A CAD APPROACH TO OPTIMIZE UNDERGROUND MINE DESIGN AND PLANNING

A Thesis Presented to

The Faculty of the College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Chaluvadi K. Sridhar

June 1986
ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to Professor P. K. Chatterjee for his valuable guidance and constructive criticisms.

The author is grateful to Professor D. E. Scheck for his suggestions and encouragement. The author also wishes to thank Dr. Jay Gunasekara for having consented to sit in the defense committee.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>List of figures</td>
<td>iii</td>
</tr>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Approach</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Significance</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>LITERATURE REVIEW</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>DESCRIPTION OF THE PACKAGE</td>
<td>12</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Design Criteria</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Implementation of the Design Criteria</td>
<td>14</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Firmware routines of the TEKTRONIX 4107 terminal</td>
<td>15</td>
</tr>
<tr>
<td>3.3.2</td>
<td>ANSI Screen editing</td>
<td>16</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Developing the Algorithm for the Room and Pillar design</td>
<td>19</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Developing the Algorithm for Longwall and development design</td>
<td>20</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Developing the algorithm for retrieving the quality information</td>
<td>22</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Developing the algorithm for scheduling and simulating designed room and pillar, longwall and development sections</td>
<td>22</td>
</tr>
<tr>
<td>3.3.7</td>
<td>Developing the interface for plotting the graphics</td>
<td>25</td>
</tr>
<tr>
<td>3.3.8</td>
<td>Developing the module for directory information</td>
<td>26</td>
</tr>
<tr>
<td>3.3.9</td>
<td>Developing the module for displaying the room and pillar parameter window</td>
<td>26</td>
</tr>
<tr>
<td>3.3.10</td>
<td>Developing the module for overlays of grid and isopachs</td>
<td>27</td>
</tr>
<tr>
<td>3.3.11</td>
<td>Developing the module for guarding the program against wrong inputs</td>
<td>30</td>
</tr>
</tbody>
</table>

4 APPLICATION

5 DISCUSSION

6 CONCLUSIONS AND RECOMMENDATIONS

BIBLIOGRAPHY

APPENDIX A

APPENDIX B
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Underground Mine Planning System</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Three Stages of Planning</td>
<td>5</td>
</tr>
<tr>
<td>3a</td>
<td>Sample of a Option Screen Menu</td>
<td>17</td>
</tr>
<tr>
<td>3b</td>
<td>Sample of a Design Parameter Input Screen</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>Room and Pillar Design Parameter Window</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>User specified grid</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Layout Plot</td>
<td>33</td>
</tr>
<tr>
<td>7</td>
<td>Room and Pillar Design with Constant Dimensions</td>
<td>35</td>
</tr>
<tr>
<td>8</td>
<td>Room and Pillar Design with Varying Dimensions</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>Room and Pillar Design with Changed Pillars</td>
<td>37</td>
</tr>
<tr>
<td>10</td>
<td>Room and Pillar Design in an irregular area</td>
<td>38</td>
</tr>
<tr>
<td>11</td>
<td>Longwall Design I</td>
<td>39</td>
</tr>
<tr>
<td>12</td>
<td>Longwall Design II</td>
<td>41</td>
</tr>
<tr>
<td>13</td>
<td>Longwall Design III</td>
<td>42</td>
</tr>
<tr>
<td>14</td>
<td>Longwall Design in an irregular area</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>Complete Underground Mine Plan</td>
<td>45</td>
</tr>
<tr>
<td>16</td>
<td>Total Simulation of Four Sections</td>
<td>46</td>
</tr>
</tbody>
</table>

iii
Chapter I
INTRODUCTION

1.1 Statement of the problem

The planning of an underground coal mine requires a complex blend of technical expertise, experience and subjective design criteria implemented by the mining engineer. Currently the procedure is largely manual and requires considerable resources to generate a single mine plan. Often three or four people are assigned to work in the development of a single draft layout. Evaluation of the layouts is also tedious as it requires many repetitive calculations. The evaluation of layouts developed manually generally attempt to derive a single economically feasible solution. Operational constraints in underground coal mining such as manpower, time, equipment, etc, do not allow the generation of more than one alternative layout. As a result, the first plan is generally the only one considered.

Although recent advances in computer graphics have provided the hardware necessary for developing and evaluating mine plans rapidly, software that will allow the mining engineer to make use of the new technology is slow in developing. The need for such software packages
has been recognized and acknowledged. At several sessions in recent conferences on the applications of computers in the mining industry, the uses and benefits of computer-aided mine planning were discussed. However there were few papers presented that referred to actual applications of interactive graphics to underground mine planning.

The emphasis of this study is towards the development of a completely integrated computer-aided underground mine planning package that integrates geological modelling, engineering design and planning strategies with interactive on-screen graphics capabilities.

1.2 Objectives

The overall objective of the research effort was to develop an integrated computer-aided underground coal mine design procedure that will help maximize coal recovery at greatly reduced costs. This procedure would aid the mining engineer to develop a total mine plan from the conceptual design stage, the analysis of bore hole data, and the drafting of a layout to the final working mine plan, mine schedule, simulation and cost evaluation.
Figure 1 illustrates the integration of the various planning activities within the proposed package.

There are three distinct stages in the mine planning procedure as indicated in Figure 2. Programs for the first and the second stage, i.e., geostatistical analyses and reserve estimation and designing a mine layout, are already available. Therefore, the aim of this effort was directed to the second and third stages, i.e. room and pillar section, designing longwall sections and simulation of the designed sections with the geostatistical database.

