOL(IPNUM)	00196000
C		00197000
	GO TO (300,310,320,330),IPNUM	00198000
C		00199000
C	SET FOREGROUND TO WHITE	00200000
300	IBUFF(LEN+1)=01	00201000
	IBUFF(LEN+2)=67	00202000
	IBUFF(LEN+3)=55	00203000
	LEN=LEN+3	00204000
	RETURN	00205000
C		00206000
C	SET FOREGROUND TO BLUE (ACTUALLY CYAN)	00207000
310	IBUFF(LEN+1)=01	00208000
	IBUFF(LEN+2)=67	00209000
	IBUFF(LEN+3)=51	00210000
	LEN=LEN+3	00211000
	RETURN	00212000
C		00213000
C	SET FOREGROUND TO GREEN	00214000
320	IBUFF(LEN+1)=01	00215000
	IBUFF(LEN+2)=67	00216000
	IBUFF(LEN+3)=50	00217000
	LEN=LEN+3	00218000
	RETURN	00219000
C		00220000
C	SET FOREGROUND TO RED	00221000
330	IBUFF(LEN+1)=01	00222000
	IBUFF(LEN+2)=67	00223000

IBUFF(LEN+3)=52	00224000
LEN=LEN+3	00225000
RETURN	00226000
END	00227000
C	00228000
C-----	00229000
C SUBROUTINE GT40 -	00230000
C DEVICE SUPPORT FOR SAI TERMINAL (NEWD) AND DUMMY ROUTINES FOR	00231000
C THE DEC GT40	00232000
C-----	00233000
C	00234000
SUBROUTINE GT40(INIT,FRA,*)	00235000
LOGICAL INIT,FRA	00236000
COMMON /B/ SF,XL,YL /PTS/PT,BAD	00237000
LOGICAL BAD	00238000
COMMON /MISC/ IDT	00239000
COMMON /VECTOR/DX(1000),DY(1000),ISW(1000),NMP	00240000
RETURN 1	00241000
ENTRY NEWD(INIT,FRA,*)	00242000
IF (.NOT.INIT) GO TO 10	00243000
IF (IDT.NE.5 .AND.IDT.NE.6) GO TO 3	00244000
1 CALL FREAD(4,&1,&2)	00245000
2 CALL RPARM(ST,0.0,&1)	00246000
3 CALL BUFF3A(1)	00247000
10 DO 20 I=1,NMP	00248000
IX=(DX(I)*SF+XL)*60	00249000
IY=(DY(I)*SF+YL)*60	00250000
CALL BUFF3(IX,IY,ISW(I))	00251000
20 CONTINUE	00252000
IF (.NOT.BAD) CALL BUFF3(0,2,-1)	00253000
RETURN 1	00254000
ENTRY NEWIN(INIT,FRA,*)	00255000
ENTRY GT40IN(INIT,FRA,*)	00256000
RETURN 1	00257000
END	00258000
C	00259000
C-----	00260000
C SUBROUTINE BUFF3 IS USED TO ROUTE OUTPUT TO THE SAI TERMINAL	00261000
C-----	00262000
C	00263000
SUBROUTINE BUFF3(IX,IY,IP)	00264000
DIMENSION IBUFF(2031)	00265000
DATA LEN/0/,MODE/-2/,IXH/32/,IXL/32/,IYH/32/,IYL/32/	00266000
C DATA FRAME FORMAT (USE 2 BYTE DATA FRAME)	00267000
C OFFSET BOTH BYTES 32 TO MISS STANDARD CONTROL CHARACTERS	00268000
C BYTE ONE CONTAINS LOW ORDER BITS (MOD(VAL,32))	00269000
C BYTE TWO CONTAINS HIGH ORDER BITS (VAL/32)	00270000
C STANDARD CONTROL FRAME FORMAT	00271000
C NUMBER BETWEEN 0 AND 31	00272000
C SPECIAL CONTROL FRAME FORMAT	00273000
C OFFSET ALL BYTES 32 TO MISS STANDARD CONTROL CHARACTERS	00274000
C ADD 32 FOR 1 BYTE CONTROL FRAME	00275000
C ADD 40 FOR 2 BYTE CONTROL FRAME	00276000
C MODE DEFINITIONS	00277000
C -2 ALPHA MODE	00278000
C -1 SCREEN IN FRAMING (GRAPHICS) MODE BUT NOT POSTIONED	00279000

C	0 POSITION MODE	00280000
C	1 VECTOR OR DRAW MODE	00281000
C		00282000
	IF (MODE .NE. -2) GO TO 5	00283000
C	SET FRAMING (GRAPHICS) MODE/DISABLE ALPHA CURSOR	00284000
C	PUT 'ESCAPE' CHARACTER IN BUFFER	00285000
	IBUFF(LEN+1)=27	00286000
C	TURN OFF ALPHA CURSOR	00287000
	IBUFF(LEN+2)=72	00288000
	IBUFF(LEN+3)=36	00289000
	LEN=LEN+3	00290000
	MODE=-2	00291000
5	CONTINUE	00292000
	IF (IP .EQ. 1) GO TO 10	00293000
C	SET POSITION MODE	00294000
	LEN=LEN+1	00295000
	IBUFF(LEN)=67	00296000
	MODE=0	00297000
	GO TO 30	00298000
10	IF (MODE .EQ. 1) GO TO 30	00299000
	IF (MODE .EQ. 0 .OR. LEN.GT. 5) GO TO 20	00300000
C	SET POSITION MODE/MOVE TO LAST KNOWN POSITION	00301000
	IBUFF(LEN+1)=67	00302000
	MODE=0	00303000
	IBUFF(LEN+2)=IXL	00304000
	IBUFF(LEN+3)=IXH	00305000
	IBUFF(LEN+4)=IYL	00306000
	IBUFF(LEN+5)=IYH	00307000
	LEN=LEN+5	00308000
20	CONTINUE	00309000
C	SET VECTOR MODE	00310000
	LEN=LEN+1	00311000
	IBUFF(LEN)=68	00312000
	MODE=1	00313000
30	CONTINUE	00314000
C	DRAW OR MOVE TO SPECIFIED POSITION	00315000
	IXH=32 + IX/32	00316000
	IXL=32 + (IX-IX/32*32)	00317000
	IYH=32 + IY/32	00318000
	IYL=32 + (IY-IY/32*32)	00319000
	IBUFF(LEN+1)=IXL	00320000
	IBUFF(LEN+2)=IXH	00321000
	IBUFF(LEN+3)=IYL	00322000
	IBUFF(LEN+4)=IYH	00323000
	LEN=LEN+4	00324000
C	INSURE BUFFER SPACE FOR NEXT INSTRUCTION	00325000
C	BUFFER LENGTH 2031	00326000
C	ALREADY IN FRAMENIG MODE (DON'T NEED ESCAPE)	00327000
C	NEED 4 OR 5 FOR MOVE/DRAW	00328000
C	NEED 3 TO SET ALPHA MODE/ENABLE CURSOR	00329000
C	OUTPUT ALSO IF IP IS -1 (SIGNAL TO OUTPUT BUFFER)	00330000
	IF ((LEN .LE. 2031-8) .AND. (IP .NE. -1)) RETURN	00331000
C	OUTPUT BUFFER	00332000
	MODE=-2	00333000
	IBUFF(LEN+1)=73	00334000
	IBUFF(LEN+2)=36	00335000

	IBUFF(LEN+3)=64	00336000
	LEN=LEN+3	00337000
	CALL USTRAN(LEN,IBUFF)	00338000
	LEN=0	00339000
	RETURN	00340000
C	CLEAR SCREEN	00341000
	ENTRY BUFF3A	00342000
	IF (MODE .EQ. -2) GO TO 200	00343000
C	ENABLE ALPHA CURSOR/SET ALPHA MODE	00344000
	IBUFF(LEN+1)=73	00345000
	IBUFF(LEN+2)=36	00346000
	IBUFF(LEN+3)=64	00347000
	MODE=-2	00348000
	LEN=LEN+3	00349000
200	CONTINUE	00350000
C	CLEAR SCREEN	00351000
	LEN=LEN+1	00352000
	IBUFF(LEN)=12	00353000
C	INSURE BUFFER SPACE FOR NEXT MOVE/DRAW INSTRUCTION	00354000
C	BUFFER LENGTH 2031	00355000
C	NEED 3 TO SET FRAMING (GRAPHICS) MODE/DISABLE CURSOR	00356000
C	NEED 5 FOR MOVE INSTRUCTION	00357000
C	NEED 5 FOR DRAW INSTRUCTION	00358000
C	NEED 3 TO ENABLE CURSOR/SET ALPHA MODE	00359000
	IF (LEN .LT. 2031-16) GO TO 210	00360000
	CALL USTRAN(LEN,IBUFF)	00361000
	LEN=0	00362000
210	CONTINUE	00363000
	IXH=32	00364000
	IXL=32	00365000
	IYH=32	00366000
	IYL=32	00367000
	RETURN	00368000
	ENTRY SAICDE(N,IC1,IC2)	00369000
	WRITE(3,987) N,IC1,IC2	00370000
987	FORMAT(3I6)	00371000
	IBUFF(LEN+1)=64	00372000
	IBUFF(LEN+2)=27	00373000
	IBUFF(LEN+3)=IC1	00374000
	IF(N .LE.1) GO TO 300	00375000
	IBUFF(LEN+4)=IC2	00376000
300	LEN=LEN+2+N	00377000
	RETURN	00378000
	ENTRY SAIDAT(IX)	00379000
	IXH=32+IX/32	00380000
	IXL=32+(IX-IX/32*32)	00381000
	IBUFF(LEN+1)=IXH	00382000
	IBUFF(LEN+2)=IXL	00383000
	LEN=LEN+2	00384000
	IBUFF(LEN+1)=64	00385000
	RETURN	00386000
	END	00387000
C		00388000
C		00389000
C	-----	00390000
C	SUBROUTINE UMPL ~ TO OUTPUT ON THE 30" CALCOMP PLOTTER.	00391000

C-----		00392000
	SUBROUTINE UMPL(INT,FRA,*)	00393000
	COMMON/VECTOR/DX(1000),DY(1000),ISW(1000),NMP	00394000
	COMMON/B/SF,XL,YL	00395000
	COMMON/COLOUR/ICHPN(31,2)	00396000
	LOGICAL INT,FRA	00397000
	IF(.NOT.INT)GO TO 10	00398000
	CALL PLDNO(8)	00399000
10	K=1	00400000
	IPCOL=0	00401000
	DO 20 I=1,NMP	00402000
	IPNUM=0	00403000
	IF(I.NE.ICHPN(K,1)) GO TO 30	00404000
	IF(IPCOL.EQ.ICHPN(K,2)) GO TO 30	00405000
	IPNUM=ICHPN(K,2)	00406000
	K=K+1	00407000
	IPCOL=IPNUM	00408000
30	IF(ISW(I).EQ.0)GO TO 21	00409000
	CALL PENDNS(DX(I)*SF+XL,DY(I)*SF+YL,IPNUM)	00410000
	GO TO 20	00411000
21	CALL PENUPS(DX(I)*SF+XL,DY(I)*SF+YL,IPNUM)	00412000
20	CONTINUE	00413000
	IF(FRA)CALL PLTEND	00414000
	RETURN 1	00415000
	END	00416000
	IPNUM=0	00403000
	IF(I.NE.ICHPN(K,1)) GO TO 30	00404000
	IF(IPCOL.EQ.ICHPN(K,2)) GO TO 30	00405000
	IPNUM=ICHPN(K,2)	00406000
	K=K+1	00407000
	IPCOL=IPNUM	00408000
30	IF(ISW(I).EQ.0) GO TO 21	00409000
	CALL PENDNS(DX(I)*SF+XL,DY(I)*SF+YL,IPNUM)	00410000
	GO TO 20	00411000
21	CALL PENUPS(DX(I)*SF+XL,DY(I)*SF+YL,IPNUM)	00412000
20	CONTINUE	00413000
	IF(FRA)CALL PLTEND	00414000
	RETURN 1	00415000
	END	00416000

```

C                                     00001000
C                                     00002000
C----- 00003000
C SUBROUTINE BLDMAT - INITIATES PROCESS FOR GROUP(IDG). 00004000
C                                     CONSTRUCTS PROPER MATRICES AND PERFORMS 00005000
C                                     COORDINATE MULTIPLICATION. 00006000
C----- 00007000
      SUBROUTINE BLDMAT(IDG) 00008000
      LOGICAL SLT 00009000
      COMMON /TABLES/NAME,GROUP,IOBJ,R /TABLE2/IG,RG,NGT 00010000
1 /MATRIX/DRAT,MATG,MATO,MAT/POINTS/POINT /MISC/IDT,ID,SF 00011000
      COMMON/DATA/X(1000),Y(1000),Z(1000),LINE(1000.2) 00012000
      COMMON /VECTOR/DX(1000),DY(1000),ISB(1000),NSTOR,GOOD 00013000
      COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF, 00014000
1NEXTTP,NEXTTL,ITP,H1,NTRAT 00015000
      LOGICAL INSAV,NODRAT,HID,H1 00016000
      INTEGER NAME(30,5),GROUP(30,3),IOBJ(300,3),IG(100) 00017000
      REAL R(300,8),RG(100,5),DRAT(4,4),MATG(4,4),MAT(4,4),POINT(350,4) 00018000
      REAL P(4),SAVE(200),MATGG(4,4),MATO(4,4),T(4,4) 00019000
      LOGICAL SW,GOOD,SWG 00020000
      INTEGER CT,COUNT 00021000
      SWG=.FALSE. 00022000
      COUNT=0 00023000
10 SW=.FALSE. 00024000
      N1=1 00025000
      IF(NGT.EQ.0)GO TO 50 00026000
      DO 5 I=1,NGT 00027000
      IF(IG(I).NE.IDG) GO TO 5 00028000
      SW=.TRUE. 00029000
      N1=I 00030000
      GO TO 100 00031000
5 CONTINUE 00032000
      N1=NGT+1 00033000
      GO TO 50 00034000
100 DO 25 II=N1,NGT 00035000
      IF(IG(II).EQ.IDG) GO TO 26 00036000
25 CONTINUE 00037000
      GO TO 210 00038000
26 CALL MAT1(RG(II,1),RG(II,2),RG(II,3),RG(II,4),RG(II,5),MATG) 00039000
      IF(.NOT.SWG)GO TO 27 00040000
      CALL CCMB(MATG,MATGG,T) 00041000
      CALL SWAP(T,MATG) 00042000
27 N1=II+1 00043000
      GO TO 51 00044000
50 IF(SWG)CALL SWAP(MATGG,MATG) 00045000
51 J=GROUP(IDG,1) 00046000
      JA=GROUP(IDG,2) 00047000
      ISS=1 00048000
C 00049000
C_PROCESS ALL OBJECTS OF GROUP "IDG". 00050000
55 DO 200 I=ISS,JA 00051000
      IF(IOBJ(J,1).LT.1) GO TO 99 00052000
      IF(IOBJ(J,1).GT.100)GO TO 500 00053000
      CALL MATF(R(J,1),R(J,2),R(J,3),R(J,4),R(J,5),R(J,6),R(J,7),R(J,8), 00054000
1MATO) 00055000

```

C	R IS ROBJ	00056000
	IF(.NOT.SW.AND..NOT.SWG) GO TO 1000	00057000
	CALL COMB(MATO,MATG,T)	00058000
	CALL SWAP(T,MATO)	00059000
1000	IF(.NOT.NODRAT)GO TO 1001	00060000
	CALL SWAP(MATO,MAT)	00061000
	GO TO 39	00062000
1001	CALL COMB(MATO,DRAT,MAT)	00063000
39	JJ=IOBJ(J,1)	00064000
	NS=NAME(JJ,2)	00065000
	NF=NS+NAME(JJ,3)-1	00066000
	K=1	00067000
C		00068000
C_SAVE	POINTS FROM OBJECT JJ.	00069000
	DO 40 M=NS,NF	00070000
	POINT(K,1)=X(M)	00071000
	POINT(K,2)=Y(M)	00072000
	POINT(K,3)=Z(M)	00073000
	POINT(K,4)=1.	00074000
40	K=K+1	00075000
	K=K-1	00076000
C		00077000
C_EXPAND	POINTS.	00078000
	DO 80 IN=1,K	00079000
	DO 79 J1=1,4	00080000
	F=0.0	00081000
	DO 70 K1=1,4	00082000
70	F=POINT(IN,K1)*MAT(K1,J1)+F	00083000
79	P(J1)=F	00084000
	DO 80 J3=1,4	00085000
	POINT(IN,J3)=P(J3)	00086000
80	CONTINUE	00087000
C		00088000
C_ADD	OBJECT "JJ" TO THE OUTPUT BUFFER.	00089000
	IF(.NOT.NODRAT)GO TO 98	00090000
	CALL SSAVE(JJ,K,&99)	00091000
98	CALL LOUT(JJ)	00092000
99	J=IOBJ(J,2)	00093000
200	CONTINUE	00094000
C		00095000
	IF(N1.LE.NGT) GO TO 100	00096000
210	IF(COUNT.EQ.0)RETURN	00097000
	GO TO 600	00098000
500	CT=COUNT*20+1	00099000
	COUNT=COUNT+1	00100000
	SAVE(CT)=IDG	00101000
	SAVE(CT+1)=N1	00102000
	SAVE(CT+2)=J	00103000
	SAVE(CT+3)=I	00104000
	SAVE(CT+4)=0.	00105000
	IF(.NOT.SWG.AND..NOT.SW)GO TO 555	00106000
	IC=CT+3	00107000
	DO 550 IS=1,4	00108000
	DO 550 JS=1,4	00109000
	IC=IC+1	00110000
550	SAVE(IC)=MATG(IS,JS)	00111000

555	CALL MATF(R(J,1),R(J,2),R(J,3),R(J,4),R(J,5),R(J,6),R(J,7),	00112000
	1R(J,8),MATGG)	00113000
	IF(.NOT.SWG.AND..NOT.SW)GO TO 560	00114000
	CALL COMB(MATGG,MATG,T)	00115000
	CALL SWAP(T,MATGG)	00116000
560	SWG=.TRUE.	00117000
	IDG=IOBJ(J,1)-100	00118000
	GO TO 10	00119000
600	COUNT=COUNT-1	00120000
	SLT=.FALSE.	00121000
	CT=COUNT*20+1	00122000
	IC=CT+3	00123000
	IDG=SAVE(CT)	00124000
	N1=SAVE(CT+1)	00125000
	ISS=SAVE(CT+3)+1	00126000
601	J=SAVE(CT+2)	00127000
	IF(SLT) GO TO 602	00128000
	J=IOBJ(J,2)	00129000
	JA=GROUP(IDG,2)	00130000
602	DO 650 IS=1,4	00131000
	DO 650 JS=1,4	00132000
	IC=IC+1	00133000
650	MATG(IS,JS)=SAVE(IC)	00134000
	IF(SLT) GO TO 653	00135000
	SW=.TRUE.	00136000
	IF(MATG(1,1).EQ.0.)SW=.FALSE.	00137000
	IF(COUNT.EQ.0)SWG=.FALSE.	00138000
651	IF(COUNT.EQ.0.AND.N1.GT.NGT.AND.ISS.GT.JA)RETURN	00139000
	IF(N1.GT.NGT.AND.ISS.GT.JA) GO TO 600	00140000
	IF(N1.LE.NGT.AND.ISS.GT.JA) GO TO 652	00141000
	GO TO 55	00142000
652	IF(COUNT.EQ.0) GO TO 100	00143000
	CT=(COUNT-1)*20+1	00144000
	IC=CT+3	00145000
	SLT=.TRUE.	00146000
	GO TO 501	00147000
653	SLT=.FALSE.	00148000
	CALL MATF(R(J,1),R(J,2),R(J,3),R(J,4),R(J,5),R(J,6),R(J,7),	00149000
	*R(J,8),MATGG)	00150000
	IF(MATG(1,1).EQ.0.) GO TO 100	00151000
	CALL COMB(MATGG,MATG,T)	00152000
	CALL SWAP(T,MATGG)	00153000
	GO TO 100	00154000
	END	00155000
C		00156000
C		00157000
C-----		00158000
C SUBROUTINE COMB - TO MULTIPLY TWO 4X4 MATRICES. AXB=C.		00159000
C-----		00160000
	SUBROUTINE COMB(A,B,C)	00161000
	REAL A(4,4),B(4,4),C(4,4)	00162000
	DO 5 I=1,4	00163000
	DO 5 J=1,4	00164000
5	C(I,J)=0.0	00165000
	DO 10 M=1,4	00166000
	DO 10 J=1,4	00167000

```

DO 20 K=1,4                                00168000
20  C(M,J)= C(M,J)+B(K,J)*A(M,K)           00169000
10  CONTINUE                                00170000
    RETURN                                  00171000
    END                                     00172000
C                                           00173000
C-----                                00174000
C SUBROUTINE DRAW - CONTROL PROGRAM FOR DRAWING GROUPS. 00175000
C-----                                00176000
    SUBROUTINE DRAW(*)                      00177000
    COMMON /DATA/ XX(1000),YY(1000),ZZ(1000),LINE(1000,2) 00178000
    COMMON/PLANE/NNP,NPLA(30,2),IVEC(800),MYLINE(1000,2) 00179000
    COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF, 00180000
1    NEXTP,NEXTL,ITP,H1,NTRAT              00181000
    LOGICAL INSAV,NODRAT,HID,H1,NGRO        00182000
    COMMON /PTS/ PT,BAD,BACK                00183000
    COMMON /B/ XST,XLOC,YLOC                00184000
    COMMON/COLOUR/ICHPN(31,2),ICOL(30)      00185000
    INTEGER GC,IPNUM                        00186000
    COMMON /MISC/JT,KT,LT,L,N,B /TABLES/NAME,GROUP /PDATA/D,OPT,RX1,RX00187000
12,R1,R2 /MISC/IDT,IDEV,SF,FRA /VECTOR/X(1000),Y(1000),IS(1000), 00188000
1KS,G                                       00189000
    COMMON/JOHN/INIT                        00190000
    LOGICAL*4 INIT                          00191000
    DIMENSION M(30),XS(1000),YS(1000),INS(1000),DTYPE(5),DEV(9) 00192000
    DATA DTYPE/'ROTA','PERS','STER','SOLA','MOVI'/      00193000
    DATA DEV/'STOR','GT40','UMPL','NEW ','CHRO','TEK ', 00194000
1    'GT40','PLOT','SAI '/                 00195000
    DATA BLANK/' '/                        00196000
    INTEGER OP,GROUP(30,3),NAME(30,5),IBLANK/' '/        00197000
    LOGICAL*1 DSC, DS/F/,DV/F/              00198000
    LOGICAL*1 BACK                          00199000
    LOGICAL OPT,G,FRA                       00200000
    LOGICAL NMOV/F/,PT,BAD                  00201000
    LOGICAL CCLRET/F/                       00202000
    REAL YES/'YES '/,YYY/'Y '/             00203000
C                                           00204000
C_INITIALIZE VARIABLES.                    00205000
    SF=1.                                   00206000
    INIT=.TRUE.                             00207000
    FRA=.TRUE.                              00208000
    NODRAT=.FALSE.                          00209000
    DSC=.FALSE.                             00210000
    GO TO 10                                00211000
C                                           00212000
C_MAIN PROMPT OF DRAWING PROGRAM.          00213000
    ENTRY DRAW2(*)                          00214000
300 GO TO 3301                              00215000
301 KS=0                                    00216000
3301 CONTINUE                              00217000
    PRINT 8                                 00218000
    8 FORMAT('/OPTION?')                     00219000
    G=.TRUE.                                00220000
    CALL CKATTN(3,&301)                      00221000
700 CALL FREAD(4,&700,&701)                 00222000
701 CALL IPARM(OP,C,&700)                   00223000

```

IF(OP.EQ.0)RETURN 1	00224000
IF(OP.GT.18) CALL ERRORS(25,0,'BAD OPT#',&300)	00225000
GO TO(310,302,303,36,51,306,307,10,308,309,403	00226000
*,200,299,1700,1800,1900,2000,3500),CP	00227000
C	00228000
C_OPTION 1 - PRINT THE LIST OF OPTIONS.	00229000
310 PRINT 9	00230000
9 FORMAT(' 0=RETURN TO COMMAND STATE'//	00231000
1 ' 1=PRINT THIS LIST'//	00232000
2 ' 2=SAVE DRAWING AS BACKGROUND'//	00233000
3 ' 3=CHANGE OUTPUT DEVICE'//	00234000
4 ' 4=CHANGE DRAWING TYPE'//	00235000
5 ' 5=CHANGE DRAWING PARAMETERS'//	00236000
6 ' 6=OUTPUT CURRENT DRAWING ON ANOTHER DEVICE'//	00237000
7 ' 7=DRAW DIFFERENT GROUP(S) AT THE SAME VIEW'//	00238000
8 ' 8=NEW DRAWING'//	00239000
9 ' 9=USE SAVED BACKGROUND'//	00240000
* ' 10=SUPPRESS FRAME'//	00241000
1 ' 11=SET DRAWING PROPORTIONS'//	00242000
2 ' 12=RECALCULATE SAME DRAWING'//	00243000
3 ' 13=REDRAW SAME DRAWING'//	00244000
4 ' 14=OUTPUT SAME MOVIE ON ANOTHER DEVICE'//	00245000
5 ' 15=REMOVE HIDDEN-LINES'//	00246000
6 ' 16=DON'T REMOVE HIDDEN-LINES'//	00247000
7 ' 17=SAVE OBJECT ON 2'//	00248000
8 ' 18=CHANGE GROUP COLORS'//	00249000
GO TO 300	00250000
C	00251000
C_OPTION 2 - SAVE DRAWING AS BACKGROUND.	00252000
302 DO 320 I=1,KS	00253000
XS(I)=X(I)	00254000
YS(I)=Y(I)	00255000
320 INS(I)=IS(I)	00256000
NSS=KS	00257000
PRINT 690	00258000
GO TO 300	00259000
C	00260000
C_OPTION 3 - CHANGE OUTPUT DEVICE.	00261000
303 DS=.FALSE.	00262000
DSC=.TRUE.	00263000
GC TO 32	00264000
C	00265000
C_OPTION 4 - CHANGE DRAWING TYPE.	00266000
36 DSC=.TRUE.	00267000
34 DV=.TRUE.	00268000
PRINT 6	00269000
6 FORMAT('ENTER DRAWING TYPE:')	00270000
710 CALL FREAD(4,&710,&711)	00271000
711 CALL APARM(TYPE,'NO ',4,&710)	00272000
DO 37 I=1,5	00273000
IF(TYPE.EQ.DTYPE(I)) GO TO 50	00274000
37 CONTINUE	00275000
IF(TYPE.NE.BLANK)GO TO 62	00276000
I=1	00277000
GO TO 50	00278000
62 CALL ERROR(13,IDEV,IDT,0.0,TYPE)	00279000

GO TO 36	00280000
50 IDT=I	00281000
IF(NGRO.AND.HID)GO TO 520	00282000
C	00283000
C_OPTION 5 - CHANGE DRAWING PARAMETERS.	00284000
51 GO TO (100,110,120,130,140,140),IDT	00285000
100 CALL ROT(&200)	00286000
110 CALL PERS(&200)	00287000
120 CALL STERO(M,NOUT,&300)	00288000
130 CALL SCLAR(&200)	00289000
140 CALL MOVIE(M,NOUT,&300)	00290000
C	00291000
C_OPTION 6 - OUTPUT CURRENT DRAWING ON ANOTHER DEVICE.	00292000
306 DS=.TRUE.	00293000
DSC=.FALSE.	00294000
IDEVT=IDEV	00295000
GO TO 32	00296000
C	00297000
C_OPTION 7 - DRAW DIFFERENT GROUPS AT THE SAME VIEW.	00298000
307 DS=.TRUE.	00299000
HID=.FALSE.	00300000
GO TO 11	00301000
C	00302000
C_OPTION 8 - NEW DRAWING.	00303000
10 DS=.FALSE.	00304000
HID=.FALSE.	00305000
CALL CKATTN(3,&300)	00306000
11 PRINT 1	00307000
1 FORMAT('ENTER GROUP NUMBERS TO BE DRAWN(0=ALL):')	00308000
720 CALL FREAD(4,&720,&721)	00309000
721 DC 722 I=1,30	00310000
722 CALL IPARM(M(I),0,&720)	00311000
NGRO=.TRUE.	00312000
IF(M(1).NE.0) GO TO 21	00313000
NOUT=L	00314000
DO 17 I=1,NOUT	00315000
17 M(I)=I	00316000
GO TO 39	00317000
21 DO 20 I=1,30	00318000
IF(M(I).EQ.0) GO TO 30	00319000
IF(GROUP(M(I),3).EQ.0) GO TO 25	00320000
GO TO 20	00321000
25 PRINT 5, M(I)	00322000
5 FORMAT('GROUP',I3,' IS INACTIVE. WISH TO ACTIVATE IT(YES,NO)?')	00323000
730 CALL FREAD(4,&730,&731)	00324000
731 CALL APARM(ANSA,'NO ',4,&730)	00325000
IF(ANSA.NE.YES)GO TO 20	00326000
CALL ACTIVE(&20)	00327000
20 CONTINUE	00328000
30 NOUT=I-1	00329000
39 IF(IDEV.NE.3.AND.IDEV.NE.5) GO TO 40	00330000
COLRET = .TRUE.	00331000
GO TO 3000	00332000
3240 COLRET = .FALSE.	00333000
40 IF(NGRO.AND.HID)GO TO 520	00334000
IF(DS) GO TO 200	00335000

31	IF(DSC) GO TO 36	00336000
	IF(DV) GO TO 34	00337000
32	PRINT 3	00338000
	3 FORMAT('ENTER DEVICE NAME:')	00339000
740	CALL FREAD(4,&740,&741)	00340000
741	CALL APARM(DEVI,' ',4,&740)	00341000
	DO 35 J=1,9	00342000
	IF(DEVI.NE.DEV(J)) GO TO 35	00343000
61	IDEV=MOD(J-1,5)+1	00344000
	IF(IDEV.NE.3.AND.IDEV.NE.5) GO TO 3230	00345000
	IF(OP.EQ.6.AND.(IDEVT.EQ.3.OR.IDEVT.EQ.5)) GO TO 3230	00346000
	GO TO 3000	00347000
3230	CONTINUE	00348000
	IF(DS) GO TO 330	00349000
	IF(DSC)PRINT 690	00350000
	IF(DSC) GO TO 300	00351000
	GO TO 36	00352000
35	CONTINUE	00353000
	IF(DEVI.NE.BLANK)GO TO 60	00354000
	J=1	00355000
	GO TO 61	00356000
60	CALL ERROR(12,M(I),NOUT,0.0,DEVI)	00357000
	GO TO 32	00358000
C		00359000
	C_OPTION 9 - USE SAVED BACKGROUND.	00360000
308	BACK=.TRUE.	00361000
	PRINT 690	00362000
690	FORMAT('DONE')	00363000
	GO TO 300	00364000
C		00365000
	C_OPTION 10 - SUPPRESS FRAME.	00366000
309	FRA=.FALSE.	00367000
	PRINT 690	00368000
	GO TO 300	00369000
C		00370000
	C_OPTION 11 - SET DRAWING PROPORTIONS.	00371000
403	PRINT 404	00372000
404	FORMAT('ENTER WIDTH/HEIGHT RATIO(EG. 1.25):')	00373000
750	CALL FREAD(4,&750,&751)	00374000
751	CALL RPARM(RATIO,0.0,&750)	00375000
	IF(RATIO.LE.0.)GO TO 900	00376000
	RY1=RY1/RATIO	00377000
	RY2=RY2/RATIO	00378000
	PRINT 690	00379000
	GO TO 300	00380000
C		00381000
	C_OPTION 12 - RECALCULATE SAME DRAWING.	00382000
200	IF(HID)CALL HIDLI(&251)	00383000
	DO 201 I=1,NOUT	00384000
	IF(GROUP(M(I),3).NE.0) GO TO 202	00385000
201	CONTINUE	00386000
202	ICHPN(I,1)=1	00387000
	DO 250 I=1,NOUT	00388000
	IF(GROUP(M(I),3).EQ.0) GO TO 250	00389000
	CALL BLDMAT(M(I))	00390000
	ICHPN(I+1,1)=KS+1	00391000

	<u>ICHPN(I,2)=ICOL(M(I))</u>	00392000
250	CONTINUE	00393000
251	IF(BACK) GO TO 400	00394000
C		00395000
	C_OPTION 13 - REDRAW SAME DRAWING.	00396000
299	CALL PUTOUT	00397000
	<u>IF(OP.EQ.6) IDEV=IDEVT</u>	00398000
	<u>GO TO 300</u>	00399000
C		00400000
	C_OPTION 14 - OUTPUT CURRENT MOVIE ON ANOTHER DEVICE.	00401000
1700	NMCV=.TRUE.	00402000
	GO TO 306	00403000
C		00404000
	C_OPTION 15 - REMOVE HIDDEN-LINES.	00405000
1800	HID=.TRUE.	00406000
	PRINT 1820	00407000
1820	FORMAT('WISH TO HAVE DASHED-LINES(YES,NO)?')	00408000
760	CALL FREAD(4,&760,&761)	00409000
761	CALL APARM(ANS,'NO ',4,&760)	00410000
	NTRAT=0	00411000
	IF(ANS.EQ.YES)NTRAT=1	00412000
	PRINT 690	00413000
	IF(.NOT.NGRO)GO TO 300	00414000
	INSAV=.TRUE.	00415000
	NODRAT=.TRUE.	00416000
	DO 1801 I=1,NCUT	00417000
	IF(GROUP(M(I),3).EQ.0)GO TO 1801	00418000
	CALL BLDMAT(M(I))	00419000
1801	CONTINUE	00420000
	NODRAT=.FALSE.	00421000
	H1=.TRUE.	00422000
	NGRO=.FALSE.	00423000
	GO TO 300	00424000
C		00425000
	C_OPTION 16 - DON'T REMOVE HIDDEN-LINES.	00426000
1900	HID=.FALSE.	00427000
	PRINT 690	00428000
	GO TO 300	00429000
C		00430000
	C_OPTION 17 - SAVE OBJECT CN 9.	00431000
2000	PRINT 2010	00432000
2010	FORMAT('ENTER OBJECT NAME:')	00433000
770	CALL FREAD(4,&770,&300)	00434000
	CALL APARM(NA1,' ',4,&770)	00435000
	IF(NA1.EQ.IBLANK)GO TO 2000	00436000
	DO 2020 I=1,JT	00437000
	IF (NA1 .EQ. NAME(I,1)) GO TO 2040	00438000
2020	CONTINUE	00439000
	PRINT 2030	00440000
2030	FORMAT(' OBJECT NOT FOUND')	00441000
	GO TO 300	00442000
2040	NPOINT=NAME(I,3)	00443000
	NLINE=NAME(I,5)	00444000
	NPLANE=NPLA(I,1)	00445000
	WRITE(2,2050) NA1, NPOINT, NLINE, NPLANE	00446000
2050	FORMAT(A4,3I4)	00447000

	J1=NAME(I,2)	00448000
	J2=J1+NPOINT-1	00449000
	WRITE(2,2060) (XX(J),YY(J),ZZ(J), J=J1,J2)	00450000
2060	FORMAT(3F10.3)	00451000
	J1=NAME(I,4)	00452000
	J2=J1+NLINE-1	00453000
	WRITE(2,2070) (LINE(J,1),LINE(J,2), J=J1,J2)	00454000
2070	FORMAT(20I4)	00455000
	IF (NPLANE .LE. 0) GO TO 2090	00456000
	J1=NPLA(I,2)	00457000
	DO 2080 K=1,NPLANE	00458000
	J2=J1+IVEC(J1)	00459000
	WRITE(2,2070) (IVEC(J), J=J1,J2)	00460000
2080	J1=J2+1	00461000
2090	WRITE(2,2100)	00462000
2100	FORMAT('LAST')	00463000
	PRINT 690	00464000
	GO TO 300	00465000
C		00466000
C	OPTION 18 - CHANGE GROUP COLORS	00467000
C		00468000
	3500 IF(IDEV.EQ.3.OR.IDEV.EQ.5) GO TO 3040	00469000
	PRINT 3501	00470000
	3501 FORMAT('OUTPUT DEVICE MUST BE UMPL,PLOT, OR CHRO'/	00471000
	1 'FOR COLOR OUTPUT.')	00472000
	GO TO 300	00473000
	3000 PRINT 3010	00474000
	3010 FORMAT('DO YOU WISH TO USE COLOR? (Y/N)')	00475000
	3020 CALL FREAD (4,&3020,&3030)	00476000
	3030 CALL APARM(ANSA,'N ',4,&3020)	00477000
	IF(ANSA.EQ.YES.OR.ANSA.EQ.YYY) GO TO 3040	00478000
	DO 3045 I=1,NOUT	00479000
	3045 ICOL(M(I))=1	00480000
	IF(COLRET) GO TO 3240	00481000
	GO TO 3230	00482000
	3040 DO 3050 I=1,NOUT	00483000
	3050 ICOL(M(I))=1	00484000
	IPNUM=1	00485000
	PRINT 3060	00486000
	3060 FORMAT('GROUPS TO BE DRAWN ARE:')	00487000
	PRINT 3070,(M(I),I=1,NOUT)	00488000
	3070 FORMAT(15I4)	00489000
	3080 PRINT 3090	00490000
	3090 FORMAT('//ENTER GROUPS TO BE BLUE:')	00491000
	IPNUM=2	00492000
	GO TO 3140	00493000
	3100 PRINT 3110	00494000
	3110 FORMAT('//ENTER GROUPS TO BE GREEN:')	00495000
	IPNUM=3	00496000
	GO TO 3140	00497000
	3120 PRINT 3130	00498000
	3130 FORMAT('//ENTER GROUPS TO BE RED:')	00499000
	IPNUM=4	00500000
C		00501000
	3140 CALL FREAD(4,&3140,&3150)	00502000
	3150 DO 3180 K=1,NOUT	00503000

CALL IPARM(GC,50,&3140)	00504000
IF(GC.EQ.50) GO TO 3190	00505000
DO 6000 I=1,NOUT	00506000
IF(GC.EQ.M(I)) GO TO 6020	00507000
6000 CONTINUE	00508000
PRINT 6010,GC	00509000
6010 FORMAT('GROUP ',I2,' IS NOT IN LIST TO BE DRAWN')	00510000
GO TO (3080,3080,3100,3120),IPNUM	00511000
6020 ICOL(GC)=IPNUM	00512000
3180 CONTINUE	00513000
3190 GO TO (3080,3100,3120,3220),IPNUM	00514000
C	00515000
3220 IF(COLRET) GO TO 3240	00516000
IF(OP.NE.18) GO TO 3230	00517000
PRINT 690	00518000
GO TO 300	00519000
	00520000
C	00521000
C	00522000
330 IF(NMOV) GO TO 1701	00523000
IF(OP.EQ.6) GO TO 200	00524000
IF(G)GO TO 331	00525000
GO TO 200	00526000
C	00527000
C	00528000
331 CALL PUTOUT	00529000
1703 IDEV=IDEVT	00530000
GO TO 300	00531000
C	00532000
C_PUT SAVED BACKGROUND INTO OUTPUT BUFFER.	00533000
400 IF(NSS+KS.LT.1000) GO TO 410	00534000
CALL PUTOUT	00535000
NS=1	00536000
NF=NSS	00537000
GO TO 415	00538000
410 NS=KS+1	00539000
NF=NS+NSS	00540000
415 J=1	00541000
DO 425 I=NS,NF	00542000
X(I)=XS(J)	00543000
Y(I)=YS(J)	00544000
IS(I)=INS(J)	00545000
425 J=J+1	00546000
KS=I-1	00547000
CALL PUTOUT	00548000
BACK=.FALSE.	00549000
IF(OP.EQ.6) IDEV=IDEVT	00550000
GO TO 300	00551000
C	00552000
C	00553000
1701 NMOV=.FALSE.	00554000
CALL SAME(&1703)	00555000
520 PRINT 521	00556000
521 FORMAT(' NEW GROUP FOR LINE REMOVAL SET OPTION #15')	00557000
GO TO 300	00558000
C	00559000

C		00560000
	900 PRINT 901	00561000
	901 FORMAT('OPTION CANCELLED.')	00562000
	GO TO 300	00563000
	END	00564000
C		00565000
C		00566000
C	-----	00567000
C	SUBROUTINE LOUT - TO CALCULATE THE END COORDINATES FOR	00568000
C	EACH LINE IN AN OBJECT.	00569000
C	-----	00570000
	SUBROUTINE LOUT(JJ)	00571000
	COMMON/POINTS/POINT(350,4)/PDATA/D,OPT/TABLES/NAME(30,5)	00572000
	1/MISCD/IDT,IDEV,SF,RAS	00573000
	COMMON/DATA/X(1000),Y(1000),Z(1000),LINE(1000,2)	00574000
	LOGICAL OPT,RAS	00575000
	DATA XL,YL/999..999./	00576000
	NS=NAME(JJ,4)	00577000
	NF=NS+NAME(JJ,5)-1	00578000
C		00579000
C	PROCESS ONE LINE AT A TIME.	00580000
	DO 200 I=NS,NF	00581000
	J1=LINE(I,1)	00582000
	X1=POINT(J1,1)	00583000
	Y1=POINT(J1,2)	00584000
	Z1=POINT(J1,3)	00585000
	J1=LINE(I,2)	00586000
	X2=POINT(J1,1)	00587000
	Y2=POINT(J1,2)	00588000
	Z2=POINT(J1,3)	00589000
	GO TO (10,100,100,10,100,100),IDT	00590000
C		00591000
C	ADD A PEN MOVE TO THE OUTPUT BUFFER(ROTA,SOLAR).	00592000
	10 CONTINUE	00593000
	IF(ABS(XL-X1).LT..01.AND.ABS(YL-Y1).LT..01)GO TO 11	00594000
	CALL VBLD(X1,Y1,0)	00595000
	11 CALL VBLD(X2,Y2,1)	00596000
	XL=X2	00597000
	YL=Y2	00598000
	GO TO 200	00599000
C		00600000
C	TRY TO CLIP THE LINE(PERS,STEREO,MOVIE).	00601000
	100 DD=-D	00602000
	IF(OPT)DD=0	00603000
	IF(Y1.LE.DD.AND.Y2.LE.DD) GO TO 200	00604000
	IF(Y1.LE.DD) GO TO 25	00605000
	IF(Y2.LE.DD) GO TO 30	00606000
	35 PERS=1+Y1/D	00607000
	IF(OPT) PERS=Y1/D	00608000
	X1=X1/PERS	00609000
	Z1=Z1/PERS	00610000
	PERS=1+Y2/D	00611000
	IF(OPT)PERS=Y2/D	00612000
	X2=X2/PERS	00613000
	Z2=Z2/PERS	00614000
	CALL RAST(X1,Z1,X2,Z2)	00615000

	GO TO 200	00616000
25	YY=-D*.99	00617000
	IF (OPT)YY=.01	00618000
	X1=((X2-X1)*(YY-Y1))/(Y2-Y1)+X1	00619000
	Z1=((Z2-Z1)*(YY-Y1))/(Y2-Y1)+Z1	00620000
	Y1=YY	00621000
	GO TO 35	00622000
30	YY=-D*.99	00623000
	IF(OPT) YY=.01	00624000
	X2=((X2-X1)*(YY-Y1))/(Y2-Y1)+X1	00625000
	Z2=((Z2-Z1)*(YY-Y1))/(Y2-Y1)+Z1	00626000
	Y2=YY	00627000
	GO TO 35	00628000
200	CONTINUE	00629000
	RETURN	00630000
	END	00631000
C		00632000
C		00633000
C	-----	00634000
C	SUBROUTINE MATF - TO CALCULATE SCALE, LOCATION AND	00635000
C	ROTATION MATRIX.	00636000
C	-----	00637000
	SUBROUTINE MATF(XO,YO,ZO,W,L,H,TYP,ANG,A)	00638000
	REAL A(4,4),L,PI/3.14159/	00639000
1	DO 5 I=1,4	00640000
	DO 5 J=1,4	00641000
5	A(I,J)=0.0	00642000
	AN=ANG*PI/180.0	00643000
	A(4,1)=XO	00644000
	A(4,2)=YO	00645000
	A(4,3)=ZO	00646000
	A(4,4)=1.0	00647000
	IF (W.NE.0) GO TO 10	00648000
	A(1,1)=1.	00649000
	A(2,2)=1.	00650000
	A(3,3)=1.	00651000
	GO TO 100	00652000
10	ITYPE=TYP+1.1	00653000
12	GO TO (20,30,40),ITYPE	00654000
C		00655000
C_"Z"	ROTATION.	00656000
20	A(1,1)=COS(AN)*W	00657000
	A(2,1)=-SIN(AN)*L	00658000
	A(1,2)=SIN(AN)*W	00659000
	A(2,2)=COS(AN)*L	00660000
	A(3,3)=H	00661000
	GO TO 100	00662000
C		00663000
C_"X"	ROTATION.	00664000
30	A(1,1)=W	00665000
	A(2,2)=COS(AN)*L	00666000
	A(2,3)=SIN(AN)*L	00667000
	A(3,2)=-SIN(AN)*H	00668000
	A(3,3)=COS(AN)*H	00669000
	GO TO 100	00670000
C		00671000

C_"Y" ROTATION.	00672000
40 A(1,1)=COS(AN)*W	00673000
A(1,3)=-SIN(AN)*W	00674000
A(2,2)=L	00675000
A(3,1)=SIN(AN)*H	00676000
A(3,3)=COS(AN)*H	00677000
100 RETURN	00678000
ENTRY MAT1(XO,YO,ZO,TYP,ANG,A)	00679000
GO TO 50	00680000
ENTRY MAT2(TYP,ANG,A)	00681000
XO=0.	00682000
YO=0.	00683000
ZO=0.	00684000
TYP=TYP-1	00685000
50 L=1.	00686000
H=1.	00687000
W=1.	00688000
GO TO 1	00689000
END	00690000
C	00691000
C	00692000
C-----	00693000
C SUBROUTINE PUTOUT - OUTPUT CONTROL PROGRAM. SCALES	00694000
C FINAL DRAWING.	00695000
C-----	00696000
SUBROUTINE PUTOUT	00697000
COMMON /VECTOR/ DX(1000),DY(1000),ISW(1000),NMP,GOOD,FIRST	00698000
COMMON/MISCD/IDT,IDEV,SF,FRA /PDATA/D,CPT,RX1,RX2,RY1,RY2	00699000
COMMON /PTS/ PT,BAD,BACK /SIZE/XM(6),YM(6),DLX,DLY	00700000
COMMON/B/XS,XLOC,YLOC	00701000
COMMON/JOHN/INIT	00702000
COMMON/JCLFLG/JCLON,NAR	00702.20
LOGICAL JCLON,NAR	00702.40
LOGICAL*1 BACK	00703000
LOGICAL*4 PT,GOOD,BAD,FIRST,INIT,RES	00704000
LOGICAL*4 FRA	00705000
DATA IA,IB /'YES','OK'/	00706000
LOGICAL*4 INT/T/	00707000
IF(INT) CALL DINI(INT)	00708000
IF(PT) GO TO 10	00709000
PRINT 99	00710000
99 FORMAT(' BLANK')	00711000
RETURN	00712000
10 IF(GOOD) GO TO 11	00713000
IF(IDT.EQ.1.OR.IDT.EQ.4) GO TO 500	00714000
IF(.NOT.BAD)GO TO 150	00715000
GO TO 500	00716000
11 IF(.NOT.BAD.OR.IDT.EQ.5.OR.IDT.EQ.6) GO TO 12	00717000
C	00718000
C	00719000
PRINT 93	00720000
93 FORMAT(' WARNING, DRAWING SIZE CALCS FOR 1ST 1000 LINES ONLY')	00721000
12 IF(IDT.NE.1.AND.IDT.NE.4) GO TO 15	00722000
C	00723000
C	00724000
13 PRINT 98	00725000