The main feature of this suite of programs is the interactive design and drafting procedure which allows continuous updating and refinement of the mine plan as it is being developed. The user is able to design a layout and to perform a deterministic simulation to determine the production levels of different extraction strategies.

The design layout is incorporated into a longwall and room and pillar simulation package that makes use of bore hole data and other coal sample and drilling information to evaluate the layout. The economic evaluations of the different layouts given by the simulation are used to select the optimum layout.
Fig. 1 Underground mine planning system
Fig. 2 Three stages of planning
1.3 Approach

The mine planning programs were developed on a Micro-Vax I and a Tektronix 4107 color graphics terminal. The programs were developed in three phases.

Phase 1:

**Development of the Tektronix firmware routines**

The Tektronix color graphics terminal has the capability of Computer Aided Design (CAD) workstation. The terminal has a large number of firmware routines in its ROM, which are capable of creating windows, overlays, segments, storing images for manipulations, etc... All these routines can be accessed through FORTRAN program calls through Escape sequences from the host computer. Since, the mine planning package uses a larger number of these features, the effort in the first phase was concentrated to develop these firmware routines.

Phase 2:

**Development of Mine planning Programs**

In this phase the programs for designing room and pillar sections, Longwall sections, simulation and
scheduling of designed sections, creation of windows for displaying design parameters, interfacing the mine plans with the geological database, creating user grids and overlays were developed. All the programs were developed with extensive graphic capabilities.

Phase 3:

Development of HELP program

VAX help facilities were used to create extensive help menus for the user. The HELP menus were designed in two different levels, the top level containing the main options of the mine planning software and the second level containing information about the various design parameters.

1.4 Significance

Computer-aided mine planning can automate many of the manual procedures and calculations involved in mine planning thus allowing many layouts and mining strategies to be evaluated and the optimum plan selected with confidence. This optimizing planning procedure also allows the evaluation of mine performance continuously and critically, thereby helping to maximize return on investment.
CHAPTER 2

Literature Review

Wright (1985) emphasized the need for new technology to implement a totally integrated mine planning procedure in operating mines. The author discussed the features of automated mapping and computer-aided design as applied to mine mapping and stated that the constraint in using these high level graphic systems is the high cost associated with the work stations necessary to implement them.

One of the earliest efforts in the area of computer-aided mine design was reported by Hempenstal, Jefferys and Davies (1982). The study was the development of an underground colliery planning package. The colliery planning procedures were developed as an addition to the computerized seam deposits and user oriented mine planning system (SUMPS). The package did not incorporate any geological analysis of the bore hole information in its layout design and evaluation.

Deaking, Puckett and Matti, 1983 reported the development of MINDRAF T which allowed proposed mine layouts and extensions to be graphically displayed. Benefits achieved through the use of this package included economy and speed (in comparison with manual
methods); sections of a mine plan can be produced quickly to any scale thus saving the resources needed to create them.

A coal mine simulator package that uses an effective video display was reported by Barczak (1983). This package dynamically displayed the progression of entries into a room and pillar section and the movement and location of various extraction equipment.

More recently, FRAPS (Friendly Room and Pillar Simulation) was developed to facilitate planning and evaluate and improve productivity and the movement and location of various extraction equipment (Haycocks et al., 1984).

Kim, Knudsen and Baafi (1980) reported a short term and long term coal mine planning package. The report emphasized graphic plots to designate room and pillar locations and current and future sampling points.


Adel, Haycocks and Lucas (1983) reviewed over sixteen mine planning programs developed at Virginia Polytechnic Institute and State University. The main
areas of applications were: exploration and evaluation, costing and scheduling, face simulation (FACESIM), transportation and support engineering services.

Topuz and Nasaf (1985), discussed the development of CONSIM - a mine planning package. CONSIM was written to be run on microcomputers and is essentially a modification of FACESIM. The CONSIM package does not use any interactive graphic displays to aide the user.

Chatterjee et. al., (1985) reported a highly interactive mine planning and drafting package designed to run on the Macintosh. Owing to memory limitations of the Macintosh a geological data base was unable to be included. Important features of the program were the generation of detailed simulation reports of production by target year, tonnage and longwall panels.

Chatterjee, Scheck, Sridhar (1986), reported a underground mine planning package incorporating the geological database, scheduling, overlays of isopachs and user grid. This system was designed to work on a Micro-Vax and a Tektronix 4107 workstation.

The literature review indicates that interactive graphics in underground mine planning is an area in which researchers are just beginning to invest time and resources. Although some recent reports and publications
have cited the use of interactive graphics, most of these are rudimentary applications of graphics software. Also, the programs that are currently available have been developed in a piece meal fashion and generally there appears to have been no effort to develop an integrated mine design and planning package with a geological data base system.

The computer-aided design procedure described in this thesis is a clear attempt to overcome the current shortcomings of available mine design software. It incorporates many of the features outlined in the simulation and drafting packages discussed in the above review. Additionally, it has the ability to relate to a geological data base and the entire program is menu driven with interactive, refresh graphics and map quality plots as output.
CHAPTER 3

DESCRIPTION OF THE PACKAGE

3.1 Introduction

As discussed earlier, the aim of this study is to provide the mining engineer with a mine design software package which is user-friendly and which will help him to design and plan underground mines with ease and confidence. The programs would require to be developed in such a way that they can be easily interfaced with future programs in the areas of ventilation, belt simulation and other auxiliary mine services.