98	FORMAT('ENTER SCALE FACTOR:')	00725000
700	CALL FREAD(4,&700,&701)	00727000
701	CALL RPARM(XS,0.0,&700)	00728000
	CALL RPARM(DLX2,0.0,&700)	00729000
	CALL RPARM(DLY2,0.0,&700)	00730000
109	IF(XS.LE.0.0) GO TO 15	00731000
	DLX=DLX2	00732000
	DLY=DLY2	00733000
	RES=.TRUE.	00734000
	GO TO 200	00735000
100	DD=XS*AMAX1(XSPAN,YSPAN)	00736000
C		00737000
C		00738000
	PRINT 95,DD	00739000
95	FORMAT('SIZE=',F5.2,',OK(OK)?')	00740000
710	CALL FREAD(4,&710,&711)	00741000
711	CALL APARM(IAN,'NO ',4,&710)	00742000
	IF(IAN.EQ.IA.OR.IAN.EQ.IB) GO TO 220	00743000
	GO TO 13	00744000
15	RES=.FALSE.	00745000
C		00746000
C		00747000
89	PRINT 96	00748000
96	FORMAT('ENTER DRAWING SIZE:')	00749000
720	CALL FREAD(4,&720,&721)	00750000
721	CALL RPARM(XMX,0.0,&720)	00751000
	CALL IPARM(IFL,0,&720)	00752000
	CALL RPARM(DLX,0.0,&720)	00753000
	CALL RPARM(DLY,0.0,&720)	00754000
	NAR=.TRUE.	00754.50
	IF(XMX.LE.XM(IDEV) .AND. XMX.LE.YM(IDEV))GO TO 91	00755000
	IF(IDEV.NE.3) GO TO 300	00756000
	IF(XMX.GT.29) GO TO 320	00757000
	PRINT 310,XMX	00758000
310	FORMAT(//'***WARNING - A DRAWING SIZE OF ',F6.2,'IS TOO LARGE'/	00759000
2	'FOR THE PLOTTER'S STANDARD 10 INCH PAPER WIDTH.'/	00760000
3	'DO YOU WISH TO ENTER A NEW DRAWING SIZE (YES/NO)?')	00761000
	GO TO 730	00762000
320	NAR=.FALSE.	
	PRINT 330,XMX	00763000
330	FORMAT(//'***WARNING - A DRAWING SIZE OF ',F6.2,'IS TOO LARGE'/	00764000
2	'FOR THE PLOTTER'S WIDE PAPER WIDTH.'/	00765000
3	'DO YOU WISH TO ENTER A NEW DRAWING SIZE (YES/NO)?')	00766000
	GO TO 730	00767000
300	PRINT 90,XM(IDEV),YM(IDEV),XMX	00768000
90	FORMAT(//'***WARNING - YOUR OUTPUT DEVICE IS ONLY ',F5.2,	00769000
1	'BY ',F5.2,',.'/	00770000
2	' A DRAWING SIZE OF ',F6.2,' WILL CAUSE A PORTION OF '/	00771000
3	' THE DRAWING TO BE CLIPPED. DO YOU WISH TO ENTER '/	00772000
4	' A NEW DRAWING SIZE(YES,NO)?')	00773000
730	CALL FREAD(4,&730,&731)	00774000
731	CALL APARM(IAN,'NO ',4,&730)	00775000
	IF(IAN.EQ.IA .OR. IAN.EQ.IB)GO TO 89	00776000
	IF(.NOT.NAR) GO TO 91	
	PRINT 600	
600	FORMAT(//'DO YOU WISH TO USE WIDE PAPER (YES/NO)?')	

```

601 CALL FREAD(4,&601,&602)
602 CALL APARM(IAN,'NO ',4,&601)
    IF(IAN.NE.IA .AND. IAN.NE.IB) GO TO 91
    NAR=.FALSE.
C
C
91 IF(XMX.GT.0.)GO TO 149
    XMX=5.
    IFL=1
    DLX=1.
    DLY=.3
149 IF((IDT.EQ.3 .OR. IDT.EQ.5 .OR. IDT.EQ.6) .AND.
2   (IFL.LE.0))IFL=-1
    IF(IFL.EQ.0.OR.IDT.EQ.1.OR.IDT.EQ.4) GO TO 200
150 IF(IFL.LE.0.OR.BAD) GO TO 170
C
C_ADD THE FRAME TO THE OUTPUT BUFFER.
    CALL VBL(RX1,RY1,0)
    CALL VBL(RX2,RY1,1)
    CALL VBL(RX2,RY2,1)
    CALL VBL(RX1,RY2,1)
    CALL VBL(RX1,RY1,1)
C
C_ADD THE FRAME DIVIDER FOR A STEREO VIEW.
    IF(IDT.NE.3 .AND. IDT.NE.6)GO TO 170
    CALL VBL(-RX1,RY2,0)
    CALL VBL(-RX1,RY1,1)
170 IF(.NOT.GOOD) GO TO 500
    XMAX=RX2
    YMAX=RY2
    XMIN=RX1
    YMIN=RY1
    GO TO 215
C
C
C
200 XMAX=-1.0 E20
    YMAX=-1.0 E20
    XMIN= 1.0 E20
    YMIN= 1.0 E20
    DO 210 I=1,NMP
        IF(XMAX.LT.DX(I)) XMAX=DX(I)
        IF(YMAX.LT.DY(I)) YMAX=DY(I)
        IF(XMIN.GT.DX(I)) XMIN=DX(I)
        IF(YMIN.GT.DY(I)) YMIN=DY(I)
210 IF(YMIN.GT.DY(I)) YMIN=DY(I)
215 XSPAN=XMAX-XMIN
    YSPAN=YMAX-YMIN
    IF(RES) GO TO 100
    XS=AMIN1(XM(IDEV)-DLX,XMX)/XSPAN
    YS=AMIN1(YM(IDEV)-DLY,XMX)/YSPAN
    XS=AMIN1(XS,YS)
220 XLOC=-XMIN*XS+DLX
    YLOC=-YMIN*XS+DLY
C
C
C

```

00777000
00778000
00779000
00780000
00781000
00782000
00783000
00784000
00785000
00786000
00787000
00788000
00789000
00790000
00791000
00792000
00793000
00794000
00795000
00796000
00797000
00798000
00799000
00800000
00801000
00802000
00803000
00804000
00805000
00806000
00807000
00808000
00809000
00810000
00811000
00812000
00813000
00814000
00815000
00816000
00817000
00818000
00819000
00820000
00821000
00822000
00823000
00824000
00825000
00826000
00827000
00828000

500	IF(BAD) FRA=.FALSE.	00829000
	CALL DEVICE(INIT,FRA)	00830000
	FRA=.TRUE.	00831000
	INIT=.TRUE.	00832000
	IF(BAD) INIT=.FALSE.	00833000
	FIRST=.FALSE.	00834000
	RETURN	00835000
	END	00836000
C		00837000
C		00838000
C	-----	00839000
C	SUBROUTINE RAST ~ TO CLIP LINES OUTSIDE OF VIEW FIELD.	00840000
C	-----	00841000
	SUBROUTINE RAST(X1,Y1,X2,Y2)	00842000
	COMMON/PDATA/D,OPT,RX1,RX2,RY1,RY2	00843000
	LOGICAL OPT,ENT	00844000
	REAL M	00845000
	INTEGER Q	00846000
	DATA XS/999./,YS/999./	00847000
	DIMENSION XX(2), YY(2)	00848000
	ENT=.FALSE.	00849000
	GO TO 200	00850000
	ENTRY RTEST(X1,Y1,X2,Y2)	00851000
200	IQ=0	00853000
200	IQ=0	00853000
	IF(X1.LT.RX1.AND.X2.LT.RX1.OR.X1.GT.RX2.AND.X2.GT.RX2)GO TO 100	00854000
	IF(Y1.LT.RY1.AND.Y2.LT.RY1.OR.Y1.GT.RY2.AND.Y2.GT.RY2) GO TO 100	00855000
	IF(X1.GE.RX1.AND.X1.LE.RX2.AND.Y1.GE.RY1.AND.Y1.LE.RY2) GO TO 19	00856000
	IF(ABS(X2-X1).LE..001) GO TO 2	00857000
	M=(Y2-Y1)/(X2-X1)	00858000
2	X=X1	00859000
	Y=Y1	00860000
	Q=1	00861000
10	IQ=IQ+1	00862000
	IF (IQ .EQ. 4)GO TO 100	00863000
	IF (X .LT. RX1) GO TO 11	00864000
	IF (X .GT. RX2) GO TO 12	00865000
	IF (Y .LT. RY1) GO TO 13	00866000
	IF (Y .GT. RY2) GO TO 14	00867000
	GO TO (20,30),Q	00868000
11	XX(Q)=RX1	00869000
	GO TO 15	00870000
12	XX(Q)=RX2	00871000
	GO TO 15	00872000
13	YY(Q)=RY1	00873000
	GO TO 16	00874000
14	YY(Q)=RY2	00875000
	GO TO 16	00876000
21	XX(Q)=((YY(Q)-Y)/M)+X	00877000
	GO TO 17	00878000
16	IF (X2 .NE. X1) GO TO 21	00879000
	XX(Q)=X	00880000
17	X=XX(Q)	00881000
	Y=YY(Q)	00882000
	GO TO 10	00883000
15	YY(Q)=M*(XX(Q)-X)+Y	00884000

```

      GO TO 17                                00885000
19  XX(1)=X1                                00886000
      YY(1)=Y1                                00887000
20  IF(X2.GE.RX1.AND.X2.LE.RX2.AND.Y2.GE.RY1.AND.Y2.LE.RY2) GO TO 29 00888000
      IF(ABS(X2-X1).LE.0.001) GO TO 4         00889000
      M=(Y2-Y1)/(X2-X1)                      00890000
4    Q=2                                    00891000
      X=X2                                    00892000
      Y=Y2                                    00893000
      IQ=0                                    00894000
      GO TO 10                                00895000
29  XX(2)=X2                                00896000
      YY(2)=Y2                                00897000
30  CONTINUE                                00898000
      IF(ENT)GO TO 120                        00899000
      IF(ABS(XS-XX(1)).LT..0001.AND.ABS(YS-YY(1)).LT..0001) GO TO 35 00900000
      CALL VELD(XX(1),YY(1),0)                00901000
35  CALL VBLD(XX(2),YY(2),1)                00902000
      XS=XX(2)                                00903000
      YS=YY(2)                                00904000
100 RETURN                                00905000
120 X1=XX(1)                                00906000
      X2=XX(2)                                00907000
      Y1=YY(1)                                00908000
      Y2=YY(2)                                00909000
      RETURN                                00910000
      END                                    00911000
C                                           00912000
C                                           00913000
C-----                                00914000
C SUBROUTINE SSAVE - TO EXPAND THE GROUP DATA STRUCTURE FOR 00915000
C                               HIDDEN-LINE REMOVAL OR FOR SAVING TO A FILE. 00916000
C-----                                00917000
      SUBROUTINE SSAVE(JJ,K,*)                00918000
      COMMON/DATA/X(1000),Y(1000),Z(1000),LINE(1000,2) 00919000
      COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF, 00920000
1  NEXTP,NEXTL,ITP,H1,NTRAT                  00921000
      COMMON/POINTS/POINT(350,4)/TABLES/NAME(30,5)      00922000
      COMMON/PLANE/NPP,NPLA(30,2),IVEC(800),MYLINE(1000,2) 00923000
      LOGICAL NODRAT,INSAV,HID,H1              00924000
      IF(.NOT.INSAV)GO TO 20                  00925000
      NEXTF=1                                00926000
      NEXTP=0                                00927000
      NEXTL=0                                00928000
      ITP=0                                  00929000
      INSAV=.FALSE.                          00930000
20  IF(NEXTP+K.GE.350)GO TO 100              00931000
      DO 10 I=1,K                            00932000
      DO 10 I1=1,3                            00933000
10  PUNT(NEXTP+I,I1)=POINT(I,I1)             00934000
      NEXTP=NEXTP+K                          00935000
      NS=NAME(JJ,4)                          00936000
      NF=NS+NAME(JJ,5)-1                     00937000
      N=1                                    00938000
      IF(NEXTL+NF.GE.1000)GO TO 100          00939000
      DO 30 I=NS,NF                          00940000

```

	DO 31 I1=1,2	00941000
31	MYLINE(NEXTL+N,I1)=LINE(I,I1)+NEXTP-K	00942000
30	N=N+1	00943000
	NEXTL=NEXTL+N-1	00944000
	NP=NPLA(JJ,1)	00945000
	NPS=NPLA(JJ,2)	00946000
	IF(NEXTP+5*NP.GE.1200)GO TO 100	00947000
	DO 40 I=1,NP	00948000
	N=0	00949000
	IST=IVEC(NPS)+NPS	00950000
	DO 41 I1=NPS,IST	00951000
	IPLA(NEXTF+N)=IVEC(I1)+NEXTP-K	00952000
41	N=N+1	00953000
	IPLA(NEXTF)=IVEC(NPS)	00954000
	NPS=IST+1	00955000
	NEXTF=NEXTF+N	00956000
40	CONTINUE	00957000
	ITP=ITP+NP	00958000
42	RETURN 1	00959000
100	CALL ERRORS(29,350,'NO SPACE',&42)	00960000
	RETURN	00961000
	END	00962000
C		00963000
C		00964000
C	-----	00965000
C	SUBROUTINE SWAP - TO CHANGE NAME OF MATRIX. B=A.	00966000
C	-----	00967000
	SUBROUTINE SWAP(A,B)	00968000
	REAL A(4,4),B(4,4)	00969000
	DO 10 I=1,4	00970000
	DO 10 J=1,4	00971000
10	B(I,J)=A(I,J)	00972000
	RETURN	00973000
	END	00974000
C		00975000
C		00976000
C	-----	00977000
C	SUBROUTINE TRAT - TO SET UP DASHED-LINES.	00978000
C	-----	00979000
	SUBROUTINE TRAT(X1,Y1,X2,Y2,S)	00980000
100	FIN=0	00981000
	CALL HVB(X1,Y1,0)	00982000
	IF(X1.EQ.X2.AND.Y1.EQ.Y2)RETURN	00983000
	DX=X2-X1	00984000
	DY=Y2-Y1	00985000
	DL=SQRT(DX*DX+DY*DY)	00986000
	DEPX=DX*S/DL	00987000
	DEPY=DY*S/DL	00988000
	TEST=ABS(DX)	00989000
	ADEP=ABS(DEPX)	00990000
	IF(TEST)1,1,2	00991000
1	TEST=ABS(DY)	00992000
	ADEP=ABS(DEPY)	00993000
2	X=X1	00994000
	Y=Y1	00995000
5	X=X+DEPX	00996000

	Y=Y+DEPY	00997000
	FIN=FIN+ADEP	00998000
	IF(FIN-TEST)11,20,20	00999000
11	CALL HVB(X,Y,1)	01000000
	X=X+DEPX	01001000
	Y=Y+DEPY	01002000
	FIN=FIN+ADEP	01003000
	IF(FIN-TEST)14,20,20	01004000
14	CALL HVB(X,Y,0)	01005000
	GO TO 5	01006000
20	CALL HVB(X2,Y2,1)	01007000
	RETURN	01008000
	END	01009000
C		01010000
C		01011000
C	-----	01012000
C	SUBROUTINE VBLD - TO BUILD THE OUTPUT BUFFERS(X,Y,ISW).	01013000
C	THERE IS AN AUTOMATIC OUTPUT AFTER 994 LINES.	01014000
C	-----	01015000
	SUBROUTINE VBLD(XX,YY,IS)	01016000
	COMMON /PTS/PT,BAD	01017000
	COMMON/MISCD/IDT,IDEV,SF,FRA	01018000
	COMMON/SDATA/IOT,RTV	01019000
	LOGICAL RTV	01020000
	COMMON /VECTOR/X(1000),Y(1000),ISW(1000),NSTOR,GOOD,FIRST	01021000
	LOGICAL GOOD,FIRST	01022000
	LOGICAL PT,BAD	01023000
	BAD=.FALSE.	01024000
	PT=.TRUE.	01025000
	IF(FIRST) GO TO 20	01026000
	NSTOR=0	01027000
20	IF(IDT.NE.3 .AND. IDT.NE.6)GO TO 21	01028000
	IF(IOT.EQ.3)GO TO 30	01029000
	IF(.NOT.RTV)GO TO 21	01030000
	XX=XX+9.2	01031000
	GO TO 21	01032000
30	IF(RTV)GO TO 31	01033000
	T=XX	01034000
	XX=YY	01035000
	YY=-T	01036000
	GO TO 21	01037000
31	T=XX	01038000
	XX=-YY+9.2/1.333	01039000
	YY=T	01040000
	ENTRY VBL(XX,YY,IS)	01041000
C		01042000
C_ADD	THE PEN MOVE TO THE OUTPUT BUFFER.	01043000
21	NSTOR=NSTOR+1	01044000
	X(NSTOR)=XX	01045000
	Y(NSTOR)=YY	01046000
	ISW(NSTOR)=IS	01047000
	IF(NSTOR .EQ.994) GO TO 100	01048000
	FIRST=.TRUE.	01049000
	RETURN	01050000
100	CONTINUE	01051000
	BAD=.TRUE.	01052000

200	CALL PUTOUT	01053000
	GOOD=.FALSE.	01054000
	RETURN	01055000
	END	01056000
	RETURN	01050000
100	CONTINUE	01051000
	BAD=.TRUE.	01052000
200	CALL PUTOUT	01053000
	GOOD=.FALSE.	01054000
	RETURN	01055000
	END	01056000

C		00001000
C		00002000
C	-----	00003000
C	SUBROUTINE CELATE - TO REMOVE THE HIDDEN LINES.	00004000
C	-----	00005000
	SUBROUTINE CELATE	00006000
	COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF,	00007000
	1NEXTP,NEXTL,ITP,H1,NTRAT	00008000
	LOGICAL NODRAT,HID,INSAV,H1	00009000
	COMMON PUPRVI(350,3),PUNTRA(350,3),IFACP(5000),	00010000
	1IREPU(500,2),IREFA(500,2),IPTR(250,3),SIPA(250),ICONV(500),	00011000
	2NP,NF,NR,LIB,IES,XO,YO,ZO,TO,ISIPU(350),IPPC(1000)	00012000
	DIMENSION INIZIO(350),NOFA(2650),TAT(500),A(4,4),	00013000
	1RIORD(500,2),LIRIO(500),LIFA(500),SIRE(500),SITRA(250)	00014000
	DIMENSION AAA(3)	00015000
	EQUIVALENCE (INIZIO(1),IFACP(1)),(LIRIO(1),IFACP(351)),(LIFA(1),IF	00016000
	1ACP(851)),(NOFA(1),IFACP(1351)),(RIORD(1,1),IFACP(4001))	00017000
	LOGICAL ER,PG1,PG2,IDEA(350),ODI, P1,P2,P1C,P2C	00018000
	9EAL NN	00019000
	CAR=3.141516 E30	00020000
	DO 9 I=1,NP	00021000
9	INIZIO(I)=0	00022000
	LIR =1	00023000
	N=4	00024000
	ALPHA=0.	00025000
	NN=1.	00026000
	SIG=1.	00027000
	IF(XO.LT.0.) SIG=-1.	00028000
	SI=-SIG	00029000
	DO 10 I=1,NF	00030000
	I1=IPTR(I,1)	00031000
	I2=IPTR(I,2)	00032000
	I3=IPTR(I,3)	00033000
	A1=PUNTRA(I1,2)-PUNTRA(I3,2)	00034000
	B1=PUNTRA(I1,3)-PUNTRA(I3,3)	00035000
	A2=PUNTRA(I2,2)-PUNTRA(I3,2)	00036000
	B2=PUNTRA(I2,3)-PUNTRA(I3,3)	00037000
	D=A1*B2-A2*B1	00038000
10	SITRA(I)=D*SIPA(I)*SI	00039000
	DO 20 I=1,NR	00040000
	SIRE(I)=-1	00041000
	KS1=IREFA(I,1)	00042000
	KS2=IREFA(I,2)	00043000
	ER=ICONV(I).GT.0	00044000
	PG1=SITRA(KS1).GT.0.	00045000
	PG2=SITRA(KS2).GT.0.	00046000
	IF(PG1.OR.PG2)SIRE(I)=1	00047000
	IF(.NOT.ER.AND..NOT.PG1.AND.PG2.OR..NOT.ER.AND.PG1.AND..NOT.PG2)	00048000
	1SIRE(I)=0.	00049000
20	CONTINUE	00050000
	IF(IES.EQ.0)GO TO 950	00051000
	DO 951 I=1,NR	00052000
	KS1=IREPU(I,1)	00053000
	KS2=IREPU(I,2)	00054000
	IF(SIRE(I).LE.0.)GO TO 953	00055000

	CALL HVB(PUNTRA(KS1,2)*SIG,PUNTRA(KS1,3),0)	000560C0
	CALL HVB (PUNTRA(KS2,2)*SIG,PUNTRA(KS2,3),1)	00057000
	GO TO 951	00058000
953	IF(NTRAT.EQ.0)GO TO 951	00059000
	CALL TRAT(PUNTRA(KS1,2)*SIG,PUNTRA(KS1,3),PUNTRA(KS2,2)*SIG,PUNTRA(KS2,3),0.2)	000600C0
	CONTINUE	00061000
951	GO TO 1000	00062000
	DO 30 I=1,NR	00063000
950	IF(SIRE(I).GT.0.)GO TO 973	00064000
	IF(NTRAT.EQ.0)GO TO 30	00065000
	KS1=IREPU(I,1)	00066000
	KS2=IREPU(I,2)	00067000
	CALL TRAT(PUNTRA(KS1,2)*SIG,PUNTRA(KS1,3),PUNTRA(KS2,2)*SIG,PUNTRA(KS2,3),0.2)	00068000
	GO TO 30	00069000
973	KS1=IREPU(I,1)	00070000
	KS2=IREPU(I,2)	00071000
	PG1=INIZIO(KS1).NE.0	00072000
	PG2=INIZIO(KS2).NE.0	00073000
	P1C=ISIPU(KS1).EQ.0	00074000
	P2C=ISIPU(KS2).EQ.0	00075000
	P1=PG1.AND.P1C	00076000
	P2=PG2.AND.P2C	00077000
	ER=P1.OR.P2	00078000
	KVOLT=0	00079000
	IF(ER)GO TO 710	00080000
	ODI=.TRUE.	00081000
	IF(PG1)GO TO 111	00082000
	IF(PG2)GO TO 722	00083000
	ODI=.FALSE.	00084000
	GO TO 111	00085000
710	IF(P2)GO TO 722	00086000
	GO TO 111	00087000
722	IKK=KS1	00088000
	KS1=KS2	00089000
	KS2=IKK	00090000
111	KSS1=INIZIO(KS1)	00091000
	IF(KSS1.EQ.0)GO TO 113	00092000
112	NOF=NOFA(KSS1)	00093000
	IF(NOF.EQ.(-1)) GO TO 113	00094000
	KVOLT=KVOLT+1	00095000
	LIFA(KVOLT)=NOF	00096000
	KSS1=KSS1+1	00097000
	GO TO 112	00098000
113	TMIN=1.	00099000
18	Y1P=PUNTRA(KS1,2)	00100000
	AY1=PUNTRA(KS2,2)-Y1P	00101000
	Z1P=PUNTRA(KS1,3)	00102000
	AZ1=PUNTRA(KS2,3)-Z1P	00103000
	X3P=PUPRVI(KS1,1)	00104000
	Y3P=PUPRVI(KS1,2)	00105000
	Z3P=PUPRVI(KS1,3)	00106000
	CX1=PUPRVI(KS2,1)-X3P	00107000
	CY1=PUPRVI(KS2,2)-Y3P	00108000
	CZ1=PUPRVI(KS2,3)-Z3P	00109000
		00110000

	CY1X1P=CY1*X3P	00112000
	CY1X0=CY1*X0	00113000
	CZ1X1P=CZ1*X3P	00114000
	CZ1X0=CZ1*X0	00115000
	DO 16 J=1,NR	00116000
16	TAT(J)=CAR	00117000
	DISP=0.0002/(ABS(AY1)+ABS(AZ1))	00118000
15	DO 290 J=1,NR	00119000
	IF(SIRE(J).LT.0.) GO TO 29	00120000
	IF(I.EQ.J) GO TO 29	00121000
	KS3=IREPU(J,1)	00122000
	KS4=IREPU(J,2)	00123000
	IF(KS1.EQ.KS3.OR.KS1.EQ.KS4.OR.KS2.EQ.KS3.OR.KS2.EQ.KS4) GO TO 29	00124000
19	Y2P=PUNTRA(KS3,2)	00125000
	AY2=PUNTRA(KS4,2)-Y2P	00126000
	Z2P=PUNTRA(KS3,3)	00127000
	AZ2=PUNTRA(KS4,3)-Z2P	00128000
	IF(AY2.EQ.0.) GO TO 21	00129000
	DET=AZ1*AY2-AZ2*AY1	00130000
	IF (DET.EQ.0.) GO TO 29	00131000
	T1M=(AY2*(Z2P-Z1P)-AZ2*(Y2P-Y1P))/DET	00132000
	IF(T1M.LT.(-DISP)) GO TO 29	00133000
	IF ((T1M-1.).GT.DISP) GO TO 29	00134000
	YSM=AY1*T1M+Y1P	00135000
	T2M=(YSM-Y2P)/AY2	00136000
	GO TO 22	00137000
21	IF (AY1.EQ.0.) GO TO 29	00138000
	T1MN=(Y2P-Y1P)	00139000
	T1M=T1MN/AY1	00140000
	IF(T1M.LT.(-DISP)) GO TO 29	00141000
	IF ((T1M-1.).GT.DISP) GO TO 29	00142000
39	YSM=Y2P	00143000
	T2M=(AZ1*T1M+Z1P-Z2P)/AZ2	00144000
22	DIS2=0.0002/(ABS(AY2)+ABS(AZ2))	00145000
	IF(T2M.LT.(-DIS2)) GO TO 29	00146000
	IF(T2M.LT.DIS2) GO TO 40	00147000
	IF((T2M-1.).GT.DIS2) GO TO 29	00148000
42	KS=KS4	00149000
	INN=2	00150000
	IF ((T2M-1.).LT.(-DIS2)) GO TO 43	00151000
	GO TO 33	00152000
34	KS=KS2	00153000
	INN=1	00154000
	GO TO 33	00155000
40	KS=KS3	00156000
	INN=2	00157000
	GO TO 33	00158000
32	KS=KS1	00159000
	INN=1	00160000
33	PUNTRA(KS,2)=PUNTRA(KS,2)+NN*0.0011582	00161000
	PUNTRA(KS,3)=PUNTRA(KS,3)+NN*0.0021824	00162000
	NN=NN+1.	00163000
	GO TO (18,19),INN	00164000
43	IF(T1M.LT.DISP) GO TO 32	00165000
	IF ((T1M-1.).GT.(-DISP)) GO TO 34	00166000
	CX2=PUPRVI(KS4,1)-PUPRVI(KS3,1)	00167000

	CY2=PUPRVI(KS4,2)-PUPRVI(KS3,2)	00168000
	YM=YSM	00169000
	IF(TO.EQ.0)YM=0.	00170000
	YMY0=YM-YO	00171000
	DET1=CY1X0+CX1*(YMY0)	00172000
	IF(DET1.EQ.0.) GO TO 23	00173000
	V1=(CY1X1P+CX1*(YSM-Y3P))/DET1	00174000
	GO TO 24	00175000
23	ZSM=AZ1*T1M+Z1P	00176000
	ZM=ZSM	00177000
	IF(TO.EQ.0)ZM=0.	00178000
	ZMZ0=ZM-ZO	00179000
	DET1=CZ1X0+CX1*(ZMZ0)	00180000
	V1=(CZ1X1P+CX1*(ZSM-Z3P))/DET1	00181000
24	DET2=CY2*X0+CX2*(YMY0)	00182000
	IF(DET2.EQ.0) GO TO 25	00183000
	V2=(CY2*PUPRVI(KS3,1)+CX2*(YSM-PUPRVI(KS3,2)))/DET2	00184000
	GO TO 26	00185000
25	ZSM=AZ1*T1M+Z1P	00186000
	ZM=ZSM	00187000
	IF(TO.EQ.0)ZM=0.	00188000
	ZMZ0=ZM-ZO	00189000
	CZ2=PUPRVI(KS4,3)-PUPRVI(KS3,3)	00190000
	V2=(CZ2*PUPRVI(KS3,1)+CX2*(ZSM-PUPRVI(KS3,3)))/(CZ2*X0+CX2*ZMZ0)	00191000
26	IF (V1 .GE.V2) GO TO 29	00192000
	TAT(J)=T1M	00193000
	IF(TMIN.LE.T1M) GO TO 29	00194000
	TMIN=T1M	00195000
	SX=X0*V1	00196000
	SY=YSM-YMY0*V1	00197000
	SZ=ZSM-ZMZ0*V1	00198000
29	CONTINUE	00199000
290	CONTINUE	00200000
	KP=KS1	00201000
	KS=KS2	00202000
100	MAX=0	00203000
	DO133 J=1,NR	00204000
	TAS=TAT(J)	00205000
	IF(TAS.EQ.CAR)GO TO 133	00206000
	MAX=MAX+1	00207000
	RIORD(MAX,1)=TAS	00208000
	RIORD(MAX,2)=J	00209000
	LIRIO(MAX)=1	00210000
133	CONTINUE	00211000
	MA1=MAX-1	00212000
	IF(MA1)729,730,731	00213000
731	DO136 I1=1,MA1	00214000
	MIN=I1+1	00215000
	DO136 I2=MIN,MAX	00216000
	IF(RIORD(I1,1).LE.RIORD(I2,1)) GO TO 136	00217000
	TEMP=RIORD(I1,1)	00218000
	RIORD(I1,1)=RIORD(I2,1)	00219000
	RIORD(I2,1)=TEMP	00220000
	TEMP=RIORD(I1,2)	00221000
	RIORD(I1,2)=RIORD(I2,2)	00222000
	RIORD(I2,2)=TEMP	00223000

136	CONTINUE	00224000
	GO TO 730	00225000
729	IF(ER)GO TO 800	00226000
	YT=(PUNTRA(KP,2)+PUNTRA(KS,2))/2.	00227000
	ZT=(PUNTRA(KP,3)+PUNTRA(KS,3))/2.	00228000
	DO 397 K8=1,3	00229000
397	AAA(K8)=(PUPRVI(KP,K8)+PUPRVI(KS,K8))/2.	00230000
	GO TO 1736	00231000
730	IF(ER)GO TO 800	00232000
	TT=RIORD(1,1)	00233000
	YT=AY1*TT+Y1P	00234000
	ZT=AZ1*TT+Z1P	00235000
	YT=(YT+PUNTRA(KP,2))/2.	00236000
	ZT=(ZT+PUNTRA(KP,3))/2.	00237000
	AAA(1)=(SX+PUPRVI(KP,1))/2.	00238000
	AAA(2)=(SY+PUPRVI(KP,2))/2.	00239000
	AAA(3)=(SZ+PUPRVI(KP,3))/2.	00240000
1736	AP=0.	00241000
736	DO 17 J=1,NF	00242000
17	IDEA(J)=.TRUE.	00243000
737	AP=AP+0.01	00244000
	DO 740 J=1,NR	00245000
	IF(SIRE(J).LT.0.)GO TO 740	00246000
	IF (I.EQ.J) GO TO 740	00247000
	IF (.NOT.ODI) GO TO 739	00248000
	DO 435 K1=1,2	00249000
	K5=IREFA(J,K1)	00250000
	K6=ISIPU(KS1)	00251000
434	IF (IPPC(K6).EQ.(-1)) GO TO 435	00252000
	IF (IPPC(K6).EQ.K5) GO TO 739	00253000
	K6=K6+1	00254000
	GO TO 434	00255000
435	CONTINUE	00256000
	GO TO 740	00257000
739	K1=IREPU(J,1)	00258000
	K2=IREPU(J,2)	00259000
	Y1T=PUNTRA(K1,2)-YT	00260000
	Z1T=PUNTRA(K1,3)-ZT	00261000
	D1=Z1T-AP*Y1T	00262000
	IF(ABS(D1).LT.0.0001)GO TO 736	00263000
	Y2T=PUNTRA(K2,2)-YT	00264000
	Z2T=PUNTRA(K2,3)-ZT	00265000
	D2=Z2T-AP*Y2T	00266000
	IF(ABS(D2).LT.0.0001)GO TO 736	00267000
	IF((D1.GT.0..AND.D2.GT.0.).OR.(D1.LT.0..AND.D2.LT.0.))GO TO 740	00268000
	D2=Y1T*Z2T-Y2T*Z1T	00269000
	IF((D1.GT.0..AND.D2.GT.0.).OR.(D1.LT.0..AND.D2.LT.0.))GO TO 740	00270000
	KF1=IREFA(J,1)	00271000
	KF2=IREFA(J,2)	00272000
	IDEA(KF1)=.NOT.IDEA(KF1)	00273000
	IDEA(KF2)=.NOT.IDEA(KF2)	00274000
740	CONTINUE	00275000
	KI1=IREFA(I,1)	00276000
	KI2=IREFA(I,2)	00277000
	IF(ODI)GO TO 741	00278000
	INIZIO(KP)=LIR	00279000

	DO 60 J=1,NF	00280000
	IF(SITRA(J).LE.0..OR.IDEA(J)) GO TO 60	00281000
	IF (J.EQ.KI1.OR.J.EQ.KI2) GO TO 60	00282000
	M1=IPTR(J,1)	00283000
	M2=IPTR(J,2)	00284000
	M3=IPTR(J,3)	00285000
	DO 50 K1=1,4	00286000
50	A(K1,4)=1.	00287000
	DO 51 K1=1,3	00288000
	A(1,K1)=AAA(K1)	00289000
	A(2,K1)=PUPRVI(M1,K1)	00290000
	A(3,K1)=PUPRVI(M2,K1)	00291000
51	A(4,K1)=PUPRVI(M3,K1)	00292000
	CALL KDET(A,ALPHA,N,N,N,DET1)	00293000
	IF (DET1/SIPA(J).LE.0.) GO TO 60	00294000
	KVOLT=KVOLT+1	00295000
	LIFA(KVOLT)=J	00296000
	ISI=ISIPU(KP)	00297000
	IF (ISI.EQ.0) GO TO 760	00298000
759	K=IPPCC(ISI)	00299000
	IF (K.EQ.(-1)) GO TO 760	00300000
	IF(J.EQ.K) GO TO 60	00301000
	ISI=ISI+1	00302000
	GO TO 759	00303000
760	NOFA(LIR)=J	00304000
	LIR=LIR+1	00305000
60	CONTINUE	00306000
	NOFA(LIR)=-1	00307000
	LIR=LIR+1	00308000
	GO TO 800	00309000
741	KT=ISIPU(KP)	00310000
540	J=IPPCC(KT)	00311000
	IF(J.EQ.(-1))GO TO 800	00312000
	IF(SITRA(J).LE.0..OR.IDEA(J))GO TO 550	00313000
	IF (J.EQ.KI1.OR.J.EQ.KI2) GO TO 550	00314000
	M1=IPTR(J,1)	00315000
	M2=IPTR(J,2)	00316000
	M3=IPTR(J,3)	00317000
	DO750 K1=1,4	00318000
750	A(K1,4)=1	00319000
	DO751 K1=1,3	00320000
	A(1,K1)=AAA(K1)	00321000
	A(2,K1)=PUPRVI(M1,K1)	00322000
	A(3,K1)=PUPRVI(M2,K1)	00323000
751	A(4,K1)=PUPRVI(M3,K1)	00324000
	CALL KDET(A,ALPHA,N,N,N,DET1)	00325000
	IF (DET1/SIPA(J).LE.0.) GO TO 550	00326000
	KVOLT=KVOLT+1	00327000
	LIFA(KVOLT)=J	00328000
550	KT=KT+1	00329000
	GO TO 540	00330000
800	NFCMN=KVOLT	00331000
	IPEP=3	00332000
	IF(NFCMN.EQ.0)IPEP=2	00333000
	J=1	00334000
	IF(MAX.EQ.0)GO TO 116	00335000

137	AKK=RIORD(J,1)	00336000
	IKK=RIORD(J,2)	00337000
	M=J	00338000
	KS1=IREFA(IKK,1)	00339000
	KS2=IREFA(IKK,2)	00340000
	PG1=SITRA(KS1).GT.0.	00341000
	PG2=SITRA(KS2).GT.0.	00342000
138	IF(NFCMN.EQ.0) GO TO 146	00343000
	IF(.NOT.PG1) GO TO 141	00344000
	DO140 K=1,KVOLT	00345000
	LIF=LIFA(K)	00346000
	IF(LIF.NE.KS1) GO TO140	00347000
	IF(.NOT.PG2) GO TO147	00348000
	LIFA(K)=KS2	00349000
	GO TO155	00350000
140	CONTINUE	00351000
	IF(PG2) GO TO 141	00352000
	KVOLT=KVOLT+1	00353000
	NFCMN=NFCMN+1	00354000
	LIFA(KVOLT)=KS1	00355000
	IF(NFCMN.EQ.1) LIRIO(M)=3	00356000
	GO TO 155	00357000
141	DO145 K=1,KVOLT	00358000
	LIF=LIFA(K)	00359000
	IF(LIF.NE.KS2) GO TO 145	00360000
	IF(.NOT.PG1) GO TO 147	00361000
	LIFA(K)=KS1	00362000
	GO TO 155	00363000
145	CONTINUE	00364000
	KVOLT=KVOLT+1	00365000
	NFCMN=NFCMN+1	00366000
	LIFA(KVOLT)=KS2	00367000
	IF(NFCMN.EQ.1)LIRIO(M)=3	00368000
	GO TO 155	00369000
146	KVOLT=KVOLT+1	00370000
	NFCMN=1	00371000
	LIFA(KVOLT)=KS1	00372000
	IF(PG2)LIFA(KVOLT)=KS2	00373000
	LIRIO(M)=3	00374000
	GO TO 155	00375000
147	LIFA(K)=-1	00376000
	NFCMN=NFCMN-1	00377000
	IF(NFCMN.EQ.0)LIRIO(M)=2	00378000
155	J=J+1	00379000
	IF(J.LE.MAX) GO TO 137	00380000
116	IF (INIZIO(KS).NE.0) GO TO 119	00381000
	INIZIO (KS)=LIR	00382000
	IF (NFCMN.EQ.0) GO TO 118	00383000
	ISI=ISIPU(KS)	00384000
	DO 117 J=1,KVOLT	00385000
	LIF=LIFA(J)	00386000
	IF (LIF.EQ.(-1)) GO TO 117	00387000
	IF (ISI.EQ.0)GO TO 128	00388000
833	IPP=IPFCC(ISI)	00389000
	IF (IPP.EQ.(-1))GO TO 128	00390000
	IF (LIF.EQ.IPP) GO TO 117	00391000

	ISI=ISI+1	00392000
	GO TO 833	00393000
128	NOFA(LIR)=LIF	00394000
	LIR=LIR+1	00395000
117	CONTINUE	00396000
118	NOFA(LIR)=-1	00397000
	LIR=LIR+1	00398000
119	IF (MAX.NE.0) GO TO 436	00399000
	IF((IPEP.EQ.3).AND.(NTRAT.EQ.0)) GO TO 30	00400000
	GO TO 1190	00401000
436	J8=1	00402000
	IF (IPEP.EQ.2) GO TO 1190	00403000
	DO 485 J8=1,MAX	00404000
	IF (LIRIO(J8).EQ.2) GO TO 1190	00405000
485	CONTINUE	00406000
	IF (NTRAT.EQ.0) GO TO 30	00407000
	MAX=0	00408000
1190	XPR=PUNTRA(KP,2)*SIG	00409000
	YPR=PUNTRA(KP,3)	00410000
	CALL HVB(XPR,YPR,0)	00411000
	IF(MAX.EQ.0)GO TO 437	00412000
	DO 120 J=J8,MAX	00413000
	LII=LIRIO(J)	00414000
	IF(LII.EQ.1)GO TO 120	00415000
	RIO=RIORD(J,1)	00416000
	YY=SIG*(AY1*RIC+Y1P)	00417000
	ZZ=AZ1*RIC+Z1P	00418000
	IF(IPEP.EQ.2)GO TO 922	00419000
	IF(NTRAT.NE.0)GO TO 933	00420000
922	KC=1	00421000
	IF(IPEP.EQ.3)KC=0	00422000
	CALL HVB (YY,ZZ,KC)	00423000
	GO TO 934	00424000
933	CALL TRAT(XPR,YPR,YY,ZZ,0.2)	00425000
934	IPEP=LII	00426000
	YPR=YY	00427000
	YPR=ZZ	00428000
120	CONTINUE	00429000
437	XST=SIG*PUNTRA(KS,2)	00430000
	IF(NTRAT.NE.0 .AND.IPEP.EQ.3) GO TO 493	00431000
	IF (IPEP.EQ.3) GO TO 30	00432000
	CALL HVB(XST,PUNTRA(KS,3),1)	00433000
	GO TO 30	00434000
493	CALL TRAT(XPR,YPR,XST,PUNTRA(KS,3),0.2)	00435000
30	CONTINUE	00436000
1000	RETURN	00437000
	END	00438000
C		00439000
C		00440000
C	-----	00441000
C	SUBROUTINE ENTRA - HIDDEN LINE ROUTINE.	00442000
C	GALIMBERTI AND MONTANARI.	00443000
C	CALCULATES OBJECT DATA INDEPENDENT FROM VIEW.	00444000
C	-----	00445000
	SUBROUTINE ENTRA	00446000
	COMMON PUPRVI(350,3),PUNTRA(350,3),IFACP(5000),	00447000

```

1IREPU(500,2),IREFA(500,2),IPTR(250,3),SIPA(250),ICONV(500),      00448000
2NP,NF,NR,LIB,IES,XO,YO,ZO,TO,ISIPU(350),IPPC(1000)                00449000
COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF,          00450000
1NEXTP,NEXTL,ITP,H1,NTRAT                                           00451000
LOGICAL NODRAT,INSAV,HID,H1                                         00452000
DIMENSION LINEA(100),XYZ(100,3),ZZZ(250,3),IRE(1000,2),IORD1(1000) 00453000
DIMENSION IR(1000),IORD2(1000),A(3,3)                               00454000
EQUIVALENCE (IRE(1,1),IFACP(1)),(IORD1(1),IFACP(2001)),(IR(1),IFAC00455000
1P(3001)),(IORD2(1),IFACP(4001))                                     00456000
DO 68 I=1,NP                                                         00457000
68  ISIPU(I)=0                                                         00458000
LIB2=0                                                                00459000
LIB=1                                                                00460000
NH1=1                                                                00461000
DO 100 I=1,NF                                                         00462000
LL=IPLA(NH1)                                                         00463000
DO 101 K=1,LL                                                         00464000
LINEA(K)=IPLA(NH1+K)                                                 00465000
101 CONTINUE                                                         00466000
NH1=NH1+LL+1                                                         00467000
LINEA(LL+1)=-100                                                     00468000
DO 52 K=1,3                                                           00469000
KS1=LINEA(K)                                                         00470000
52  IPTR(I,K)=KS1                                                    00471000
COS1=0                                                                00472000
COS2=0                                                                00473000
COS3=0                                                                00474000
LT=1                                                                00475000
DO 21 J=1,LL                                                         00476000
K1=J+1                                                                00477000
IF (LINEA(K1).GE.0) GO TO 396                                         00478000
LINEA(K1)=-LINEA(K1)                                                 00479000
K1=LT                                                                00480000
396 LIB2=LIB2+1                                                       00481000
IF(LINEA(J).GT.LINEA(K1))GO TO 30                                     00482000
IRE(LIB2,1)=LINEA(J)                                                 00483000
IRE(LIB2,2)=LINEA(K1)                                                 00484000
IORD1(LIB2)=1                                                         00485000
GO TO 20                                                             00486000
30  IRE(LIB2,1)=LINEA(K1)                                             00487000
IRE(LIB2,2)=LINEA(J)                                                 00488000
IORD1(LIB2)=0                                                         00489000
20  IR(LIB2)=I                                                         00490000
IF (LT.NE.K1) GO TO 21                                               00491000
K3=LINEA(LT)                                                         00492000
LT1=LT+1                                                             00493000
DO 50 K=LT1,J                                                         00494000
K2=LINEA(K)                                                         00495000
DO 50 K1=1,3                                                         00496000
50  XYZ(K-1,K1)=PUNT(K2,K1)-PUNT(K3,K1)                             00497000
MAX=J-2                                                             00498000
DO 51 K=LT,MAX                                                       00499000
K1=K+1                                                                00500000
COS1=XYZ(K,2)*XYZ(K1,3)-XYZ(K,3)*XYZ(K1,2)+COS1                   00501000
COS2=XYZ(K,3)*XYZ(K1,1)-XYZ(K,1)*XYZ(K1,3)+COS2                   00502000
COS3=XYZ(K,1)*XYZ(K1,2)-XYZ(K,2)*XYZ(K1,1)+COS3                   00503000

```



```

      ICONV(I)=-1                                00560000
      IR(1)=KS1                                  00561000
      IR(2)=KS2                                  00562000
      DO 139 K1=1,2                              00563000
      KS=IR(K1)                                  00564000
      IF (ISIPU(KS).NE.0) GO TO 139               00565000
      ISIPU(KS)=LIB                             00566000
      ISI=LIB                                    00567000
      IPPCC(LIB)=IREFA(I,1)                     00568000
      IPPCC(LIB+1)=IREFA(I,2)                   00569000
      LIB=LIB+2                                  00570000
      DO 130 K2=1,2                              00571000
      DO 130 K3=1,NR                             00572000
      IF (IREPU(K3,K2).NE.KS) GO TO 130          00573000
      MAX=LIB-1                                  00574000
      DO 129 K5=1,2                              00575000
      ITS=IREFA(K3,K5)                          00576000
      DO 125 K4=ISI,MAX                         00577000
      IF (ITS.EQ.IPPCC(K4)) GO TO 129            00578000
125  CONTINUE                                  00579000
      IPPCC(LIB)=ITS                            00580000
      LIB=LIB+1                                  00581000
129  CONTINUE                                  00582000
130  CONTINUE                                  00583000
      IPPCC(LIB)=-1                             00584000
      LIB=LIB+1                                  00585000
139  CONTINUE                                  00586000
60   CONTINUE                                  00587000
      IF ((NP+NF).NE.(NR+2)) IES=0              00588000
      RETURN                                     00589000
      END                                         00590000
C                                               00591000
C                                               00592000
C----- 00593000
C SUBROUTINE HIDLI - INPUT INTERFACE FOR HIDDEN-LINE REMOVAL. 00594000
C----- 00595000
      SUBROUTINE HIDLI(*)                        00596000
      COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF, 00597000
      1NEXTP,NEXTL,ITP,H1,NTRAT                 00598000
      COMMON PUPRVI(350,3),PUNTRA(350,3),INFACP(5000), 00599000
      1IREPU(500,2),IREFA(500,2),IPTR(250,3),SIPA(250),ICONV(500), 00600000
      2NP,NF,NR,LIB,IES,X0,Y0,Z0,T0,ISIPU(350),IPPC(1000) 00601000
      COMMON/MATRIX/DRAT/PDATA/L,OPT,RX1,RX2,RY1,RY2/MISCD/IDT,IDEV 00602000
      LOGICAL INSAV,NODRAT,HID,H1,OPT           00603000
      DIMENSION HDRAT(4,4),VE(4),VU(4),DRAT(4,4) 00604000
      REAL RY(4,4)/2*0.,1.,0.,1.,4*0.,1.,5*0.,1./ 00605000
      REAL RZ(4,4)/0.,1.,2*0.,-1.,5*0.,1.,4*0.,1./ 00606000
      NP=NEXTP                                  00607000
      NF=ITP                                     00608000
      T0=0.                                     00609000
      X0=0.                                     00610000
      Y0=0.                                     00611000
      Z0=0.                                     00612000
      IF (IDT.EQ.1.OR.IDT.EQ.4) GO TO 100        00613000
      CALL COMB(DRAT,RZ,HDRAT)                  00614000
      IF (.NOT.OPT) GO TO 30                    00615000

```


	DD=HDRAT(4,1)	00616000
	HDRAT(4,1)=0.	00617000
	RRA=D/DD	00618000
	CALL RSCA(RRA)	00619000
30	TO=1.	00620000
	X0=-D	00621000
	IF(OPT)X0=-DD	00622000
	DO 10 I=1,NP	00623000
	DO 12 J=1,3	00624000
12	VE(J)=PUNT(I,J)	00625000
	CALL PROVM(VE,HDRAT,VU,3)	00626000
	IF(VU(1).LE.X0)RETURN	00627000
	PERS=1+VU(1)/D	00628000
	IF(OPT)PERS=1+VU(1)/DD	00629000
	DO 15 J=1,3	00630000
	PUPRVI(I,J)=VU(J)	00631000
15	PUNTRA(I,J)=VU(J)/PERS	00632000
10	CONTINUE	00633000
	GO TO 200	00634000
100	CALL COMB(DRAT,RY,HDRAT)	00635000
	DO 20 I=1,NP	00636000
	DO 21 J=1,3	00637000
21	VE(J)=PUNT(I,J)	00638000
	CALL PROVM(VE,HDRAT,VU,3)	00639000
	DO 25 J=1,3	00640000
	PUPRVI(I,J)=VU(J)	00641000
25	PUNTRA(I,J)=VU(J)	00642000
20	CONTINUE	00643000
	X0=1.	00644000
200	IF(H1)CALL ENTRA	00645000
	H1=.FALSE.	00646000
	CALL CELATE	00647000
	RETURN 1	00648000
	END	00649000
C		00650000
C		00651000
C	-----	00652000
C	SUBROUTINE HVB - OUTPUT INTERFACE FOR HIDDEN-LINE REMOVAL.	00653000
C	-----	00654000
	SUBROUTINE HVB(X,Y,I)	00655000
	COMMON/MISCD/IDT	00656000
	IF(IDT.EQ.1.OR.IDT.EQ.4)GO TO 10	00657000
	X1=X	00658000
	Y1=Y	00659000
	X1=X1*F	00660000
	Y1=Y1*F	00661000
5	IF(I.EQ.1)GO TO 6	00662000
	XS=X1	00663000
	YS=Y1	00664000
	RETURN	00665000
6	CALL RAST(XS,YS,X1,Y1)	00666000
	XS=X1	00667000
	YS=Y1	00668000
	RETURN	00669000
10	CALL VBLD(X,Y,I)	00670000
	ENTRY RSCA(F)	00671000

```

      RETURN
      END
C
C
C-----
C LOGICAL FUNCTION INOUT - TO DECIDE IF A POINT IS INSIDE
C                      A POLYGON.
C-----
      LOGICAL FUNCTION INOUT(N,P,X,Y)
      INTEGER TOT
      REAL P(200,2)
C
C_COPY FIRST POINT TO END OF ARRAY
      P(N+1,1)=P(1,1)
      P(N+1,2)=P(1,2)
      INOUT=.TRUE.
C
C_TOT = NUMBER OF INTERSECTIONS ABOVE POINT(X,Y)
      TOT=0
C
C_DOES A VERTICAL LINE THROUGH POINT(X,Y)
C_INTERSECT ANY OF THE POLYGON SIDES?
      DO 10 I=1,N
        IF(P(I,1).EQ.X .OR. P(I+1,1).EQ.X)X=X+2
        IF((P(I,1).LT.X .AND. P(I+1,1).GT.X) .OR.
          1 (P(I,1).GT.X .AND. P(I+1,1).LT.X))GO TO 20
        GO TO 10
      20 IF(VERDIS(X,Y,P(I,1),P(I,2),P(I+1,1),P(I+1,2)).GT.0.)
        1 TOT=TOT+1
      10 CONTINUE
C
C_IS TOT ODD OR EVEN?
      IF((TOT/2)*2 .EQ. TOT)INOUT=.FALSE.
      RETURN
      END
C
C
C
      SUBROUTINE KDE1 (A,ALPHA,N1,N2,N3,DET1)
      DIMENSION A(4,4)
      DO 10 I=1,3
      DO 10 J=1,3
10  A(I,J)=A(I,J)-A(4,J)
      DET1=A(1,1)*(A(2,2)*A(3,3)-A(2,3)*A(3,2))-A(1,2)*(A(2,1)*A(3,3)-A(
12,3)*A(3,1))+A(1,3)*(A(2,1)*A(3,2)-A(2,2)*A(3,1))
      RETURN
      END
C
C
C
      SUBROUTINE PROVM(VE,AME,VU,NC)
      DIMENSION VE(4),VU(4),AME(4,4),V(4)
      4 NM=NC+1
      VE(NM)=1
      DO 1 I=1,NM
      V(I)=0.