3.2 Design Criteria

The design criteria for the proposed package are as follows:

i) The software must be user-friendly and highly interactive so that the user will have control over the execution of the package with provisions to use various input and output devices.
ii) A typical user of this package will be familiar with the underground mine planning, scheduling of mining operations, geological information about the coal seam and the information concerning the mining rates, mine working calendar and the exception information for the sections.

iii) The programs should be designed in a modular fashion, so that people with different expertise and background can work on selective modules which are relevant to their field of expertise.

iv) The software must be robust so that any absurd input is ignored and the user is instructed to correct the input.

v) Adequate on-line Help information to use the package, should be available to the user.

vi) Prompts for the inputs should be as short as possible. At the same time these should be clear enough for the user to understand and respond with the information that is expected.

vii) Provisions should be provided in the
software to keep a log of the different users and the files they have used or created.

viii) The programs should be device independent. If total independence is not possible there should be provisions to adapt the programs for a particular device.

ix) The terms or verbage used should be compatible to mining practice usage and visual aids in the form of windows should be provided for the better understanding of the various design parameters.

3.3 Implementation of design criteria

The different stages in the development of the software for modeling the underground mine planning software are:

i. Developing the firmware routines to take full advantage of the Tektronix 4107 terminal.

ii. Developing the screen editing routines to handle all input/outputs to and from the program.

iii. Developing algorithm for designing the room and pillar sections.

iv. Developing algorithm for designing the Longwall
and development sections.

v. Developing the interface for retrieving quality information from a geological grid.

vi. Developing the algorithm for scheduling and simulating designed room & pillar and longwall sections.

vii. Developing the interface for plotting the screen graphics.

viii. Developing a directory program for keeping a log of the users.

ix. Developing window for displaying the room and pillar parameters.

x. Developing the overlays of grid and isopachs.

xi. Developing modules for protection against wrong inputs from the user.

3.3.1 Firmware routines of the TEKTRONIX 4107 terminal

The tektronix 4107 is a smart terminal with a 256K built in RAM and a number of firmware routines which reside on the ROM. All these firmware routines can be accessed through two modes.

i) SETUP mode

ii) FORTRAN calls
In the SETUP mode, the user puts the terminal into SETUP mode and issues the command through the keyboard (e.g. MOVE 200 200 will move the beam to the co-ordinate 200,200).

Through FORTRAN calls, the commands to the terminal are sent using escape sequences. The second mode, namely the calls through FORTRAN have been widely used for developing the underground mine planning software. Commands for dividing the screen, accessing the fill patterns, colors, line styles, key macros, etc. are done through FORTRAN calls.

3.3.2 ANSI Screen editing capabilities of the TEKTRONIX terminal.

The Tektronix 4107 supports ANSI screen editing, which can be used to create screen menus through FORTRAN calls. Different operations like reverse video, highlighting, flashing, cursor control can be achieved through FORTRAN escape sequences. The menus created through this mode are easy to read and helps in preventing errors and also standardizes the inputs to the program (fig. 3a & 3b).
1. Boundary Definition
2. View Directory
3. Define Grid
4. Choose Input Method
5. Change Existing Layout
6. Store Design Layout
7. Design Room & Pillar Sections
8. Design Longwall & Development Sections
9. Retrieve Quality Information
10. Simulation of R&P, LW and Development Sections
11. Add Symbols
12. Final Detailing
13. Terminate Session

Enter your option

Fig. 3a Sample of a option screen menu
**Fig. 3b Sample of design parameter input screen**

<table>
<thead>
<tr>
<th>DEFAULT VALUES</th>
<th>VARIABLES</th>
<th>USER INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>LOWER LEFT X</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>LOWER LEFT Y</td>
<td></td>
</tr>
<tr>
<td>10000.00</td>
<td>UPPER RIGHT X</td>
<td></td>
</tr>
<tr>
<td>10000.00</td>
<td>UPPER RIGHT Y</td>
<td></td>
</tr>
</tbody>
</table>

For default values press return.
3.3.3 Developing the algorithm for Room and Pillar design

The algorithm for designing the room and pillar sections involves

- Identification of the area through four corner points which have to be specified in a clock-wise direction starting from the lower left hand corner.
- The main entry orientation has to be specified by the user (can be any one of the four sides of the identified area).
- The dimensions of the entry, pillar length, pillar width, the staggering distance have to be specified.
- Any of the four sides of the designed pillars can be altered through a change routine. This routine has the capability to rotate the specified side of the pillar through a specified angle. A single pillar, a set of pillars or all the pillars can be changed.

- Pillars can be manually drawn using either the cursor or through key-ins.
- Surveyor's information can be fed into the room & pillar file and the pillar information can be updated.
This routine first constructs the main entry lines parallel to one of the four sides specified by the user. Lines parallel to the main entry are drawn according to the dimensions of the entry and the pillar length. These lines are drawn till the other end of the section is reached. The next step is to draw the cross-cuts according to the pillar length and cross-cut width.

Once the main lines and the cross lines are constructed, the intersections of these two sets of lines give rise to four corners of each of the pillars. If a staggering distance is specified, the cross-lines are offset by the staggering distance on alternative main lines.