```

	DO 1 J=1,NM	00728000
1	V(I)=V(I)+AME(J,I)*VE(J)	00729000
	DO 2 I=1,NC	00730000
2	VU(I)=V(I)	00731000
	RETURN	00732000
	END	00733000
C	-----	00734000
C	FUNCTION VERDIS - THE VERTICAL DISTANCE FROM POINT X,Y	00735000
C	TO LINE X1,Y1,X2,Y2 .	00736000
C	-----	00737000
	FUNCTION VERDIS(X,Y,X1,Y1,X2,Y2)	00738000
	VERDIS=((Y2-Y1)/(X2-X1))*(X-X1)+(Y1-Y)	00739000
	RETURN	00740000
	END	00741000

C-----	00001000
SUBROUTINE CKABTN(I)	00002000
C-----	00003000
CALL CKGCUR(I,X,Y)	00004000
RETURN	00005000
END	00006000
C-----	00007000
SUBROUTINE CKMA (X,Y,ITR)	00008000
C-----	00009000
C	00010000
C MOVE TO A NEW POINT	00011000
C	00012000
DIMENSION CKRINF(9,12)	00013000
COMMON /CKRI/ BASE, CKRINF, ICKRAC	00014000
CALL NUMARG(NARG)	00015000
IX = CKRINF(6,ICKRAC) + X*CKRINF(8,ICKRAC) + 0.5	00016000
IY = CKRINF(7,ICKRAC) + Y*CKRINF(9,ICKRAC) + 0.5	00017000
IF (NARG .EQ. 2) GO TO 10	00018000
CALL CKDM(IX,IY,0,ITR)	00019000
GO TO 20	00020000
10 CALL CKDM(IX,IY,0,0)	00021000
20 RETURN	00022000
END	00023000
C-----	00024000
SUBROUTINE CKVA(X,Y,ITR)	00025000
C-----	00026000
C	00027000
C INTERFACE TO DRAW TO A NEW POINT	00028000
C	00029000
CALL NUMARG(NARG)	00030000
IF (NARG .EQ. 2) GO TO 10	00031000
CALL CKDA(X,Y,ITR)	00032000
GO TO 20	00033000
10 CALL CKDA(X,Y,0)	00034000
20 RETURN	00035000
END	00036000
C	00037000
C	00038000
C-----	00039000
C SUBROUTINE DINI - TO SET UP OUTPUT DEVICES.	00040000
C-----	00041000
SUBROUTINE DINI(INT)	00042000
LOGICAL INIT,FRA,INT	00043000
COMMON /SIZE/ XM(6),YM(6)	00044000
COMMON /MISC/ IDT,IDEV	00045000
LOGICAL#1 TR,FA	00046000
TR=.TRUE.	00047000
FA=.FALSE.	00048000
INT=.FALSE.	00049000
XM(1)=8.25	00050000
YM(1)=6.4	00051000
XM(2)=7.0	00052000
YM(2)=6.0	00053000
XM(3)=100.0	00054000
YM(3)=10.0	00055000

	<u>XM(4)=8.5</u>	00056000
	<u>YM(4)=8.5</u>	00057000
	<u>XM(5)=14.5</u>	00058000
	<u>YM(5)=10.25</u>	00059000
	RETURN	00060000
C		00061000
C	C_CALLS OUTPUT DEVICES.	00062000
	ENTRY DEVICE(INIT,FRA)	00063000
	GO TO (1,2,3,4,5,1,2,3,4),IDEV	00064000
	1 CALL COMP(INIT,FRA,&10)	00065000
	2 CALL GT40(INIT,FRA,&10)	00066000
	3 CALL UMPL(INIT,FRA,&10)	00067000
	4 CALL NEWD(INIT,FRA,&10)	00068000
	5 CALL CHRO(INIT,FRA,&10)	00069000
10	RETURN	00070000
	END	00071000
C		00072000
C		00073000
C	-----	00074000
C	SUBROUTINE ERROR - CENTRAL ERROR PROCESSOR.	00075000
C	-----	00076000
	SUBROUTINE ERROR(I,J,K,A,B)	00077000
	REAL*8 AA	00078000
	LOGICAL BA,BAT/F/	00079000
	PRINT 100,I,J,K,A,B	00080000
	IF(BAT) CALL SYSTEM	00081000
100	FORMAT(' ERROR NUMBER',I3,2I5,F7.3,A4)	00082000
	RETURN	00083000
	ENTRY ERR(BA,*)	00084000
	BAT=BA	00085000
	RETURN 1	00086000
	ENTRY ERRORS(I,J,AA,*)	00087000
	PRINT 101,I,J,AA	00088000
101	FORMAT(' ERROR NUMBER',2I5,5X,A8)	00089000
	IF(BAT) CALL SYSTEM	00090000
	RETURN 1	00091000
	END	00092000
	SUBROUTINE FREAD (IUNIT,*,*)	00093000
C		00094000
C	SUBROUTINE FREAD IS USED TO READ A RECORD FROM THE SPECIFIED	00095000
C	UNIT AND INITIALIZE THE DECODING ROUTINES, WHERE	00096000
C		00097000
C	IUNIT - THE FORTRAN UNIT NUMBER FROM WHICH THE RECORD	00098000
C	IS TO BE READ.	00099000
C		00100000
C	* - THE ALTERNATE RETURN POINT IN CASE OF ERROR	00101000
C		00102000
C	* - THE ALTERNATE RETURN POINT IC CASE OF END OF FILE	00103000
C		00104000
C	NOTE: IF THE FIRST 4 CHARACTERS OF ANY INPUT LINE IS 'ATTN'	00105000
C	AN ATTENTION WILL BE SIGNALLED AND CONTROL WILL RETURN TO	00106000
C	THE POINT INDICATED BY THE LAST CALL TO CKATTN.	00107000
C		00108000
	INTEGER CARD(20), BLANK, ATTN	00109000
	LOGICAL*1 LVAR(1),LVAL(1)	00110000
	DATA BLANK/' '/, ATTN/'ATTN'/	00111000

	READ(IUNIT,100,END=110) CARD	00112000
100	FORMAT(20A4)	00113000
	IF (CARD(1) .EQ. ATTN) CALL ATTNEX	00114000
	CALL <u>FRINIT</u> (CARD,80,&910)	00115000
	GO TO 900	00116000
110	DC 120 I=1,20	00117000
120	CARD(I)=BLANK	00118000
	REWIND IUNIT	00119000
	CALL <u>FRINIT</u> (CARD,80,&910)	00120000
	GO TO 940	00121000
C		00122000
	ENTRY APARM(LVAR,LVAL,LEN,*)	00123000
C		00124000
C	ENTRY APARM WILL DECODE THE NEXT PARAMETER ON THE INPUT RECORD AS	00125000
C	FOUR CHARACTER STRING, WHERE	00126000
C		00127000
C	LVAR - IS THE VALUE TO BE RETURNED TO THE CALLING ROUTINE	00128000
C		00129000
C	LVAL - IS THE VALUE TO BE ASSIGNED IF THE VALUE IS MISSING	00130000
C		00131000
C	LEN - IS THE LENGTH OF THE STRING TO BE RETURNED	00132000
C		00133000
C	* - IS THE ALTERNATE RETURN POINT IN CASE OF ERROR	00134000
C		00135000
	CALL <u>FRWORD</u> (LVAR,LEN,DUM,&200,&910)	00136000
	GO TO 900	00137000
200	DO 210 I=1,LEN	00138000
210	LVAR(I)=LVAL(I)	00139000
	GO TO 900	00140000
C		00141000
	ENTRY IPARM(IVAR,IVAL,*)	00142000
C		00143000
C	ENTRY IPARM WILL DECODE THE NEXT PARAMETER AS AN INTEGER,WHERE	00144000
C		00145000
C	IVAR - IS THE VALUE TO BE RETURNED	00146000
C		00147000
C	IVAL - IS THE DEFAULT VALUE IS VALUE IS MISSING	00148000
C		00149000
C	* - IS THE ALTERNATE RETURN POINT IN CASE OF ERROR	00150000
C		00151000
	CALL <u>FRINTG</u> (IVAR,&300,&910)	00152000
	GO TO 900	00153000
300	IVAR=IVAL	00154000
	GO TO 900	00155000
C		00156000
	ENTRY RPARM(RVAR,RVAL,*)	00157000
C		00158000
C	ENTRY RPARM WILL DECODE THE NEXT PARAMETER AS A REAL VARIABLE,	00159000
C	WHERE	00160000
C		00161000
C	RVAR - IS THE VALUE TO BE RETURNED TO THE CALLING ROUTINE	00162000
C		00163000
C	RVAL - IS THE DEFAULT VALUE WHEN THE PARAMTER IS MISSING	00164000
C		00165000
C	* - IS THE ALTERNATE RETURN POINT IN CASE OF ERROR	00166000
C		00167000

	CALL FRREAL(RVAR,&400,&910)	00168000
	GO TO 900	00169000
400	RVAR=RVAL	00170000
C		00171000
C	RETURN TO CALLING PROGRAM	00172000
C		00173000
900	RETURN	00174000
C		00175000
C	RETURN TO CALLING PROGRAM AFTER ERROR FROM PARSING ROUTINE	00176000
C		00177000
910	CALL ERRORS(15,0,'INPUT',&920)	00178000
920	RETURN 1	00179000
940	RETURN 2	00180000
	END	00181000
C		00182000
C		00183000
C	-----	00184000
C	SUBROUTINE JOY - MOVIE INPUT FOR THE TEKTRONIX.	00185000
C	-----	00186000
	SUBROUTINE JOY(M,NCUT,*)	00187000
	DIMENSION M(30)	00188000
	COMMON/B/XS	00189000
	COMMON/MATRIX/DRAT(4,4)	00190000
	COMMON/TABLES/NAME, GROUP/MISC/JT,KT,LT,L	00191000
	COMMON/JOHN/INIT,JCK	00192000
	COMMON/TUBE/X,Y,Z,XV,YV,ZV,NFR,CONE,OPT,N	00193000
	COMMON/VECTOR/DX(1000),DY(1000),ISW(1000),NMP,GOOD,FIRST	00194000
	COMMON/MISC/DIDT,IDEV,SF,FRA	00195000
	DIMENSION X(100),Y(100),Z(100)	00196000
	LOGICAL FIR,OPT	00197000
	LOGICAL*4 GOOD,INIT,STERSW	00198000
	DATA FIR/.TRUE./	00199000
	INTEGER NAME(30,5),GROUP(30,3)	00200000
	INTEGER OP	00201000
	STERSW=.FALSE.	00202000
	IF(IDT.EQ.6)STERSW=.TRUE.	00203000
	IDT=1	00204000
	PRINT 1	00205000
1	FORMAT('ENTER #FRAMES, CONE OF VISION:')	00206000
700	CALL FREAD(4,&700,&701)	00207000
701	CALL IPARM(NFR,6,&700)	00208000
	CALL RPARM(CONE,60.0,&700)	00209000
	IF(NFR.LE.1)NFR=6	00210000
	PRINT 3	00211000
3	FORMAT('ENTER VIEWING OPTION(0=POINT,1=AHEAD):')	00212000
710	CALL FREAD(4,&710,&711)	00213000
711	CALL IPARM(OP,0,&710)	00214000
	OPT=.FALSE.	00215000
	IF(OP.NE.0) OPT=.TRUE.	00216000
20	PRINT 5	00217000
5	FORMAT('ENTER MAX DISTANCE FROM ORIGIN:')	00218000
720	CALL FREAD(4,&720,&721)	00219000
721	CALL RPARM(DIS,100.0,&720)	00220000
	IF(DIS.LE.0.)DIS=100.	00221000
	IF(FIR)GO TO 21	00222000
	CALL OTT(NUM,1,-DIS,-DIS,0.)	00223000

	IF(II.EQ.89 .OR. II.EQ.121)GO TO 94	00280000
	IF(II.EQ.27 .OR. II.EQ.5)GO TO 400	00281000
C		00282000
C		00283000
	CALL CKMA(660.,670.,1)	00284000
	PRINT 140	00285000
140	FORMAT('CONSTANT Z(Y,N)?')	00286000
	CALL CKABTN(IIS)	00287000
	IF(IIS.EQ.89 .OR. IIS.EQ.121)GO TO 200	00288000
	IF(IIS.EQ.27 .OR. IIS.EQ.5)GO TO 400	00289000
	GOOD=.FALSE.	00290000
	INIT=.FALSE.	00291000
	DO 145 IR=1,4	00292000
	DO 145 JR=1,4	00293000
145	DRAT(IR,JR)=0.0	00294000
	DRAT(1,1)=1.	00295000
	DRAT(2,3)=-1.	00296000
	DRAT(3,2)=1.	00297000
	DRAT(4,4)=1.	00298000
	YDISW=DIS	00299000
	YDISJ=DIS	00300000
	DRAT(4,2)=YDISJ	00301000
	DO 148 K=1,NNOUT	00302000
	IF(GROUP(M(K),3).EQ.0)GO TO 148	00303000
	CALL BLDMAT(M(K))	00304000
148	CONTINUE	00305000
	CALL PUTOUT	00306000
	GOOD=.TRUE.	00307000
	INIT=.TRUE.	00308000
150	CONTINUE	00309000
	CALL CKMA(660.,650.,1)	00310000
	PRINT 160	00311000
160	FORMAT('ENTER EACH Z:')	00312000
	YDISJ=(YDISJ/.008*XS)+(YDISW/.008*XS)	00313000
	DO 162 I=1,N	00314000
	CALL CKMA(X(I),Y(I),1)	00315000
	CALL CKVA(X(I),800.,1)	00316000
	IF(I.GT.1)GO TO 161	00317000
	CALL CKMA(X(I),YDISJ,1)	00318000
	GO TO 163	00319000
161	CALL CKMA(X(I),Z(I-1),1)	00320000
163	CALL CKGCUR(II,DUMB,Z(I))	00321000
	IF(II.EQ.27 .OR. II.EQ.5)GO TO 400	00322000
162	CONTINUE	00323000
	DO 164 I=1,N	00324000
	Z(I)=Z(I)-YDISJ	00325000
164	CONTINUE	00326000
	GO TO 250	00327000
200	CALL CKMA(660.,650.,1)	00328000
	PRINT 210	00329000
210	FORMAT('ENTER Z:')	00330000
740	CALL FREAD(4,&740,&741)	00331000
741	CALL RPARM(ZZ,0.0,&740)	00332000
	DO 230 K=1,N	00333000
230	Z(K)=ZZ	00334000
C		00335000

CALL OTS(NUM,1,2.*DIS,2.*DIS,DIS/2.)	00224000
GO TO 22	00225000
21 CALL GRIN(1,'CUBE',-DIS,-DIS,0.,2.*DIS,2.*DIS,DIS/2.,'X ',0.,&47	00226000
1)	00227000
47 FIR=.FALSE.	00228000
NUM=L	00229000
22 DO 70 IR=1,4	00230000
DO 71 JR=1,4	00231000
71 DRAT(IR,JR)=0.	00232000
70 DRAT(IR,IR)=1.	00233000
48 GROUP(NUM,3)=1	00234000
M(NOUT+1)=NUM	00235000
NNOUT=NOUT	00236000
IF(NOUT.NE.L)NNOUT=NOUT+1	00237000
DO 23 K=1,NNOUT	00238000
IF(GROUP(M(NNOUT-K+1),3).EQ.0) GO TO 23	00239000
CALL BLDMAT(M(NNOUT-K+1))	00240000
23 CONTINUE	00241000
CALL PUTOUT	00242000
CALL CKMA(96.,20.,1)	00243000
PRINT 31	00244000
31 FORMAT('Y=YES N=NO E=END')	00245000
CALL CKMA(660.,750.,1)	00246000
PRINT 25	00247000
25 FORMAT('IS THE DISTANCE OK(Y,N)?')	00248000
CALL CKABTN(II)	00249000
IF(II.EQ.78 .OR. II.EQ.110)GO TO 26	00250000
IF(II.EQ.27.OR.II.EQ.5)GO TO 400	00251000
C	00252000
C	00253000
94 CALL CKMA(660.,730.,1)	00254000
95 CONTINUE	00255000
PRINT 100	00256000
100 FORMAT('TO MARK PATH LOCATE')	00257000
CALL CKMA(660.,710.,1)	00258000
PRINT 105	00259000
105 FORMAT('CURSOR AND PRESS A KEY')	00260000
N=0	00261000
CALL CKMA(300.,300.,1)	00262000
107 CALL CKGCR(II,XT,YT)	00263000
IF(II.EQ.69 .OR. II.EQ.101)GO TO 110	00264000
IF(II.EQ.27 .OR. II.EQ.5)GO TO 400	00265000
N=N+1	00266000
X(N)=XT	00267000
Y(N)=YT	00268000
IF(N.EQ.1)GO TO 107	00269000
CALL CKMA(X(N-1),Y(N-1),1)	00270000
CALL CKVA(X(N),Y(N),1)	00271000
GO TO 107	00272000
26 CALL CKDER	00273000
GO TO 20	00274000
110 CONTINUE	00275000
CALL CKMA(660.,690.,1)	00276000
PRINT 120	00277000
120 FORMAT('WISH NEW PATH(Y,N)?')	00278000
CALL CKABTN(II)	00279000

C		00336000
250	IF(OPT) GO TO 300	00337000
	CALL CKMA(660.,630.,1)	00338000
	PRINT 260	00339000
260	FORMAT('ENTER VIEWED POINT:')	00340000
	CALL CKMA(300.,300.,1)	00341000
	CALL CKGCUR(II,XV,YV)	00342000
	IF(II.EQ.27 .OR. II.EQ.5)GO TO 400	00343000
	YTEMP=YV	00344000
	XTEMP=XV	00345000
	XV=(.008*XV/XS)-DIS	00346000
	YV=(.008*YV/XS)-DIS	00347000
	IF(IIS.EQ.89 .OR. IIS.EQ.121)GO TO 271	00348000
	CALL CKMA(XTEMP,YTEMP,1)	00349000
	CALL CKVA(XTEMP,800.,1)	00350000
271	CALL CKMA(660.,610.,1)	00351000
	PRINT 270	00352000
270	FORMAT('ENTER Z VALUE:')	00353000
	IF(.NOT.(IIS.EQ.89 .OR. IIS.EQ.121))GO TO 272	00354000
750	CALL FREAD(4,&750,&751)	00355000
751	CALL RPARM(ZV,0.0,&750)	00356000
	DO 273 KKK=1,N	00357000
	X(KKK)=(.008*X(KKK)/XS)-DIS	00358000
273	Y(KKK)=(.008*Y(KKK)/XS)-DIS	00359000
	GO TO 400	00360000
272	CALL CKMA(XTEMP,YDISJ,1)	00361000
	CALL CKGCUR(II,DUMB,ZV)	00362000
	IF(II.EQ.27 .OR. II.EQ.5)GO TO 400	00363000
	ZV=ZV-YDISJ	00364000
	ZV=(ZV*.008/XS)	00365000
300	CONTINUE	00366000
	DO 310 KKK=1,N	00367000
	Z(KKK)=(.008*Z(KKK)/XS)	00368000
	X(KKK)=(.008*X(KKK)/XS)-DIS	00369000
310	Y(KKK)=(.008*Y(KKK)/XS)-DIS	00370000
400	GOOD=.TRUE.	00371000
	INIT=.TRUE.	00372000
	IDT=5	00373000
	IF(STERSW)IDT=6	00374000
	GROUP(NUM,3)=0	00375000
	CALL CKMA(660.,590.,1)	00376000
500	RETURN 1	00377000
	END	00378000
C		00379000
C-----		00380000
C	SUBROUTINE PLDNO - SET UP FORMAT	00381000
C	FOR INPUT FILE TO CALCOMP PLOTTER	00382000
C-----		00383000
C		00384000
	SUBROUTINE PLDNO(JUNIT)	00385000
	COMMON/JCLFLG/JCLON,NAR, LASTNR	00386000
	LOGICAL JCLON,NAR, LASTNR	00387000
	DATA IPT/0/,XMAX/0.0/,YMAX/0.0/	00388000
	DIMENSION X(4),Y(4),IPEN(4)	00389000
	IUNIT=JUNIT	00390000
	IPT=0	00391000

	XMAX=0.0	00392000
	YMAX=0.0	00393000
	IF(JCLON.AND.NAR) GO TO 60	00393.20
	IF(JCLON.AND..NOT.NAR) GO TO 70	00393.40
	IF(LASTNR.AND..NOT.NAR) GO TO 70	00394.20
	IF(.NOT.LASTNR.AND.NAR) GO TO 60	00394.40
	RETURN	00395000
C JCL	FOR NARROW PAPER	00395.20
60	LASTNR=.TRUE.	00395.40
	WRITE(IUNIT,50)	00396000
50	FORMAT('/GO EXEC PGM=HRDCPY,COND=(0,LT)'/	00397000
2	'//STEPLIB DD DSN=DS450.ARCHGRAF.LIB,DISP=SHR'//	00398000
3	'//PLOTLIB DD DSN=SYS1.PLOTTER.DYNALIB,DISP=SHR'//	00399000
4	'//FT06F001 DD SYSOUT=A'//	00400000
5	'//FT09F001 DD SYSOUT=A,DCE=RECFM=FA'//	00401000
6	'//PLOTTER DD UNIT=PLOTTER'//	00402000
7	'//FT05F001 DD *'//	00403000
	JCLON=.FALSE.	00404000
	RETURN	00405000
C JCL	FOR WIDE PAPER	00405.10
70	LASTNR=.FALSE.	00405.15
	WRITE(IUNIT,80)	00405.20
80	FORMAT('/GO EXEC PGM=HRDCPY,COND=(0,LT)'/	00405.25
2	'//STEPLIB DD DSN=DS450.ARCHGRAF.LIB,DISP=SHR'//	00405.30
3	'//PLOTLIB DD DSN=SYS1.PLOTTER.DYNALIB,DISP=SHR'//	00405.35
4	'//PLTOPT DD *'//	00405.40
5	' PAPERWIDTH=W#'//	00405.45
6	'//FT06F001 DD SYSOUT=A'//	00405.50
7	'//FT09F001 DD SYSOUT=A,DCE=RECFM=FA'//	00405.55
8	'//PLOTTER DD UNIT=PLOTTER'//	00405.60
9	'//FT05F001 DD *'//	00405.65
	JCLON=.FALSE.	00405.70
	RETURN	00405.75
	ENTRY PENDNS(X1,Y1,IPNUM)	00406000
	JPEN=2	00407000
	IF(IPNUM.NE.0) JPEN=(IPNUM*100)+JPEN	00408000
	GO TO 10	00409000
	ENTRY FENUPS(X1,Y1,IPNUM)	00410000
	JPEN=3	00411000
	IF(IPNUM.NE.0) JPEN=(IPNUM*100)+JPEN	00412000
10	IPT=IPT+1	00413000
	X(IPT)=X1	00414000
	Y(IPT)=Y1	00415000
	IPEN(IPT)=JPEN	00416000
	IF (X1 .GT. XMAX) XMAX=X1	00417000
	IF (Y1 .GT. YMAX) YMAX=Y1	00418000
	IF (IPT .GE. 4) GO TO 30	00419000
	RETURN	00420000
	ENTRY PLTEND	00421000
	IPT=IPT+1	00422000
	X(IPT)=XMAX	00423000
	Y(IPT)=YMAX	00424000
	IPEN(IPT)=-3	00425000
	PRINT 20, XMAX,YMAX	00426000
20	FORMAT(' PLOT COMPLETE: XMAX =',F6.2,', YMAX =',F6.2)	00427000
30	WRITE(IUNIT,40) (X(I),Y(I),IPEN(I),I=1,IPT)	00428000

40	FORMAT(4(2F8.4,I4))	00429000
	IPT=0	00430000
	RETURN	00431000
	END	00432000
C		00433000
C-----		00434000
	SUBROUTINE SYSTEM	00435000
C-----		00436000
	STOP	00437000
	END	00438000

```

C                                                    00001000
C                                                    00002000
C-----
C SUBROUTINE EYESP - TO CALCULATE EYE COORDINATES FOR STEREO. 00003000
C-----
C SUBROUTINE EYESP(XV,YV,XB,YB,DX,DY) 00004000
C-----
C SUBROUTINE EYESP(XV,YV,XB,YB,DX,DY) 00005000
COMMON/SDATA/IOT,RTV,EYE 00006000
DD=SQRT((XB-XV)**2+(YB-YV)**2) 00007000
IF(DD.EQ.0.) GO TO 10 00008000
DX=EYE/2*(XB-XV)/DD 00009000
DY=EYE/2*(YB-YV)/DD 00010000
RETURN 00011000
10 DX=EYE/2.0 00012000
DY=0. 00013000
RETURN 00014000
END 00015000
C 00016000
C 00017000
C-----
C SUBROUTINE MOVIE - CONTROL SECTION FOR PRODUCING 00018000
C MULTIPLE PERSPECTIVES. 00019000
C-----
C SUBROUTINE MOVIE(M,NOUT) 00020000
COMMON /VECTOR/ XT(1000),YT(1000),ISG(1000),NSTOR,GOOD 00021000
COMMON /TABLES/ NAME,GROUP 00022000
COMMON/MISCD/IDT,IDEV,SF,FRA 00023000
COMMON/PDATA/DP,OPTS,RX1,RX2,RY1,RY2/SDATA/IOT,RTV,EYE 00024000
COMMON/TUBE/X,Y,Z,XV,YV,ZV,NFR,CONE,OPT,N 00025000
COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF, 00026000
1NEXTP,NEXTL,ITP,H1,NTRAT 00027000
COMMON/COLOUR/ICHPN(31,2),ICOL(30) 00028000
LOGICAL NODRAT,INSAV,HID,H1 00029000
DIMENSION ANSW(2) 00030000
DATA ANSW/'YES','NO'/ 00031000
LOGICAL FIRST,GOOD,OPTS,RTV,FRA 00032000
INTEGER NAME(30,5),GROUP(30,3),M(30) 00033000
REAL INC,X(100),Y(100),Z(100),D(99),STOP/'STOP'/ 00034000
LOGICAL PT,OPT 00035000
9 FIRST=.TRUE. 00036000
IF(IDEV .EQ. 3)GO TO 73 00037000
C 00038000
C 00039000
C 00040000
PRINT 51 00041000
51 FORMAT('WISH TO USE GRAPHIC INPUT(YES,NO)?') 00042000
700 CALL FREAD(4,&700,&701) 00043000
701 CALL APARM(ANS,'YES ',4,&700) 00044000
IF(ANS.EQ.ANSW(2)) GO TO 73 00045000
GO TO (71,72,73,74),IDEV 00046000
71 CALL JOY(M,NOUT,&40) 00047000
72 CALL GT40IN(M,NOUT,&40) 00048000
74 CALL NEWIN(M,NOUT,&40) 00049000
C 00050000
C 00051000
73 PRINT 1 00052000
1 FORMAT('ENTER #FRAMES, #POINTS, CONE OF VISION:') 00053000
00054000
00055000