The User has the option to try out different designs by changing the various dimensions until a satisfactory design is created. This design can be stored in a data file for future use and also for scheduling and simulation modules. The data file contains the four corner points of the area, the number of pillars, the corner points of each of the pillars, the dimensions of the entries, cross-cut, pillar length and pillar width.

3.3.4. Developing the algorithm for Longwall and development design.
This algorithm is very similar to the room and pillar design algorithm. The algorithm involves

- Identification of the area for designing longwall and development sections.
- Identification of the base line from which the longwalls and the developments start.
- Orientation of the longwalls and developments from the base line.
- Specification of the widths of the longwall faces and the developments.
- Specification of whether the section should start with a longwall or with the development.

The algorithm constructs lines from the base line with the longwall/development widths and the orientation angle. The algorithm computes the total area of the section, the area utilized by the longwalls and developments and the left over area.

The user has the option to change the widths and the orientation for a more efficient design. The final design can be stored in data file for future use and for the simulation and scheduling module. This file contains the four corner points of the area, the co-ordinates of each of the longwall panles, and the development
sections.

3.3.5 Developing the algorithm for retrieving the quality information.

This algorithm interfaces the mine planning program to the geological grid. The quality information of the coal (thickness, sulphur content, ash content, etc.) is used for determining the production quantities and the time involved in mining a specified section. The data is stored in an indexed file with the x & y coordinates of the center of the blocks as the index. Whenever there is a request for the quality from the simulation/scheduling module, this program searches for the blocks which fall within the target area and sends back the weighted average to the requesting module. An error message is echoed in case if there is no data for the requested target area.

3.3.6 Developing algorithm for scheduling and simulating designed room and pillar, longwall and development sections.

The room and pillar, longwall and development sections are designed using the previously mentioned algorithms. The next stage after this design is the
scheduling and simulation of mining of these designed areas.

The volumes to be extracted are based on the areas to be mined and the coal heights involved. In case of room and pillar advance mining, the area to be mined is the entries area and the cross cut area, multiplied by the thickness of the coal. The thickness of the coal is obtained from the geological database, and the area is computed from the data file associated with the room and pillar section. Retreat mining can also be simulated by specifying the percentage extraction of the pillars.

In case of longwall sections the area is obtained from the data file associated with the specified section. The coal thickness is obtained from the geological grid.

The user initially creates the following four sets of data files.

i. Calendar Information: This file contains the year, month and the number of working days in that month. This information is used for all room and pillar, longwall and development sections.

ii. Section Information: This file contains the identification of the room and pillar or the longwall
sections to be mined, the mining advance rate of the section, the number of shifts/day and the cuts/shift.

iii. Exception Information: The information provided in the section information file is average information, and the exceptions which might occur during certain periods are specified in this file. This file contains the year, month, the number of working days, the advance rate, the shifts/day and the cuts/shift information.

iv. Interrelationship information: Mining of each section starts either on a specified date or is related to other sections. This file contains the section identification, a code for specifying whether the section starts on a specified date or is sequential to another section or concurrent to another section. If the code indicates, use of starting date, a starting date is specified for that section.

These four data files are indexed files. The first file namely the calendar information file is indexed by the year and the month. The section information file is indexed by the section identification number. The exception information file is indexed by the section identification, the year and the month. The interrelationship file is indexed by the section identification, the year and the month.
identification number.

The direction of mining is specified in both the cases. The program graphically outputs month by month position of mining of the section. Monthly Production information about the tonnages, quality of the coal is provided for the section.

3.3.7 Developing the interface for plotting the graphics.

The graphical output as produced by the TEK is not compatible with the HI plotter. A plotter driver was developed to obtain the plots of the designed layouts, and the simulation results.

The driver module contains subroutines to create an ASCII file with DM/PL III instructions for driving the plotter. These instructions consist of a single or dual characters, or the characters are followed by additional information. Whenever a move or draw command is executed on the screen an equivalent plotter move and draw command is executed which in turn creates the necessary instructions in the file. Routines have also been written to write different sizes of text at specified angles.
The draw and move routines have provisions for scaling the plotter output. Since the available coordinates for the plotter were from 0,0 to 10000, 10000 units and all the plots are transformed to fit into these boundary limits.

At the end of the session, this ASCII file is sent to the plotter, which in turn interprets the file to generate the plots. The files are sent through the terminal port P0: to the plotter at a baud rate of 600 to the plotter.

3.3.8 Developing the module for directory information

A directory module is essential to keep a log of all the users. This module has two levels of directories, the first level of the directory stores the names of the layout files and their versions. The second level of the directory contains the user name, date of creation/modification, plot scales, and the boundary definition.

3.3.9. Developing the module for displaying the room and pillar parameter window.

The window for displaying the room and pillar
parameters, the length of the pillar \([a]\), the width of the pillar \([c]\), the cross cut width \([c]\) is formed as a segment 50 on the terminal. This window is created with the screen Tektronix co-ordinates and always appears at the top right hand corner of the screen irrespective of the boundaries used in the drafting and design program (fig. 4).

3.3.10. Developing the module for overlays of grid and isopach.

The Tektronix 4107 has the capability to divide the physical screen into a maximum of four virtual screens. These virtual screens can be switched on and off through escape commands. With single screen sixteen colors can be obtained, but with division of the screen into four virtual screens, only two colors (black and white) can be obtained.

For the underground mine planning software, the screen was divided into three virtual screens. The first screen (four colors) is used for displaying the layouts, the second screen (black and white) for the user grid and the third screen (black and white) for the isopachs.