```

```

710 CALL FREAD(4,&710,&711)
711 CALL IPARM(NFR,6,&710)
    CALL IPARM(N,6,&710)
    CALL RPARM(CONE,60.0,&710)
    IF(NFR.LE.0)NFR=6
    IF(N .LE.0)N =6
C
C
    PRINT 100
100 FORMAT('ENTER VIEWING OPTION(0=POINT,1=AHEAD):')
720 CALL FREAD(4,&720,&721)
721 CALL IPARM(OP,0,&720)
    OPT=.FALSE.
    IF(OP.NE.0) OPT=.TRUE.
5   DO 10 I=1,N
    PRINT 3,I
3   FORMAT('ENTER POINT ',I2,' XYZ:')
730 CALL FREAD(4,&730,&731)
731 CALL RPARM(X(I),0.0,&730)
    CALL RPARM(Y(I),0.0,&730)
    CALL RPARM(Z(I),0.0,&730)
10  CONTINUE
    IF(OPT) GO TO 40
C
C
    PRINT 6
6   FORMAT('ENTER XYZ OF VIEWED POINT:')
740 CALL FREAD(4,&740,&741)
741 CALL RPARM(XV,0.0,&740)
    CALL RPARM(YV,0.0,&740)
    CALL RPARM(ZV,0.0,&740)
40  N1=N-1
    DT=0.
    XC=0.
    YC=0.
    DO 20 I=1,N1
    D(I)=SQRT((X(I+1)-X(I))**2+(Y(I+1)-Y(I))**2+(Z(I+1)-Z(I))**2)
20  DT=DT+D(I)
    GO TO 21
C
C
    ENTRY SAME(*)
    FIRST=.TRUE.
    PRINT 101
101 FORMAT('ENTER #FRAMES:')
750 CALL FREAD(4,&750,&751)
751 CALL IPARM(NFR,0,&750)
    IF(NFR.LE.1)NFR=6
21  CONTINUE
    INC=DT/(NFR-1)
    XN=X(1)
    YN=Y(1)
    ZN=Z(1)
    REM=G.
    NF=0
    DO 30 J=1,N1

```

IN=(D(J)+REM)/INC +.01	00112000
IF(IN.LE.0) GO TO 30	00113000
XD=(X(J+1)-X(J))/D(J)*INC	00114000
YD=(Y(J+1)-Y(J))/D(J)*INC	00115000
ZD=(Z(J+1)-Z(J))/D(J)*INC	00116000
XR=(INC-REM)*XD/INC	00117000
YR=(INC-REM)*YD/INC	00118000
ZR=(INC-REM)*ZD/INC	00119000
C	00120000
C_LET THE PERSPECTIVE ROUTINE CALCULATE THE DRAWING MATRICES.	00121000
DO 25 I=1,IN	00122000
XB=X(J)+XD*(I-1)+XR	00123000
YB=Y(J)+YD*(I-1)+YR	00124000
ZB=Z(J)+ZD*(I-1)+ZR	00125000
45 NF=NF+1	00126000
IF(IDT.NE.6)GOTO 60	00127000
RTV=.FALSE.	00128000
63 IF(OPT)GOTO 62	00129000
CALL EYESP(XN,YN,XV,YV,DX,DY)	00130000
XC=-DX	00131000
YC=DY	00132000
GO TO 54	00133000
62 CALL EYESP(XN,YN,XB,YB,DX,DY)	00134000
XC=-DX	00135000
YC=DY	00136000
GO TO 50	00137000
60 IF(OPT) GO TO 50	00138000
54 CALL PERS1(XN+XC,YN+YC,ZN,XV+XC,YV+YC,ZV,5.,5.,CONE,&23)	00139000
GO TO 9	00140000
50 CALL PERS1(XN+XC,YN+YC,ZN,XB+XC,YB+YC,ZB,5.,5.,CONE,&23)	00141000
23 IF(HID)CALL HIDLI(&200)	00142000
DO 201 K=1,NOUT	00143000
IF(GROUP(M(K),3).NE.0) GO TO 202	00144000
201 CONTINUE	00145000
202 ICHPN(I,1)=1	00146000
DO 24 K=1,NOUT	00147000
IF(GROUP(M(K),3).EQ.0) GO TO 24	00148000
CALL BLDMAT(M(K))	00149000
ICHPN(K+1,1)=NSTOR+1	00150000
ICHPN(K,2)=ICOL(M(K))	00151000
24 CONTINUE	00152000
200 IF(IDT.EQ.6.AND..NOT.RTV)GO TO 81	00153000
IF(.NOT.GOOD.AND.FIRST)GO TO 83	00154000
GOOD=.FALSE.	00155000
IF(FIRST) GOOD=.TRUE.	00156000
83 FIRST=.FALSE.	00157000
81 IF(IDT.NE.6)GO TO 66	00158000
IF(RTV)GO TO 67	00159000
RTV=.TRUE.	00160000
XC=DX	00161000
YC=-DY	00162000
GO TO 63	00163000
67 RXO=RX1	00164000
RYO=RY1	00165000
IF(IOT.GE.3)GO TO 68	00166000
RX2=RX2+9.2	00167000

	GO TO 80	00168000
68	RX2=RY2*3	00169000
	RX1=-RY2	00170000
	RY1=-4.6	00171000
	RY2=4.6	00172000
80	CALL PUTOUT	00173000
	RX1=RX0	00174000
	RX2=-RX0	00175000
	RY1=RY0	00176000
	RY2=-RY0	00177000
	GO TO 69	00178000
66	CALL PUTOUT	00179000
69	IF(NF.NE.NFR-1) GO TO 26	00180000
	XN=X(N)	00181000
	XB=XN+XD	00182000
	YN=Y(N)	00183000
	YB=YN+YD	00184000
	ZN=Z(N)	00185000
	ZB=ZN+ZD	00186000
	GO TO 45	00187000
26	XN=XB	00188000
	YN=YB	00189000
	ZN=ZB	00190000
760	CALL FREAD(4,&760,&761)	00191000
761	CALL APARM(DUMB,'STOP',4,&760)	00192000
	IF(DUMB.EQ.STOP)GO TO 29	00193000
25	CONTINUE	00194000
30	REM=D(J)-INC*IN+REM	00195000
29	RETURN 1	00196000
	END	00197000
C		00198000
C		00199000
C	-----	00200000
C	SUBROUTINE PERS - TO CONSTRUCT PERSPECTIVE DRAWING MATRIX.	00201000
C	-----	00202000
	SUBROUTINE PERS(*)	00203000
	COMMON /PTS/PT,BAD,BACK	00204000
	LOGICAL PT,BAD	00205000
	LOGICAL*1 BACK	00206000
	COMMON/PDATA/D,OPT,RX1,RX2,RY1,RY2 /MATRIX/DRAT(4,4),MATG,MAT	00207000
	1/MISCD/IDT,IDEV,SF,RAS	00208000
	LOGICAL*4 RAS,OPT,SW	00209000
	REAL T(4,4),MATG(4,4),MAT(4,4)	00210000
	DATA PI/3.14159/	00211000
	SW=.FALSE.	00212000
	GO TO 100	00213000
C		00214000
C	THIS IS A STRANGE WAY TO ENTER THE PERSPECTIVE PARAMETERS.	00215000
10	PRINT 15	00216000
15	FORMAT('ENTER D,X,Y,Z,RZ,RX:')	00217000
	OPT=.FALSE.	00218000
700	CALL FREAD(4,&700,&701)	00219000
701	CALL RPARM(D,0.0,&700)	00220000
	CALL RPARM(X0,0.0,&700)	00221000
	CALL RPARM(Y0,0.0,&700)	00222000
	CALL RPARM(Z0,0.0,&700)	00223000

	CALL RPARM(RZ,0.0,&700)	00224000
	CALL RPARM(RX,0.0,&700)	00225000
	IF(D.EQ.0. .AND. XO.EQ.0. .AND.	00226000
2	YO.EQ.0. .AND. ZO.EQ.0. .AND.	00227000
3	RZ.EQ.0. .AND. RX.EQ.0.)GO TO 100	00228000
	PRINT 7	00229000
710	CALL FREAD(4,&710,&711)	00230000
711	CALL RPARM(VAN,0.0,&710)	00231000
	IF(VAN.LE.0)VAN=60.	00232000
	RX2=D*TAN(VAN*PI/360.)	00233000
	RX1=-RX2	00234000
	RY1=-RX2	00235000
	RY2=RX2	00236000
C		00237000
	C_BUILD PERSPECTIVE DRAWING MATRIX.	00238000
50	RZ=RZ*PI/180.	00239000
	RX=RX*PI/180.	00240000
	SX=SIN(RX)	00241000
	CX=COS(RX)	00242000
	SZ=SIN(RZ)	00243000
	CZ=CCS(RZ)	00244000
	DRAT(1,1)=CZ	00245000
	DRAT(1,2)=SZ*CX	00246000
	DRAT(1,3)=SZ*SX	00247000
	DRAT(1,4)=0.0	00248000
	DRAT(2,1)=-SZ	00249000
	DRAT(2,2)=CZ*CX	00250000
	DRAT(2,3)=CZ*SX	00251000
	DRAT(2,4)=0.0	00252000
	DRAT(3,1)=0.0	00253000
	DRAT(3,2)=-SX	00254000
	DRAT(3,3)=CX	00255000
	DRAT(3,4)=0.0	00256000
	DRAT(4,1)=XO	00257000
	DRAT(4,2)=YO	00258000
	DRAT(4,3)=ZO	00259000
	DRAT(4,4)=1.0	00260000
	PT=.TRUE.	00261000
	IF(.NOT.BACK) PT=.FALSE.	00262000
	IF(OPT) GO TO 30	00263000
	RETURN 1	00264000
	ENTRY PERS2(XV,YV,ZV,*)	00265000
22	XO=0.0	00266000
	YO=0.0	00267000
	ZO=0.0	00268000
	OPT=.TRUE.	00269000
	GO TO 50	00270000
C		00271000
	C_ADD THE PERSPECTIVE ELEMENTS TO THE DRAWING MATRIX.	00272000
30	CALL MATF(-XV,-YV,-ZV,0.,0.,0.,0.,MAT)	00273000
	CALL COMB(MAT,DRAT,T)	00274000
	DO 1000 I=1,4	00275000
	DO 1000 J=1,4	00276000
1000	DRAT(I,J)=T(I,J)	00277000
	IF(SW) GO TO 201	00278000
	PRINT 7	00279000

```

7      FORMAT('ENTER CONE OF VISION:')
720    CALL FREAD(4,&720,&721)
721    CALL RPARM(VAN,0.0,&720)
201    IF(VAN.LE.0)VAN=60.0
        D=RX2/TAN(VAN*PI/360.)
        RETURN 1
C
C_THIS IS THE NORMAL WAY TO ENTER THE PARAMETERS.
100    PRINT 101
101    FORMAT('ENTER XYZ OF VIEWER AND VIEWED POINT:')
730    CALL FREAD(4,&730,&731)
731    CALL RPARM(XV,0.0,&730)
        CALL RPARM(YV,0.0,&730)
        CALL RPARM(ZV,0.0,&730)
        CALL RPARM(XB,0.0,&730)
        CALL RPARM(YB,0.0,&730)
        CALL RPARM(ZB,0.0,&730)
        IF(XV.EQ.0. .AND. YV.EQ.0. .AND.
2      ZV.EQ.0. .AND. XB.EQ.0. .AND.
3      YB.EQ.0. .AND. ZB.EQ.0.)GO TO 10
        GO TO 200
C
CC ENTRY POINT FOR STEREO VIEWS
C
C ENTRY POINT FOR STEREO VIEWS.
        ENTRY PERS1(XV,YV,ZV,XB,YB,ZB,RW,RH,VAN,*)
        SW=.TRUE.
C
C_CALCULATE THE DISTANCE FROM EYE TO POINT.
200    HY=YB-YV
        HX=XB-XV
        H=HY**2+HX**2
        HZ=ZB-ZV
        TH=H+HZ**2
        IF(H.NE.0) GO TO 202
        RTZ=0.
        GO TO 203
C
C_DO EYE AND POINT COINCIDE?
202    RTZ=ARSIN(HX/SQRT(H))
203    IF(TH.NE.0) GO TO 204
        CALL ERROR(17,0,0,0.,'LOCA')
        IF(IDT.NE.5 .AND. IDT.NE.6) GO TO 100
        RETURN
C
C_EYE IS DIRECTLY ABOVE POINT.
204    RTX=-ARSIN(HZ/(SQRT(TH)))
        RZ=RTZ*180./PI
        RX=RTX*180./PI
        IF(HY.GE.0) GO TO 22
        RZ=180.0-RZ
        GO TO 22
        END
C
C
C-----

```

C SUBROUTINE ROT - TO CONSTRUCT ROTATION MATRIX.	00336000
C-----	00337000
SUBROUTINE ROT(*)	00338000
INTEGER*2 AXA(7) / 'Z','X','Y','Z','X','Y',' ' /, AXA	00339000
INTEGER ANS, ELV, CNTR	00340000
COMMON/MATRIX/DRAT(4,4), TEMP(4,4)	00341000
REAL T(4,4)	00342000
LOGICAL*1 SW/F/	00343000
SW=.FALSE.	00344000
CNTR=1	00345000
C	00346000
C_CHOOSE FORMAT FOR ROTATION	00347000
1001 PRINT 1002	00348000
1002 FORMAT (8X, '1-ELEVATION'/	00349000
2 8X, '2-ROTATION BY AXIS & DEGREES'/	00350000
3 8X, 'TYPE 1 OR 2'//)	00351000
1010 CALL FREAD(4, &1010, &1120)	00352000
1120 CALL IPARM(ANS, 2, &1010)	00353000
IF(ANS.EQ.2) GO TO 10	00354000
C	00355000
C_INPUT ELEVATION	00356000
1130 PRINT 1140	00357000
1140 FORMAT (8X, 'TYPE NUMBER FOR DESIRED ELEVATION'/	00358000
2 8X, '1-PLAN'/	00359000
3 8X, '2-NORTH'/	00360000
4 8X, '3-EAST'/	00361000
5 8X, '4-SOUTH'/	00362000
6 8X, '5-WEST'//)	00363000
1150 CALL FREAD(4, &1150, &1160)	00364000
1160 CALL IPARM(ELV, 0, &1150)	00365000
IF(ELV.EQ.0) GO TO 1001	00366000
1170 GO TO (1180, 1190, 1220, 1250, 1270), ELV	00367000
C	00368000
C_PLAN - NO ROTATION	00369000
1180 GO TO 52	00370000
C	00371000
C_NORTH ELEVATION - Y=180, X=-90	00372000
1190 GO TO (1200, 1210, 52), CNTR	00373000
1200 I=3	00374000
AN=0180.000	00375000
GO TO 71	00376000
1210 I=2	00377000
AN=-090.000	00378000
GO TO 71	00379000
C	00380000
C_EAST ELEVATION - Y=-90, X=-90	00381000
1220 GO TO (1230, 1240, 52), CNTR	00382000
1230 I=3	00383000
AN=-090.000	00384000
GO TO 71	00385000
1240 I=2	00386000
AN=-090.000	00387000
GO TO 71	00388000
C	00389000
C_SOUTH ELEVATION - X=-90	00390000
1250 GO TO (1260, 52), CNTR	00391000

1260 I=2	00392000
AN=-090.000	00393000
GO TO 71	00394000
C	00395000
C_WEST ELEVATION - Y=90, X=-90	00396000
1270 GO TO (1280,1290,52),CNTR	00397000
1280 I=3	00398000
AN=0090.000	00399000
GO TO 71	00400000
1290 I=2	00401000
AN=-090.000	00402000
GO TO 71	00403000
C	00404000
C_INPUT THE PARAMETERS.	00405000
10 IF(CNTR.EQ.4) GO TO 52	00406000
PRINT 1	00407000
1 FORMAT('AXIS, ANGLE?')	00408000
700 CALL FREAD(4,&700,&701)	00409000
701 CALL APARM(AXX,' ',2,&700)	00410000
CALL RPARM(AN,0.0,&700)	00411000
2 FORMAT(A2,F8.3)	00412000
C	00413000
C_VALID AXIS?	00414000
DO 70 I=1,6	00415000
IF(AXX.EQ.AXA(I)) GO TO 71	00416000
70 CONTINUE	00417000
IF(AXX.NE.AXA(7)) CALL ERRORS(21,0,'BAD AXIS',&10)	00418000
GO TO 52	00419000
C	00420000
C_BUILD TEMP MATRIX.	00421000
71 AX=MOD(I-1,3)+1	00422000
CALL MAT2(AX,AN,TEMP)	00423000
IF(SW) GO TO 30	00424000
C	00425000
C_COPY TEMP MATRIX INTO DRAWING MATRIX(DRAT)	00426000
DO 20 I=1,4	00427000
DO 20 J=1,4	00428000
20 DRAT(I,J)=TEMP(I,J)	00429000
SW=.TRUE.	00430000
CNTR=CNTR+1	00431000
IF(ANS.EQ.1) GO TO 1170	00432000
GO TO 10	00433000
C	00434000
C_COMBINE TEMP AND DRAT.	00435000
30 DO 50 I=1,4	00436000
DO 50 J=1,4	00437000
50 T(I,J)=DRAT(I,J)	00438000
CALL COMB(TEMP,T,DRAT)	00439000
CNTR=CNTR+1	00440000
IF(ANS.EQ.1) GO TO 1170	00441000
GO TO 10	00442000
C	00443000
C_NO ROTATION.	00444000
52 IF(SW) RETURN 1	00445000
DO 60 I=1,4	00446000
DO 60 J=1,4	00447000

	DRAT(I,J)=0.	00448000
60	DRAT(I,I)=1.	00449000
	RETURN 1	00450000
	END	00451000
C		00452000
C		00453000
C	-----	00454000
C	SUBROUTINE SOLAR - TO CONSTRUCT SOLAR ROTATION MATRIX.	00455000
C	-----	00456000
	SUBROUTINE SOLAR(*)	00457000
	COMMON/MATRIX/DRAT(4,4)	00458000
	REAL L,L1,YRDAY,HR,TEMP	00459000
	INTEGER ANSR,MO,DAY	00460000
	INTEGER MON(12,2)/31,28,31,30,31,30,31,31,30,31,30,31,	00461000
	10,31,59,90,120,151,181,212,243,273,304,334/	00462000
	DATA PI/3.14159265/	00463000
C		00464000
C	_ZERO OUT DRAWING MATRIX(DRAT).	00465000
	DO 5 I=1,4	00466000
	DO 5 J=1,4	00467000
5	DRAT(I,J)=0.	00468000
C		00469000
	PRINT 100	00470000
100	FORMAT('ENTER 1 OR 2'/	00471000
1	'1 - BY DECLINATION'/	00472000
2	'2 - BY DATE')	00473000
10	CALL FREAD(4,&10,&20)	00474000
20	CALL IPARM(ANSR,1,&10)	00475000
	IF(ANSR.NE.2) GO TO 70	00476000
30	PRINT 40	00477000
40	FORMAT('ENTER LATITUDE,MONTH,DAY,HOUR: '/	00478000
1	'(EX- 39.0 5 18 17)')	00479000
50	CALL FREAD(4,&50,&60)	00480000
60	CALL RPARM(L1,0.0,&50)	00481000
	CALL IPARM(MO,3,&50)	00482000
	CALL IPARM(DAY,21,&50)	00483000
	CALL RPARM(HR,12.0,&50)	00484000
	IF(L1.GT.90.OR.L1.LT.-90) CALL ERRORS(0,0,'BAD LAT',&30)	00485000
	IF(MO.GT.12.OR.MO.LT.1) CALL ERRORS(0,0,'BAD MO',&30)	00486000
	IF(DAY.GT.MON(MO,1).OR.DAY.LT.1)	00487000
	1CALL ERRORS(0,0,'BAD DAY',&30)	00488000
	YRDAY=FLOAT(MON(MO,2)+DAY)	00489000
	TEMP=((YRDAY-80.)/370.)*360.	00490000
	D1=23.45*SIN(TEMP*0.0174532)	00491000
	GO TO 90	00492000
C	_INPUT THE PARAMETERS.	00493000
70	PRINT 80	00494000
80	FORMAT('ENTER LATITUDE,DECLINATION,HOUR: '/	00495000
1	'(EX- 39.0 23.45 17)')	00496000
700	CALL FREAD(4,&700,&701)	00497000
701	CALL RPARM(L1,0.0,&700)	00498000
	CALL RPARM(D1,0.0,&700)	00499000
	CALL RPARM(HR,0.0,&700)	00500000
90	T1=HR-12.0	00501000
C		00502000
C	_BUILD DRAWING MATRIX.	00503000

```

      L=L1*PI/180.                                00504000
      D=D1*PI/180.                                00505000
      T=T1*15.*PI/180.                            00506000
      ST=SIN(T)                                    00507000
      CT=COS(T)                                    00508000
      SD=SIN(D)                                    00509000
      CD=COS(D)                                    00510000
      SL=SIN(L)                                    00511000
      CL=COS(L)                                    00512000
      DRAT(1,1)=CT                                 00513000
      DRAT(1,2)=(-ST)*(-SD)                        00514000
      DRAT(2,1)=(-SL)*(ST)                         00515000
      DRAT(2,2)=CL*CD+((-SD)*(-SL)*CT)            00516000
      DRAT(3,1)=CL*ST                              00517000
      DRAT(3,2)=SL*CD+((-SD)*CL*CT)               00518000
      DRAT(1,3)=- (ST)*CD                          00519000
      DRAT(2,3)=CL*SD+CD*(-SL)*CT                 00520000
      DRAT(3,3)=SL*SD+CL*CT*CD                    00521000
      DRAT(4,4)=1.                                00522000
      RETURN 1                                     00523000
      END                                           00524000
C                                                    00525000
C                                                    00526000
C-----C-----00527000
C SUBROUTINE STERO - TO CONSTRUCT STEREO DRAWING MATRIX. 00528000
C-----C-----00529000
      SUBROUTINE STERO(M,NOUT,*)                   00530000
      COMMON/TABLES/NAME,GRUOP                    00531000
      COMMON/PDATA/D,OPT,RX1,RX2,RY1,RY2          00532000
      COMMON/SDATA/IOT,RTV,EYE/MISCD/IDT,IDEV,SF,FRA 00533000
      COMMON/HIDD/NODRAT,PUNT(350,3),IPLA(1200),INSAV,HID,NEXTF,
1NEXTP,NEXTL,ITP,H1,NTRAT                        00534000
      COMMON/COLOUR/ICHPN(31,2),ICOL(30)          00535000
      COMMON/VECTOR/XT(1000),YT(1000),ISG(1000),NSTOR,GOOD 00536000
      LOGICAL RTV,NODRAT,INSAV,HID,H1              00537000
      INTEGER NAME(30,5),GRUOP(30,3),M(30)         00538000
      RX1=-4.6                                     00539000
      RX2=4.6                                      00540000
      9 PRINT 1                                    00541000
      1 FORMAT('ENTER OUTPUT TYPE AND 1, FOR A STEREO MOVIE:') 00542000
8 CONTINUE                                         00543000
700 CALL FREAD(4,&700,&701)                       00544000
701 CALL IPARM(IOT,0,&700)                         00545000
      CALL IPARM(IMO,0,&700)                       00546000
      CALL IPARM(IST,0,&700)                       00547000
      IF(IOT.GT.0)GO TO 7                          00548000
      PRINT 6                                       00549000
      6 FORMAT(' 1=35MM PROPORTIONS 2=SQUARE PROPORTIONS 3=HEAD TO HEAD') 00550000
      GO TO 9                                       00551000
7 IF(IMO.GE.1)GO TO 80                            00552000
      PRINT 4                                       00553000
4 FORMAT('ENTER XYZ OF VIEWER AND VIEWED POINT:') 00554000
710 CALL FREAD(4,&710,&711)                       00555000
711 CALL RPARM(XV,0.0,&710)                       00556000
      CALL RPARM(YV,0.0,&710)                     00557000
      CALL RPARM(ZV,0.0,&710)                     00558000
                                           00559000

```

	CALL RPARM(XB,0.0,&710)	0056000C
	CALL RPARM(YB,0.0,&710)	00561000
	CALL RPARM(ZB,0.0,&710)	00562000
	PRINT 5	00563000
5	FORMAT('ENTER EYE SPACING AND CONE OF VISION:')	00564000
720	CALL FREAD(4,&720,&721)	00565000
721	CALL RPARM(EYE,0.0,&720)	00566000
	CALL RPARM(CONE,60.0,&720)	00567000
	IF(EYE.LE.0)EYE=2.675/12.	00568000
	IF(CONE.LE.0)CONE=60.	00569000
	GO TO(20,21,22),IOT	00570000
20	RY1=-4.6/1.5	00571000
	RY2=4.6/1.5	00572000
	GO TO 30	00573000
21	RY1=-4.6	00574000
	RY2=4.6	00575000
	GOTO 30	00576000
22	RY1=-4.6/1.333	00577000
	RY2=4.6/1.333	00578000
30	CALL EYESP(XV,YV,XB,YB,DX,DY)	00579000
	XL=XV-DX	00580000
	YL=YV+DY	00581000
	XR=XV+DX	00582000
	YR=YV-DY	00583000
	IF(IST.GT.0)DX=0.	00584000
	IF(IST.GT.0)DY=0.	00585000
C		00586000
C_LET	THE PERSPECTIVE ROUTINE CALCULATE THE DRAWING MATRIX.	00587000
C		00588000
32	RTV=.FALSE.	00589000
	CALL PERS1(XL,YL,ZV,XB-DX,YB+DY,ZB,.5,.5,CONE,&50)	00590000
15	RTV=.TRUE.	00591000
	CALL PERS1(XR,YR,ZV,XB+DX,YB-DY,ZB,.5,.5,CONE,&50)	00592000
50	IF(HID)CALL HIDLI(&52)	00593000
	DO 160 I=1,NOUT	00594000
	IF (GROUP(M(I),3).NE.0) GO TO 170	00595000
160	CONTINUE	00596000
170	ICHPN(I,1)=1	00597000
	DO 51 K=1,NOUT	00598000
	IF(GROUP(M(K),3).EQ.0)GO TO 51	00599000
	CALL ELEMAT(M(K))	00600000
	ICHPN(I+1,1)=NSTOR+1	00601000
	ICHPN(I,2)=ICOL(M(I))	00602000
51	CONTINUE	00603000
52	IF(.NOT.RTV)GO TO 15	00604000
	IF(IOT.GE.3)GO TO 55	00605000
	RX2=RX2+9.2	00606000
	GO TO 56	00607000
55	RX2=RY2*3.	00608000
	RX1=-RY2	00609000
	RY1=-4.6	00610000
	RY2=4.6	00611000
56	CALL PUTOUT	00612000
	RY1=-4.6	00613000
	RY2=4.6	00614000
	RX2=4.6	00615000

```
      RX1=-4.6
      RETURN 1
80  IDT=6
      CALL MOVIE(M,NCUT)
      RETURN 1
      END
```

```
00616000
00617000
00618000
00619000
00620000
00621000
```



```

C                                                    00001000
C                                                    00002000
C-----                                                    00003000
C SUBROUTINE BUILD - TO ENTER AN OBJECT BY SKETCHING ON 00004000
C                  AN INTERACTIVE GRAPHICS TERMINAL.    00005000
C-----                                                    00006000
      SUBROUTINE BUILD(*)                                00007000
      COMMON/SCALE/X,Y,SF,VSF/STOR/LB,LE,PX,PY,PZ,NL,NP,PTEMP,PLANE 00008000
      COMMON/TABLES/NAME(30,5) /MISC/JT                00009000
      REAL HT(50,2)                                     00010000
      REAL YES/'YES '/                                  00011000
      DIMENSION LB(500),LE(500),PX(350),PY(350),PZ(350) 00012000
      INTEGER PTEMP(500),FRONT,BACK,P,OBNAME,BLANK/'    '/' 00013000
      LOGICAL PLANE(50),LSW(100),CLOCK                 00014000
C                                                    00015000
C_ERASE SCREEN AND SET UP REFERENCE GRID              00016000
      9 CALL CKDER                                       00017000
      CALL GRID(X,Y)                                     00018000
      CALL CKMA(0.,80.,1)                               00019000
      PRINT 101                                          00020000
      101 FORMAT('ENTER GRID SIZE:')                   00021000
      1 CALL FREAD(4,&1,&2)                             00022000
      2 CALL RPARM(GUD,0.0,&1)                          00023000
      SF=GUD/25.                                         00024000
      IF(GUD.EQ.0.0)SF=.04                             00025000
C                                                    00026000
C_POST SKETCHING MENU                                00027000
      CALL CKMA(550.,750.,1)                           00028000
      PRINT 103                                          00029000
      103 FORMAT('SKETCH          S=STOP')              00030000
      CALL CKMA(550.,730.,1)                           00031000
      PRINT 104                                          00032000
      104 FORMAT('M=MOVE/NEW    V=VECTOR/NEW')          00033000
      CALL CKMA(550.,710.,1)                           00034000
      PRINT 105                                          00035000
      105 FORMAT('N=MOVE/OLD    B=VECTOR/OLD')          00036000
C                                                    00037000
C_INITIALIZE POINTERS                                00038000
      P=1                                                00039000
      L=1                                                00040000
      NL=0                                               00041000
      NP=0                                               00042000
      NPLA=0                                             00043000
      NPT=2                                              00044000
      PTEMP(1)=0                                         00045000
C                                                    00046000
C_ENTER SKETCHING ROUTINE                            00047000
      7 CALL SKETCH(LB,LE,PX,PY,NL,NP,&9)              00048000
      NUML=NL-L+1                                        00049000
      IF(NUML.EQ.0)GO TO 89                             00050000
      CALL HITE(YB,YT)                                   00051000
      DO 8 I=1,100                                       00052000
      8 LSW(I)=.TRUE.                                    00053000
C                                                    00054000
C_START LINE "CHAIN"                                  00055000

```

```

DO 20 J=1,NUML                                00056000
IF(.NOT.(LSW(J)))GO TO 20                      00057000
PLANE(PTEMP(1)+1)=.TRUE.                      00058000
NPLA=NPLA+NPT                                00059000
NPT=3                                         00060000
PTEMP(NPLA+1)=LB(L+J-1)                     00061000
PTEMP(NPLA+2)=LE(L+J-1)                     00062000
BACK=LB(L+J-1)                              00063000
FRONT=LE(L+J-1)                             00064000
LSW(J)=.FALSE.                              00065000
GO TO 5                                       00066000
13 FRONT=LE(L+I-1)                           00067000
GO TO 15                                     00068000
14 FRONT=LB(L+I-1)                           00069000
15 LSW(I)=.FALSE.                           00070000
IF(FRONT.EQ.BACK)GO TO 19                   00071000
PTEMP(NPLA+NPT)=FRONT                       00072000
NPT=NPT+1                                   00073000
C                                             00074000
C_SEARCH THE REMAINDER OF THE LINE LIST      00075000
C_FOR A CONTIGUOUS LINE.                   00076000
5 DO 10 I=J,NUML                             00077000
IF(.NOT.(LSW(I)))GO TO 10                   00078000
IF(FRONT.EQ.LB(L+I-1))GO TO 13              00079000
IF(FRONT.EQ.LE(L+I-1))GO TO 14              00080000
10 CONTINUE                                00081000
PLANE(PTEMP(1)+1)=.FALSE.                  00082000
C                                             00083000
C_FINISHED ONE POLYGON.                    00084000
C_NOW DRAW AND RECORD IT.                  00085000
19 PTEMP(NPLA)=NPT-1                        00086000
NUMP=PTEMP(NPLA)                           00087000
CALL CKMA(PX(PTEMP(NPLA+1))+514.,-X,YB)     00088000
CALL CKVA(PX(PTEMP(NPLA+1))+514.,-X,YT)     00089000
DO 30 K=2,NUMP                             00090000
CALL CKMA(PX(PTEMP(NPLA+K-1))+514.,-X,YT)   00091000
CALL CKVA(PX(PTEMP(NPLA+K))+514.,-X,YT)     00092000
CALL CKVA(PX(PTEMP(NPLA+K))+514.,-X,YB)     00093000
30 CALL CKVA(PX(PTEMP(NPLA+K-1))+514.,-X,YB) 00094000
CALL CKTRAN                                00095000
PTEMP(1)=PTEMP(1)+1                        00096000
HT(PTEMP(1),1)=YB                          00097000
HT(PTEMP(1),2)=YT                          00098000
20 CONTINUE                                00099000
C                                             00100000
DO 6 I=1,3                                 00101000
CALL CKMA(550.,750.,1)                     00102000
6 PRINT 106                                00103000
106 FORMAT('SKETCH')                       00104000
L=1+NL                                     00105000
P=1+NP                                     00106000
GO TO 7                                     00107000
C                                             00108000
C_FINISHED SKETCHING.                      00109000
89 NEXT=NPLA+NPT                           00110000
CALL CKMA(0.,40.,1)                        00111000

```

```

      PRINT 107
107  FORMAT('ENTER VERTICAL GRID SIZE:')
40   CALL FREAD(4,&40,&41)
41   CALL RPARM(VGUD,0.0,&40)
      VSF=VGUD/25.
      IF(VGUD.EQ.0.0)VSF=SF
      N=1
      NPLA=2
C
C_MAKE SURE TOP PLANE IS CLOCKWISE
21  IF(.NOT.(PLANE(N)))GO TO 23
    IF(CLOCK(PTEMP(NPLA)))GO TO 23
    NUM=PTEMP(NPLA)/2
    DO 22 J=1,NUM
      ITEMP=PTEMP(PTEMP(NPLA)+NPLA-J+1)
      PTEMP(PTEMP(NPLA)+NPLA-J+1)=PTEMP(NPLA+J)
      PTEMP(NPLA+J)=ITEMP
22  CONTINUE
23  IF(N.EQ.PTEMP(1))GO TO 90
    N=N+1
    NPLA=NPLA+PTEMP(NPLA)+1
    GO TO 21
C
C_SAVE THE SKETCH AS AN OBJECT?
90  CALL CKMA(0.0,760.0,1)
    PRINT 116
116  FORMAT('WISH TO SAVE SKETCH(YES,NO)?')
50   CALL FREAD(4,&50,&51)
51   CALL APARM(ANS,' ',4,&50)
    IF(ANS.NE.YES)RETURN 1
91   PRINT 118
118  FORMAT('ENTER OBJECT NAME:')
60   CALL FREAD(4,&60,&61)
61   CALL APARM(OBNAME,' ',4,&60)
    IF(OBNAME.EQ.BLANK)GO TO 91
    DO 92 I=1,JT
92   IF(OBNAME.EQ.NAME(I,1))CALL ERRORS(20,0,'OBJ NAME',&91)
      CALL PEXPND(HT,NEXT)
      CALL BINTER(NEXT,OBNAME)
      RETURN 1
    END
C
C
C-----
C LOGICAL FUNCTION CLOCK - TO DETERMINE IF A POLYGON IS DEFINED
C                               BY A SERIES OF CLOCKWISE POINTS.
C-----
      LOGICAL FUNCTION CLOCK(PLVEC)
      COMMON/STOR/LB(500),LE(500),PX,PY,PZ
      INTEGER PLVEC(1)
      REAL PX(350),PY(350),PZ(350)
      A(X1,Y1,X2,Y2)=(X2-X1)*(Y2+Y1+10000.)/2
      N=PLVEC(1)+1
      AREA=0.
      DO 10 I=3,N
10   AREA=AREA+A(PX(PLVEC(I-1)),PY(PLVEC(I-1)),

```

```

1          PX(PLVEC(I)) ,PY(PLVEC(I)))      00168000
AREA=AREA+A(PX(PLVEC(N)) ,PY(PLVEC(N))),    00169000
1          PX(PLVEC(2)) ,PY(PLVEC(2)))      00170000
CLOCK=.TRUE.                                00171000
IF(AREA.LT.0.)CLOCK=.FALSE.                 00172000
RETURN                                       00173000
END                                           00174000
C                                           00175000
C                                           00176000
C                                           00177000
SUBROUTINE FINDPT(X,Y,I,PX,PY,NP,PTOP,*)    00178000
REAL PX(1),PY(1)                           00179000
INTEGER PTOP                                00180000
NP2=NP-1                                    00181000
DO 10 I=PTOP,NP2                            00182000
IF(ABS(X-PX(I)).LE.5 .AND.                 00183000
2  ABS(Y-PY(I)).LE.5) GO TO 20             00184000
10 CONTINUE                                00185000
CALL CKMA(550.0,80.0,1)                    00186000
PRINT 100                                  00187000
100 FORMAT('POINT NOT FOUND')              00188000
RETURN 1                                    00189000
20 X=PX(I)                                  00190000
Y=PY(I)                                    00191000
RETURN                                       00192000
END                                           00193000
C                                           00194000
C                                           00195000
C----- 00196000
C SUBROUTINE GRID - PROJECTS THE SKETCHING ENVIRONMENT. 00197000
C----- 00198000
SUBROUTINE GRID(X,Y,*)                     00199000
CALL CKMA( 25.0,110.0)                     00200000
CALL CKVA( 25.0,690.0)                     00201000
CALL CKVA(486.0,690.0)                     00202000
CALL CKVA(486.0,110.0)                     00203000
CALL CKVA( 25.0,110.0)                     00204000
CALL CKMA(514.0,110.0)                     00205000
CALL CKVA(514.0,690.0)                     00206000
CALL CKVA(975.0,690.0)                     00207000
CALL CKVA(975.0,110.0)                     00208000
CALL CKVA(514.0,110.0)                     00209000
DO 10 I=1,39                               00210000
CALL CKMA((25.0*FLOAT(I)),105.0)           00211000
10 CALL CKVA((25.0*FLOAT(I)),111.0)        00212000
DO 20 I=1,32                               00213000
CALL CKMA(497.0,(25.0*FLOAT(I-1))+8.0)    00214000
20 CALL CKVA(503.0,(25.0*FLOAT(I-1))+8.0)  00215000
CALL CKTRAN                                00216000
X=25.0                                     00217000
Y=108.0                                    00218000
CALL CKMA(30.0,120.0,1)                    00219000
PRINT 100                                  00220000
100 FORMAT('PLAN:')                        00221000
CALL CKMA(520.0,120.0,1)                  00222000
PRINT 101                                  00223000

```

101 FORMAT('ELEV')	00224000
RETURN	00225000
END	00226000
C	00227000
C	00228000
C-----	00229000
C SUBROUTINE HITE - TO ENTER AND RECORD THE THIRD COORDINATES.	00230000
C-----	00231000
SUBROUTINE HITE(YB,YT)	00232000
COMMON/SCALE/X,Y,SF,VSF	00233000
LOGICAL PLANE,INOUT	00234000
C	00235000
C_BASE LOCATION	00236000
9 CALL CKMA(519.0,40.0,1)	00237000
PRINT 101	00238000
101 FORMAT('ENTER BASE LOCATION')	00239000
CALL CKGCUR(I,DUMB,YB)	00240000
IF(I.EQ.5 .OR. I.EQ.27 .OR. I.EQ.83)GO TO 9	00241000
C	00242000
C_TOP LOCATION	00243000
CALL CKMA(519.0,20.0,1)	00244000
PRINT 102	00245000
102 FORMAT('ENTER TOP LOCATION')	00246000
CALL CKGCUR(I,DUMB,YT)	00247000
IF(I.EQ.5 .OR. I.EQ.27 .OR. I.EQ.83)GO TO 9	00248000
IF(YB.GT.YI)GO TO 60	00249000
RETURN	00250000
60 YTEMP=YB	00251000
YB=YT	00252000
YT=YTEMP	00253000
RETURN	00254000
END	00255000
C	00256000
C	00257000
C	00258000
SUBROUTINE LNADD(X1,Y1,I1,X2,Y2,I2,LB,LE,NL,PX,PY,NP)	00259000
REAL PX(1),PY(1)	00260000
INTEGER LB(1),LE(1)	00261000
PX(I1)=X1	00262000
PY(I1)=Y1	00263000
LB(NL)=I1	00264000
PX(I2)=X2	00265000
PY(I2)=Y2	00266000
LE(NL)=I2	00267000
NL=NL+1	00268000
CALL CKMA(X1,Y1,1)	00269000
CALL CKVA(X2,Y2,1)	00270000
RETURN	00271000
END	00272000
C	00273000
C	00274000
C-----	00275000
C SUBROUTINE PEXPND - TO EXPAND THE SKETCHED POLYGON INTO	00276000
C A LEGAL ARCH:GRAPHIC OBJECT.	00277000
C-----	00278000
SUBROUTINE PEXPND(HT,NEXT,*)	00279000

COMMON/SCALE/X,Y,SF,VSF/STOR/LB,LE,PX,PY,PZ,NL,NP,PTEMP,PLANE	00280000
REAL PX(350),PY(350),PZ(350),HT(50,2)	00281000
INTEGER TP(100),LB(500),LE(500),PTEMP(500)	00282000
LOGICAL PLANE(50)	00283000
NUMPL=PTEMP(1)	00284000
N=1	00285000
NPLA=2	00286000
9 NUM=PTEMP(NPLA)	00287000
NP=NP+1	00288000
NL=NL+1	00289000
ITP=0	00290000
J=NPLA+1	00291000
K=NPLA+NUM	00292000
DO 10 I=J,K	00293000
C	00294000
C_ADD THIRD DIMENSION TO OLD POINT	00295000
PX(PTEMP(I))=(PX(PTEMP(I))-X)*SF	00296000
PY(PTEMP(I))=(PY(PTEMP(I))-Y)*SF	00297000
PZ(PTEMP(I))=(HT(N,2)-Y)*VSF	00298000
C	00299000
C_ADD NEW POINT BELOW OLD POINT	00300000
PX(NP)=PX(PTEMP(I))	00301000
PY(NP)=PY(PTEMP(I))	00302000
PZ(NP)=(HT(N,1)-Y)*VSF	00303000
IF(.NOT.PLANE(N))GO TO 50	00304000
C	00305000
C_ADD NEW POINT INDEX TO TEMPORARY PLANE VECTOR	00306000
ITP=ITP+1	00307000
TP(ITP)=NP	00308000
C	00309000
C_ADD VERTICAL LINE BETWEEN OLD AND NEW POINT	00310000
50 LB(NL)=PTEMP(I)	00311000
LE(NL)=NP	00312000
C	00313000
C_UPDATE LINE AND POINT POINTERS	00314000
NP=NP+1	00315000
NL=NL+1	00316000
IF(I.EQ.J)GO TO 10	00317000
IF(.NOT.PLANE(N))GO TO 51	00318000
C	00319000
C_ADD SIDE PLANE TO PLANE VECTOR	00320000
PTEMP(NEXT)=4	00321000
PTEMP(NEXT+1)=NP-2	00322000
PTEMP(NEXT+2)=NP-1	00323000
PTEMP(NEXT+3)=PTEMP(I)	00324000
PTEMP(NEXT+4)=PTEMP(I-1)	00325000
C	00326000
C_UPDATE PLANE COUNTER AND POINTER	00327000
PTEMP(1)=PTEMP(1)+1	00328000
NEXT=NEXT+5	00329000
C	00330000
C_ADD BOTTOM LINE	00331000
51 LB(NL)=NP-2	00332000
LE(NL)=NP-1	00333000
NL=NL+1	00334000
10 CONTINUE	00335000

	IF(.NOT.PLANE(N))GO TO 53	00336000
C		00337000
C_ADD	LAST BOTTOM LINE	00338000
	LB(NL)=NP-1	00339000
	LE(NL)=TP(1)	00340000
C		00341000
C_ADD	LAST PLANE	00342000
	PTEMP(NEXT)=4	00343000
	PTEMP(NEXT+1)=NP-1	00344000
	PTEMP(NEXT+2)=TP(1)	00345000
	PTEMP(NEXT+3)=PTEMP(J)	00346000
	PTEMP(NEXT+4)=PTEMP(I)	00347000
	PTEMP(1)=PTEMP(1)+1	00348000
	NEXT=NEXT+5	00349000
C		00350000
C_ADD	BOTTOM PLANE IN REVERSE ORDER	00351000
	PTEMP(NEXT)=ITP	00352000
	DO 20 I=1,ITP	00353000
20	PTEMP(NEXT+I)=TP(ITP-I+1)	00354000
	NEXT=NEXT+ITP+1	00355000
	PTEMP(1)=PTEMP(1)+1	00356000
C		00357000
C_LAST	OLD PLANE?	00358000
52	NP=NP-1	00359000
	IF(N.EQ.NUMPL)RETURN 1	00360000
	N=N+1	00361000
	NPLA=NPLA+NUM+1	00362000
	GO TO 9	00363000
53	NL=NL-1	00364000
	GO TO 52	00365000
	END	00366000
C		00367000
C		00368000
C		00369000
	SUBROUTINE PLSAVE(LX,LY,X,Y,LKEY,I,LP,L,*)	00370000
	REAL LX,LY	00371000
	LX=X	00372000
	LY=Y	00373000
	LKEY=I	00374000
	LP=L	00375000
	RETURN 1	00376000
	END	00377000
C		00378000
C		00379000
C		00380000
	SUBROUTINE SCREEN(I,*)	00381000
	INTEGER KEY(21)/3,10*0,1,2,4*0,5,2*0,4/	00382000
	IF(I.GT.86)I=I-32	00383000
	I=I-65	00384000
	IF(I.LT.0 .OR. I.GT.21)GO TO 10	00385000
	I=KEY(I)	00386000
	IF(I.NE.0)RETURN	00387000
10	CALL CKMA(550.0,60.0,1)	00388000
	PRINT 100	00389000
100	FORMAT('ILLEGAL KEY')	00390000
	RETURN 1	00391000

END	00392000
C	00393000
C	00394000
C-----	00395000
C SUBROUTINE SKETCH - POLYGON SKETCHING ROUTINE.	00396000
C-----	00397000
SUBROUTINE SKETCH(LB,LE,PX,PY,NL,NP,*)	00398000
REAL PX(1),PY(1),LX,LY	00399000
INTEGER LB(1),LE(1),PTOP	00400000
C	00401000
NL=NL+1	00402000
NP=NP+1	00403000
PTOP=NP	00404000
LKEY=0	00405000
C	00406000
9 CALL CKGCUR(I,X,Y)	00407000
IF(I.EQ.5 .OR. I.EQ.27)RETURN 1	00408000
CALL SCREEN(I,&9)	00409000
GO TO (11,12,13,14,15), I	00410000
C	00411000
C_MOVE TO A NEW POINT.	00412000
11 IF(LKEY.NE.1 .AND. LKEY.NE.2)NP=NP+1	00413000
CALL PLSAVE(LX,LY,X,Y,LKEY,I,LP,NP-1,&9)	00414000
C	00415000
C_MOVE TO AN OLD POINT.	00416000
12 CALL FINDPT(X,Y,L,PX,PY,NP,PTOP,&9)	00417000
CALL PLSAVE(LX,LY,X,Y,LKEY,I,LP,L,&9)	00418000
C	00419000
C_VECTOR TO AN OLD POINT.	00420000
13 CALL FINDPT(X,Y,INPT,PX,PY,NP,PTOP,&9)	00421000
CALL LNADD(LX,LY,LP,X,Y,INPT,LB,LE,NL,PX,PY,NP)	00422000
CALL PLSAVE(LX,LY,X,Y,LKEY,I,LP,INPT,&9)	00423000
C	00424000
C_VECTOR TO A NEW POINT.	00425000
14 IF(LKEY.EQ.0)GO TO 11	00426000
CALL LNADD(LX,LY,LP,X,Y,NP,LB,LE,NL,PX,PY,NP)	00427000
NP=NP+1	00428000
CALL PLSAVE(LX,LY,X,Y,LKEY,I,LP,NP-1,&9)	00429000
C	00430000
C_STOP SKETCHING.	00431000
15 NP=NP-1	00432000
NL=NL-1	00433000
RETURN	00434000
END	00435000

<u>//PLOTJOB JOB (___ssn___,___acct#___),name,TIME=(,5)</u>	<u>00001000</u>
<u>/*PASSWORD word</u>	
//INCASE EXEC DUMMY	00003000
//DD1 DD DSN=DS450.ARCHGRAF.LIB,DISP=(MOD,DELETE)	00004000
//CALCOMP EXEC <u>PLOTQCL</u>	<u>00005000</u>
DIMENSION X(4),Y(4),IPEN(4)	00006000
CALL PLOTS	00007000
CALL PLOT(0.0,0.0,23)	00008000
CALL PLOT(5.0,1.0,23)	00009000
J=0	00010000
10 READ(5,20,END=100) (X(I),Y(I),IPEN(I),I=1,4)	00011000
20 FORMAT(4(2F8.0,I4))	00012000
DO 60 I=1,4	<u>00013000</u>
IF (IPEN(I) .GT. 0) GO TO 50	00014000
CALL PLOT(X(I)+5.0,0.0,IPEN(I))	00015000
J=J+1	00016000
WRITE(6,30) J,X(I),Y(I)	00017000
30 FORMAT(' PLOT',I4,' COMPLETE: XMAX =',F6.3,', YMAX =',F6.3)	00018000
GO TO 10	00019000
50 IF (IPEN(I).EQ.2.OR.IPEN(I).EQ.3) GO TO 60	00020000
<u>IPNUM=IPEN(I)/100</u>	<u>00021000</u>
<u>IPEN(I)=MOD(IPEN(I),100)</u>	<u>00022000</u>
CALL NEWPEN(IPNUM)	<u>00023000</u>
60 CALL PLOT(X(I),Y(I),IPEN(I))	00024000
GO TO 10	00025000
100 CALL PLOT(10.0,0.0,999)	00026000
WRITE(6,110)	00027000
110 FORMAT(' *NORMAL END OF PROCESSING*')	00028000
RETURN	00029000
END	00030000
//LKED.SYSLMOD DD DSN=DS450.ARCHGRAF.LIB(HRDCPY),DISP=(NEW,CATLG),	00031000
// UNIT=SYSDA,SPACE=(TRK,(3,1,3),RLSE),VOL=SER=KSCC03	00032000
<u>NAME HRDCPY(R)</u>	<u>00033000</u>

The following is an extract from file GRAPH OBJECT showing the format and data for defining a primitive object.

The object definition is in four parts. The first part, formatted (A4,3I4), contains the object name, number of points, number of lines, and number of planes.