The grid is drawn with the user specified origin and the length of the grid along the \(x\) and \(y\) axes (fig. 5).
The isopachs are drawn with the data files created by the geological programs on the third virtual screen.
Fig. 4 Room and Pillar Design Parameter Window
3.3.11 Developing the module for guarding the program against wrong inputs

All the numerical input first go through a character input routine. The character string is checked for non-numeric values. An error message is returned in case of a wrong input. The inputs through the screen menus also go through these routines. In the case of screen menus, the position of the cursor remains unchanged in case of wrong input.
Fig. 5 User specified grid
CHAPTER 4

APPLICATION

This section outlines the application of the program.

The center line plot and the normal mine layout plot generated by the drafting program (fig. 6) is used to illustrate the room and pillar design, longwall and development design and the simulation/scheduling of these designed sections.

The first step in room and pillar design is the identification of the four corner points of the section. The center nodes 1 and 4 have been identified to be the extreme points of the section. Based on these two center nodes, the four corner points are generated and the area for room and pillar design is defined. The next step is the identification of the main entry orientation, and the line 1-2 identifies the main entry orientation. The pillar length (65 ft.), pillar width (65 ft), cross-cut (20 ft) and staggering distance (20 ft) is defined for design. The starting point for the pillars is next defined (point 1).
Fig. 6 Layout plot
The pillars are generated automatically for the entire section and all the co-ordinates are stored for future use (for surveyor update and for simulation/scheduling) (fig. 7).

The user has the choice to select the dimensions of each of the entry and each row of the pillar (fig. 8).

Even though initially the pillars are designed to be rectangular, any of the four sides of the pillar can be changed through the change option. The user specifies the line to be changed and the offset of the line from the current position. The user also has the provision to change, one, a set or all the pillars in that section (fig. 9).

Figure 10 shows a room and pillar design in a irregular area.

The next figure (fig. 11) illustrates the longwall and the development design. The user specifies the four corner points through the center nodes (nodes 2, 3, 8, 11) for area identification. The next step in the design process is the specification of the base line. Base line 1-4 has been chosen with longwall face starting from point 1. The orientation of the longwall is 90 degrees in relation to the base line. The width of the longwall
Fig. 7 Room and Pillar design with constant dimensions
Fig. 8 Room and pillar design with varying dimensions
Fig. 9 Room and pillar design with changed pillars
Figure 10. Room and pillar design in an irregular area
Fig. 11 Longwall design I
was chosen to be 800 ft. and the development width to be 250 ft. The area utilizations are shown below.

Longwall area = 73.3%
Development area = 22.9%
Left Over area = 3.8%

The next figure (fig. 12) shows a design in the same area with different design parameters. The orientation angle of 90 degrees, longwall face of 880 ft. and the development width of 250 ft. The area utilizations are given below.

Longwall area = 80.6%
Development area = 19.1%
Left Over area = 0.3%

The next figure (fig. 13) shows a design in the same area with different base line and design parameters. The orientation angle of 80 degrees, longwall face of 600 ft. and the development width of 250 ft. The area utilizations are given below.

Longwall area = 55.8%
Development area = 21.5%
Left Over area = 22.7%
Fig. 12 Longwall design II
Fig. 13 Longwall design III
Figure 14 shows the longwall and development design in an irregular area. The utilization figures in this are as below. Line 1-2 was the base line. An orientation of 83 degrees, a face width of 700 and a development width of 200 was chosen for the design.

Longwall area = 58.2%
Development area = 16.4%
Left over area = 25.4%

For both room and pillar design, longwall and development design, the area can be of any odd four sided polygon.

Figure 15 shows a complete layout of a underground mine with room and pillar section and the longwall sections.

The next stage in the planning process is the simulation of the designed room and pillar, longwall and development sections. User identifies the sections to be simulated, and also specifies the starting dates and the relationships of the different sections. The figure 16 shows the simulation of four sections, three room and pillar and one longwall face. Each block represents one
Figure 14. Longwall and development design in an irregular area.
Fig. 15 Complete Underground mine plan
Fig. 16 Total simulation of four sections
month of advance and has the month and the year written on it. Once all the sections have been simulated, the user can specify a target date and check the advances on each of the sections (fig. 17).

The simulation utilizes data from the four different files which have been discussed earlier and with the geological grid. The geological grid contains the following information.

<table>
<thead>
<tr>
<th>center</th>
<th>coal</th>
<th>ash</th>
<th>sulphur</th>
<th>mineable</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>thickness</td>
<td>content</td>
<td>content</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>6 ft.</td>
<td>2%</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

Tables 1, 2, 3a, 3b & 4 illustrate the type of output from the simulation module.
Fig. 17 Simulation results with a specified target date
### Table 2: User-Specified Production Parameters

<table>
<thead>
<tr>
<th>Area No.</th>
<th>Working Days/Month</th>
<th>Production Shifts/Day</th>
<th>Starting Date (Simulated - Mapped)</th>
<th>Tons/Shift</th>
<th>Std. Dev.</th>
<th>Recovery Factor</th>
<th>Minedable</th>
<th>Thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>2</td>
<td>YY.MM.DD</td>
<td>150</td>
<td>10</td>
<td>.9</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>2</td>
<td>YY.MM.DD</td>
<td>150</td>
<td>10</td>
<td>.9</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>2</td>
<td>YY.MM.DD</td>
<td>150</td>
<td>10</td>
<td>1.05</td>
<td>95</td>
<td></td>
</tr>
</tbody>
</table>

* Not needed if these are sequential

** Optional

Table 1
### Table 2: Detail Panel Report

**Panel 1:** Start Date 85 8 12

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Work Days</th>
<th>Production in 1,000's of Tons</th>
<th>Ave/Day</th>
<th>Idle Days</th>
<th>Shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>8</td>
<td>13</td>
<td>43.2</td>
<td>3.32</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>9</td>
<td>20</td>
<td>65.8</td>
<td>3.29</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>10</td>
<td>20</td>
<td>66.7</td>
<td>3.34</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>11</td>
<td>20</td>
<td>69.2</td>
<td>3.46</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>12</td>
<td>20</td>
<td>65.4</td>
<td>3.27</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>1</td>
<td>16</td>
<td>51.4</td>
<td>3.54</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Panel 1:** Finish Date 86 1 25  Total Tons Produced = 361.8*

**Panel 2:** Start Date 86 2 25

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Work Days</th>
<th>Production in 1,000's of Tons</th>
<th>Ave/Day</th>
<th>Idle Days</th>
<th>Shifts</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>2</td>
<td>4</td>
<td>15.3</td>
<td>3.83</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>3</td>
<td>20</td>
<td>62.4</td>
<td>3.12</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>4</td>
<td>20</td>
<td>66.5</td>
<td>3.32</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>5</td>
<td>20</td>
<td>64.8</td>
<td>3.24</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>6</td>
<td>20</td>
<td>69.7</td>
<td>3.49</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>7</td>
<td>20</td>
<td>73.9</td>
<td>3.70</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>8</td>
<td>15</td>
<td>48.6</td>
<td>3.62</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Panel 2:** Finish Date 86 8 23  Total Tons Produced = 401.3*

* Tons x 1,000

Table 2
### Table 3a: UnderGround Evaluation Report

#### Room and Pillar: Area 1

<table>
<thead>
<tr>
<th>Section</th>
<th>Length (feet)</th>
<th>Height of Seam (in.)</th>
<th>Mineable Thickness (in.)</th>
<th>Average Room Dimension (feet)</th>
<th>Ash</th>
<th>Sulphur</th>
<th>BTU</th>
<th>Insitu Tons</th>
<th>Clean Tons</th>
<th>E.G. of Coal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>845</td>
<td>91</td>
<td>83</td>
<td>16.5</td>
<td>46</td>
<td>12.2</td>
<td>2.1</td>
<td>14000</td>
<td>4121</td>
<td>3763</td>
</tr>
<tr>
<td>2</td>
<td>858</td>
<td>91</td>
<td>79</td>
<td>16.5</td>
<td>46</td>
<td>12.2</td>
<td>2.1</td>
<td>14000</td>
<td>4185</td>
<td>3640</td>
</tr>
<tr>
<td>3</td>
<td>900</td>
<td>91</td>
<td>85</td>
<td>16.5</td>
<td>46</td>
<td>12.2</td>
<td>2.3</td>
<td>14000</td>
<td>4427</td>
<td>4138</td>
</tr>
</tbody>
</table>
TABLE 3b UNDERGROUND EVALUATION REPORT

LONCHALL: AREA 1

<table>
<thead>
<tr>
<th>PANEL</th>
<th>PANEL LENGTH (ft.)</th>
<th>FACE LENGTH (ft.)</th>
<th>HEIGHT OF SEAM (in.)</th>
<th>MINEABLE THICKNESS (in.)</th>
<th>RAW TOXNS (in 1000)</th>
<th>CLEAN COAL TOXNS (TONS)</th>
<th>TONS PER SHIFT</th>
<th>STD. DEV.</th>
<th>% QUALITY</th>
<th>% ASH</th>
<th>% SULF</th>
<th>LB. BTU</th>
<th>E.G. OF</th>
<th>RECOVER PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2542</td>
<td>642</td>
<td>79</td>
<td>79</td>
<td>412000</td>
<td>380000</td>
<td>2000</td>
<td>200</td>
<td>12.2</td>
<td>2.1</td>
<td>1400</td>
<td>1.4</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2842</td>
<td>642</td>
<td>81</td>
<td>79</td>
<td>457000</td>
<td>422000</td>
<td>2000</td>
<td>200</td>
<td>12.2</td>
<td>2.1</td>
<td>1400</td>
<td>1.4</td>
<td>.90</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2270</td>
<td>642</td>
<td>80</td>
<td>79</td>
<td>365000</td>
<td>337000</td>
<td>2000</td>
<td>200</td>
<td>12.2</td>
<td>2.1</td>
<td>1400</td>
<td>1.4</td>
<td>.90</td>
<td></td>
</tr>
</tbody>
</table>

Table 3b
### Table 4

**Summary of Simulated Mining for Several Panels**

<table>
<thead>
<tr>
<th>AREA</th>
<th>PANEL</th>
<th>PANEL LENGTH</th>
<th>INSITU TONS</th>
<th>RAW TONS</th>
<th>START</th>
<th>FINISH</th>
<th>CLEAN TONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>3317</td>
<td>53725</td>
<td>49053</td>
<td>84 10 16</td>
<td>85 12 18</td>
<td>44147</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3317</td>
<td>55169</td>
<td>49642</td>
<td>85 01 10</td>
<td>85 01 02</td>
<td>44677</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3417</td>
<td>55013</td>
<td>50229</td>
<td>85 04 19</td>
<td>85 06 28</td>
<td>47717</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3458</td>
<td>55473</td>
<td>50832</td>
<td>85 07 03</td>
<td>85 09 15</td>
<td>46765</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3498</td>
<td>56317</td>
<td>51420</td>
<td>85 09 25</td>
<td>85 11 29</td>
<td>46278</td>
</tr>
</tbody>
</table>

**Total** 229,584
HELP:

The on-line HELP is listed in appendix A. The user has to type in HELP ICAMPS to activate the HELP session. All the HELP information is arranged in levels according to the program structure. For the room and pillar design the HELP information is arranged as follows.