The second part, formatted (3F10.3), consists of the X, Y, and Z coordinates of the vector end points. The coordinates are relative coordinates between "-1" and "1" with "0" as the origin.

The third part, formatted (20I4), defines the lines or vectors by their end points (the points are numbered corresponding to their order of entry).

The fourth part, formatted (20I4), defines the planes. The first number is the number of end points for the plane followed by the numbers of the end points defining the perimeter of the plane.

```

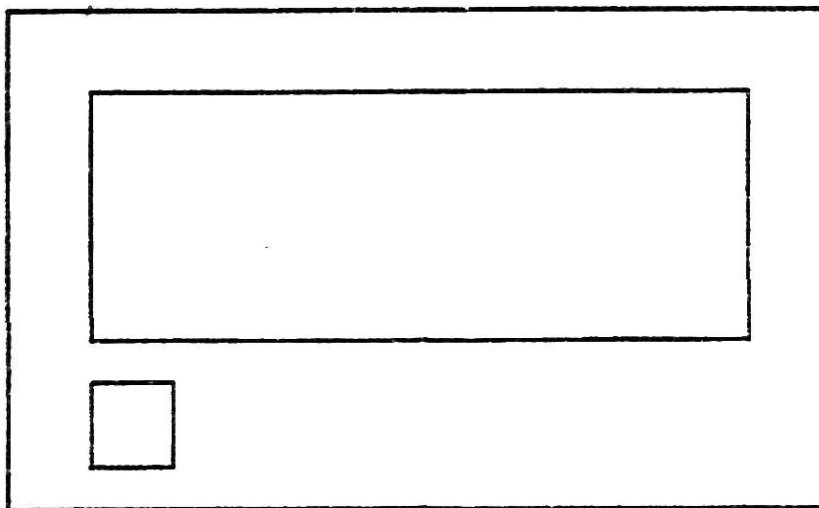
PRSM      6      9      5
          0.0      0.0      0.0
          0.0      0.5      1.0
          0.0      1.0      0.0
          1.0      0.0      0.0
          1.0      0.5      1.0
          1.0      1.0      0.0
1         4         1         3         1         2         2         3         2         5         3         6         6         5         4         6         4         5
3         3         2         1
3         4         5         6
4         1         2         5         4
4         5         2         3         6
4         4         6         3         1

```

1-ELEVATION
2-ROTATION BY AXIS & DEGREES
TYPE 1 OR 2

.1
TYPE NUMBER FOR DESIRED ELEVATION
1-PLAN
2-NORTH
3-EAST
4-SOUTH
5-VEST

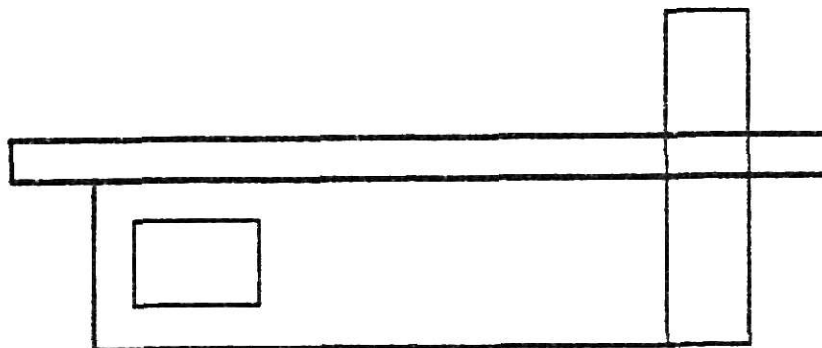
.1
ENTER SCALE FACTOR:
ENTER DRAVING SIZE:
.4



1-ELEVATION
2-ROTATION BY AXIS & DEGREES
TYPE 1 OR 2

.1
TYPE NUMBER FOR DESIRED ELEVATION
1-PLAN
2-NORTH
3-EAST
4-SOUTH
5-WEST

.2
ENTER SCALE FACTOR.
ENTER DRAWING SIZE.
.4



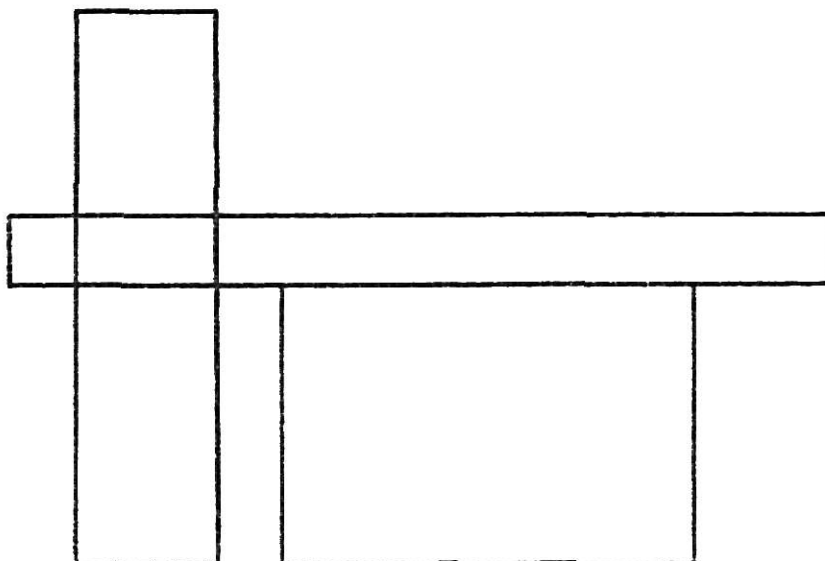
1-ELEVATION
2-ROTATION BY AXIS & DEGREES
TYPE 1 OR 2

.1
TYPE NUMBER FOR DESIRED ELEVATION
1-PLAN
2-NORTH
3-EAST
4-SOUTH
5-WEST

.3
ENTER SCALE FACTOR:

.ENTER DRAWING SIZE:

.4

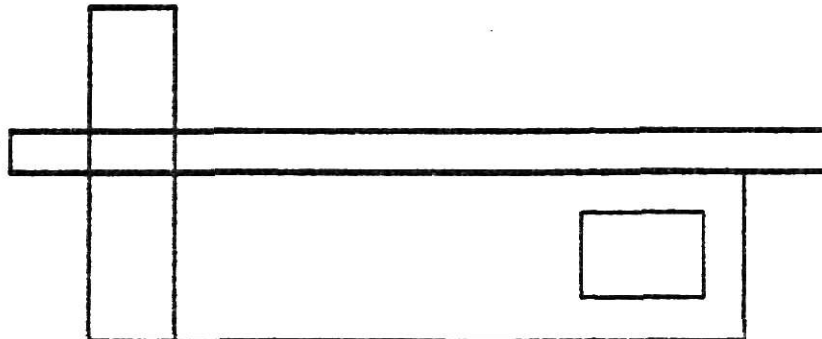


1-ELEVATION
2-ROTATION BY AXIS & DEGREES
TYPE 1 OR 2

.1
TYPE NUMBER FOR DESIRED ELEVATION
1-PLAN
2-NORTH
3-EAST
4-SOUTH
5-WEST

.4
ENTER SCALE FACTOR:

.4
ENTER DRAWING SIZE:

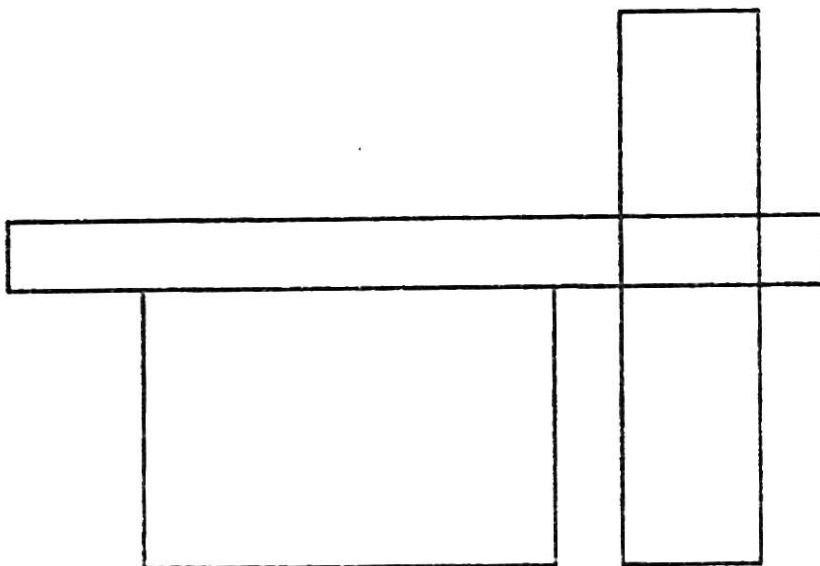


1-ELEVATION
2-ROTATION BY AXIS & DEGREES
TYPE 1 OR 2

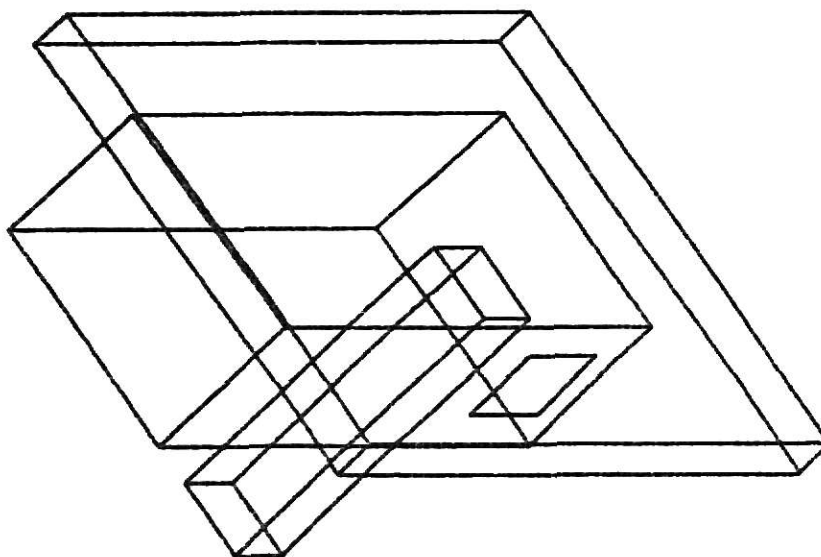
.1
TYPE NUMBER FOR DESIRED ELEVATION
1-PLAN
2-NORTH
3-EAST
4-SOUTH
5-WEST

.5
ENTER SCALE FACTOR:

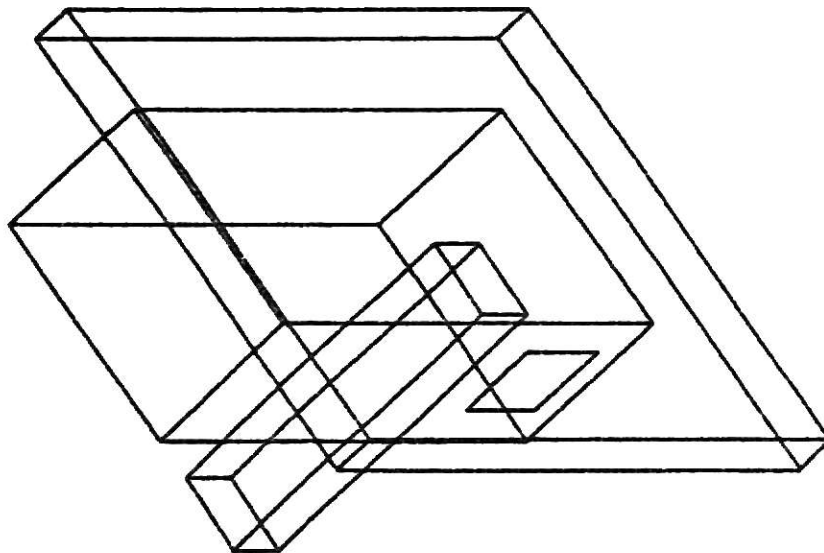
.4
ENTER DRAWING SIZE:



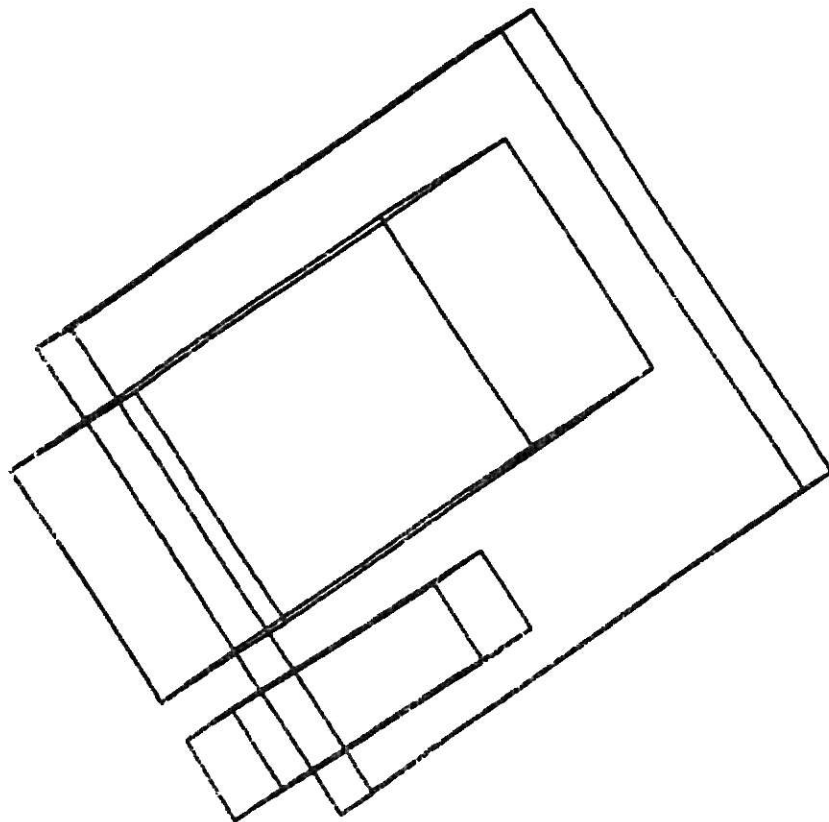
ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.1
ENTER LATITUDE, DECLINATION, HOUR,
(EX- 39.8 23.45 17)
.39 8 16
ENTER SCALE FACTOR:
ENTER DRAWING SIZE:
.4



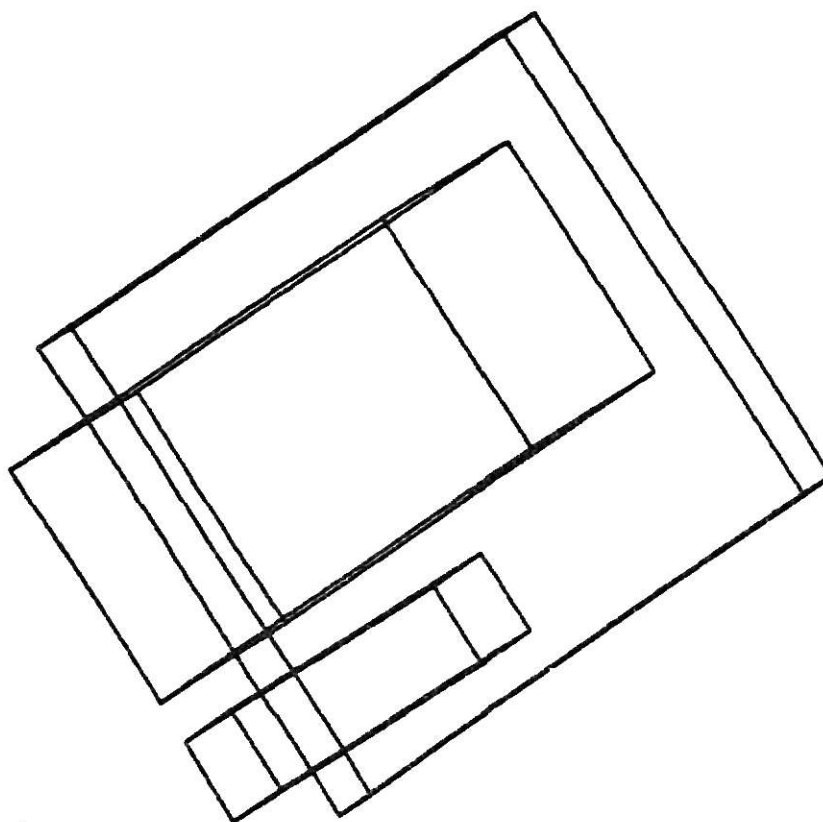
ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.2
ENTER LATITUDE, MONTH, DAY, HOUR:
(EX- 39.8 5 18 17)
.39 3 21 16
DECLINATION IS 8.8
ENTER SCALE FACTOR:
.
ENTER DRAWING SIZE:
.4



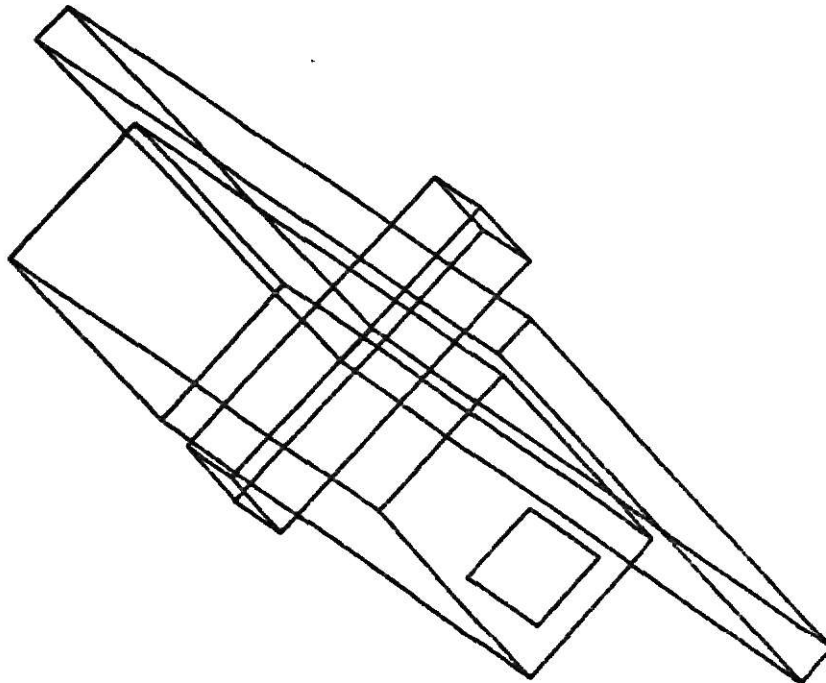
ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.1
ENTER LATITUDE, DECLINATION, HOUR,
(EX- 30.0 23.45 17)
.30 23.45 16
ENTER SCALE FACTOR:
.
ENTER DRAWING SIZE:
.4



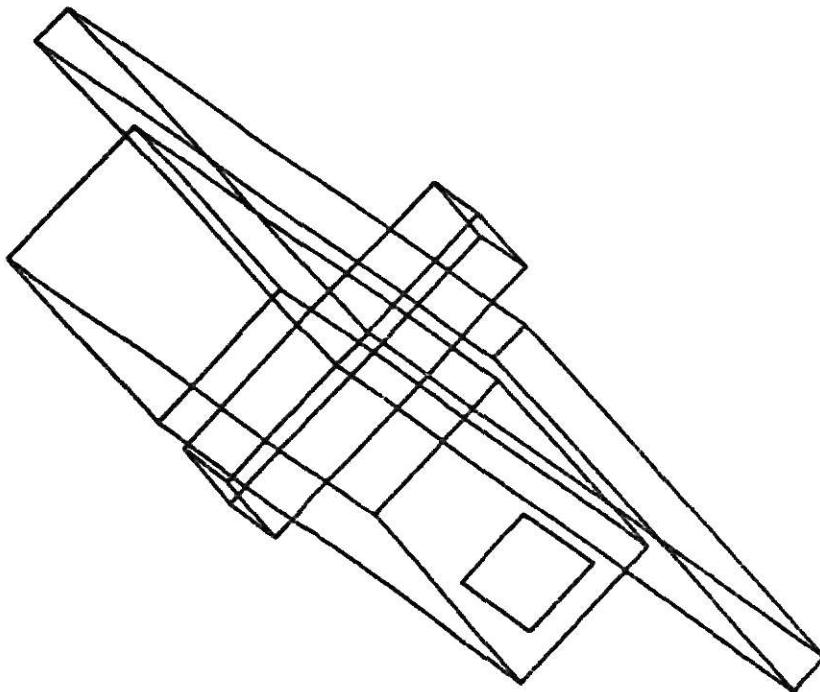
ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.2
ENTER LATITUDE, MONTH, DAY, HOUR:
(EX- 30.0 5 10 17)
.30 6 21 16
DECLINATION IS 23.43
ENTER SCALE FACTOR:
.
ENTER DRAWING SIZE:
.4

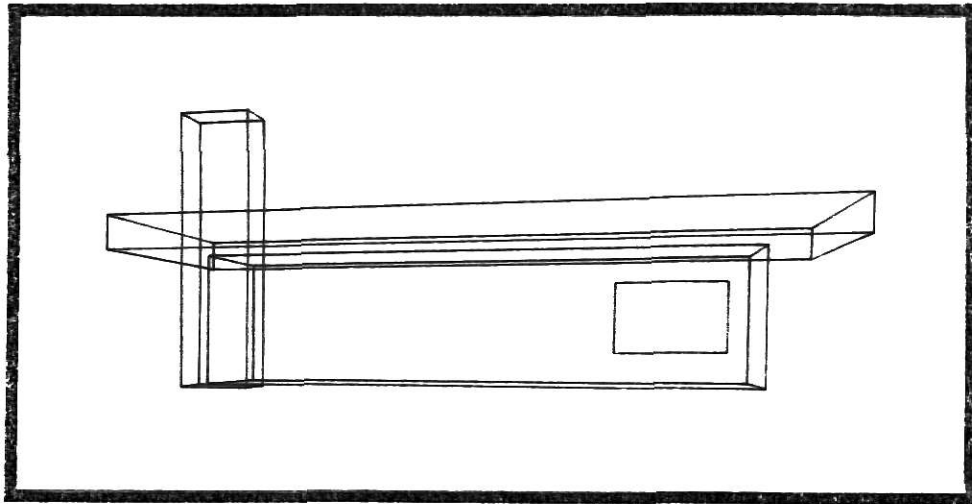


ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.1
ENTER LATITUDE, DECLINATION, HOUR:
(EX- 39.8 23.45 17)
.39 -23.45 16
ENTER SCALE FACTOR:
ENTER DRAWING SIZE:
.4

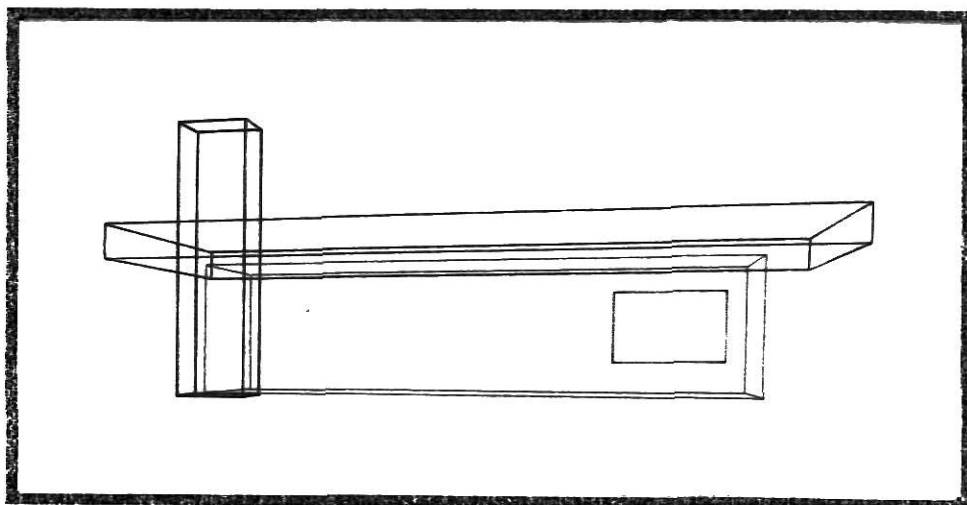


ENTER 1 OR 2
1 - BY DECLINATION
2 - BY DATE
.2
ENTER LATITUDE, MONTH, DAY, HOUR:
(EX- 39.8 5 18 17)
.39 12 21 18
DECLINATION IS -23.43
ENTER SCALE FACTOR:
.
ENTER DRAWING SIZE:
.4





No Color - Default



Color Option

ARCHGRAF.2 : A REVISION OF ARCHGRAF
AN ARCHITECTURAL GRAPHICS PROGRAM

by

GARY WAYNE LAW

B. of Arch., Texas Tech University, 1973

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1981

ABSTRACT

This report provides documentation for the reorganization and revisions to ARCHGRAF, an architectural graphics program.

The reorganization is intended to make future revisions, corrections, and modifications easier to implement and less costly in both human and computer time. Revisions include correcting various miscellaneous defects, compiling under the newer Fortran H Extended Compiler, making changes necessary to run under new system support routines, updating and consolidating plotter routines, adding interface routines for the Chromatics terminal, adding color capabilities for Chromatics and plotter output, modifying the solar and rotation drawing options, and making some options easier for the first-time or infrequent users.

Source code for the ARCHGRAF files is provided as well as source code for the supporting exec and hardcopy routines. Documentation of the revisions and reorganization and access to the source code will hopefully encourage other changes, speed the familiarization process and help facilitate future modification and maintenance.