User types ICAMPS at the Topic? prompt.

  e.g.  TOPIC? icamps

Control is transferred to the second level of the Help, wherein the different options available for room and pillar designs are displayed. The user can select these next level options by entering \topic at the Subtopic? prompt.

  e.g. SUBTOPIC? /change
Present methods of estimating coal reserves and quality generally cannot account for the complexities of actual deposits. These calculations are usually oversimplified, thus making their reserve estimates quite different from what is actually available or mineable. The package described in this study will be a significant aid to planners in making these estimates more realistically based on geology and operational information.

The mine design and drafting programs developed in this study will greatly reduce the amount of resources used in generating alternative mine maps. The more alternatives that can be examined, the more likely will the optimal layout be chosen. This selection procedure will allow the planner to design and develop a more efficient, productive mine.

The evaluation program will provide more accurate and detailed reports of productivity, operational and other output from the mine. The increased accuracy will eliminate or minimize the risks involved in investing on
extension of existing mines or a new mining venture. The program will provide the decision maker a more accurate forecast so that decisions can be made quickly and effectively.

The unique strength of the mine planning system is that it can be used to develop and design new mines and also control and schedule existing mine operations. Mine planning involves engineering design of layouts, determination of costs, production levels and manpower requirements, scheduling of mining operations to meet both sequential as well as concurrent extraction. It requires the generation of a mine plan and report which includes the tonnage, quality and cost information necessary to evaluate the feasibility or the profitability of the proposed plan. This program allows an exhaustive examination and reporting of all these functions.

Mine layouts are a necessary input to mine planning. The need to draw several layouts is implicit in determining the optimum plan. But manual drawing of layouts is labor intensive and tedious and yet, without an engineering drawing of the layout, production rates and levels cannot be computed. Because conventional manual design is so labor intensive only one plan is examined and executed. The opportunities to consider
alternatives are precluded by time and cost constraints. Interactive graphics offers the only feasible solution to expedite design and planning procedures and to examine many alternative mining strategies. Design and planning engineers, surveyors, and other technical personnel associated with mine production and operation will benefit from this innovative procedure.

Most mining companies operate with very limited engineering staff. In manual planning much of this scarce resource is spent on low level computations. This time could be spent more profitably on creative and in-depth design and analysis.

With computer-aided design it is easy to generate production and quality estimates for management on short notice. Management can quickly evaluate the effects of changing plans that are brought on by unforeseen problems such as abrupt changes in geology or major equipment failures. It would require 2-3 weeks to generate and critically evaluate the effects of a long term planning change if conventional manual methods were employed.

Computer-aided design eliminates most errors induced through manual data entry. Geological information is very extensive and subject to different interpretations. This system extracts geological and
other related information directly from the user's database so that the planning results are both consistent and reproducible.

Computer-aided mine planning allows plans to be quickly updated. Only the areas that are affected by mining or planning need to be redrawn.

The scale of plans can be manipulated freely allowing large or small plans to be plotted from the same file. Enlargements of selected areas can be drawn from the base plan file.

In conventional manual methods it is almost impossible to collect and collate different planning information in one place in order to design and draft a mine plan. For example, information relating to iso-values (thickness, ash, sulfur), geological data (roof and floor conditions), production (rates and levels), ventilation (pressure and quantity), are difficult to manually integrate. In manual planning this information is examined sequentially and in a piece meal fashion. This program allows all these features to be examined simultaneously at the time the plan is being developed.

By law, mines must maintain up to date maps of mined out areas. Deviation from the plan must be documented by continuously surveying the active working
areas and reflecting these measurements to update the mine maps. Computerized mine drafting will accept the surveyor's notes and automatically update the mine map thus saving much time and eliminating opportunities for manual error.
CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

This study was aimed at developing a mine design and drafting aid for planning engineers in the coal industry. Such a package would greatly reduce the time and effort required to develop mine maps. This reduction would enable the generation and evaluation of a number of alternative layouts.

The package developed in this study is a user friendly interactive design and drafting package that automates the planning and designing of underground mines. It also successfully simulates the designed layout providing the user with the production quantities, quality of the coal and the time scale for the mining activities. The program interfaces to the geological grid to obtain the information about the coal seam. All the graphics which appear on the screen can be easily plotted with different scaling factors.

There are additional design routines that could be included in the present package to increase its versatility.
- Interfacing a ventilation network analysis program with the layout program to compute pressure/quantity requirements.

- Incorporating a belt simulation program to design and simulate coal transportation in underground mines.

- Use of pull down menus to make the program more user friendly.

- Use of a mouse or a light pen instead of the cursor for pointing purposes.

- Creating windows for displaying the various types of optimizing and design curves for design purposes.

The addition of these features would greatly increase the package's capabilities and render manual planning to a minimum.
BIBLIOGRAPHY

Adel, G., Haycocks, C., Lucas, J.R., 1983,

Barczak, T.M., 1983,

Barua, S. L., Kim, Y.C., 1984,

Chatterjee, P. K., Scheck, D. E., Li, B. S., 1985,


Topuz, E., Nasuf, E., 1985, CONSIM - An interactive micro-computer program for

Wright, K. W., 1985,
Computer-aided mapping and design - Their potential for mining use. Mining Engineering, Vol. 37, No. 6, pp 550-553, June.
APPENDIX A
ICAMPS (Integrated Computer Aided Mine Planning Software) is a Menu driven program with one main program and fourteen modules.

NOTE: TO OBTAIN THE INFORMATION ON ANY OF THE MODULES, TYPE THE CORRESPONDING TOPIC'S NAME.

** MORE INFORMATION CAN BE OBTAINED ON THOSE OPTIONS IN CAPITALS. THOSE WITH / IF FRONT ARE ACCESSED FROM SUBTOPIC?, AND THE OTHERS ARE ACCESSED FROM TOPIC? OPTIONS OF THE HELP SYSTEM.**

The structure of the software is shown below

```
       MAIN
       |
       |
       |
       BOUNDARY  GRID  CHANGE  R-P  LW  SYMB
       |
       |
       DIRECTORY  INPUT  STORE  QUALITY  SINUL  TERMINATE
```

** I MAIN **

The main program is responsible of initializing, setting up the MAIN MENU commands and passing control to an from the MENU commands as well as the fourteen modules.

** I BOUNDARY **

This menu allows the user to establish a current session working area definition. Four inputs from the user establishes the working area definition.

1. Lower Left Hand X co-ordinate
2. Lower Left Hand Y co-ordinate
3. Upper Right Hand X co-ordinate
4. Upper Right Hand Y co-ordinate

    3,4

I DIRECTORY

This module keeps track of the users, the files they have created, the date of creation, modification, plot scale, boundary definition, etc. Each time the user terminates a session he has an option to update the directory information.

I GRID

This module sets up the user defined geometrical grid. User specifies the following quantities.

1. X co-ord of the ORIGIN
2. Y co-ord of the ORIGIN
3. Length of the grid along the X-axis
4. Length of the grid along the Y-axis

A GRID is drawn with this information, by dividing the length into 10 equal parts.

DEFAULTS: The Lower Left Hand corner of the Boundary definition is used for the ORIGIN. The Upper Right Corner of the Boundary is used for determining the length along X-axis and Y-axis.

I INPUT

This MENU allows the user to specify the input methods. The following options are provided for the user.

/OLD  /KEY-BOARD /CURSOR /DIGITIZER TO-MAIN menu

2 QUALIFIERS

/OLD

Takes the user to the DIRECTORY module from which he can select any of the
already created files.

There are three scratch files X.DAT, Y.DAT, Z.DAT which are used for temporarily storing the layouts, without accessing the directory. Once the layout is finalized, the files can be stored permanently through the directory.

\[
x.dat \quad z.dat
\]
\[
y.dat \quad \text{DIRECTORY}
\]

After the specified file is retrieved, first the center line plot is drawn automatically from the old data. After the center line plot the following menu appears.

\[
\text{CHANGE} \quad \text{TO-MAIN}
\]
\[
\text{layout} \quad \text{menu}
\]
\[
\text{CONTINUE} \quad \text{with plot}
\]

/KEY-BOARD

This option is used for creating new layouts. The user defines his layout (orientation of the drives, intersections) through absolute co-ordinates.

/CURSOR

This option is used for creating new layouts. The user defines his layout (orientation of the drives, intersections) through CURSOR key-ins.

/DIGITIZER

This option is not available yet (no digitizer available yet).

1 TO-MAIN

This option takes the user back to the MAIN MENU.

1 CONTINUE

This is the second stage of the plot operation.
This option, works on the center line plot and develops the total layout with drives, center nodes, etc...

There is an option to display the corner SNAP-ON points for identifying the areas for R-P or LW sections.

1 SNAP-ON

These are numbers used to identify the corners of the drives, etc...

NOTE: If there are large number of center nodes in a short distance, it is advisable not to use this option. See AREA.

1 CHANGE

This option is used to change the designed layout. User has the option to delete, add new drives, change drive widths, eliminate center nodes, change center node position, etc...

```
CHANGE

/ELIMINATE node
/EXTEND drive
/CHANGE param
/VIEW param

/INSERT node
/CONTINUE with plot
/TO-HAIN menu
```

2 QUALIFIERS

/ELIMINATE

A center node can be eliminated from the designed layout thorough this option. The user specifies the center node number to be deleted.

/INSERT

This option is used for inserting new center nodes between the existing center nodes. The inserted node becomes part of the old drive. Drive width, Junction width and Junction flags (NODE-FLAGS) must be specified by the user.
/EXTEND

Through this option the user can extend the existing drives or create new junctions from the existing center nodes, thereby opening up new drives.

(NODE-FLAGS)

1 NODE-FLAGS

Each center node has four flag parameters. These are 1. Drive number, 2. Junction connectivity flag, 3. Drive width flag and 4. Junction width flag as shown below.

```
NODE-FLAGS

/DRIVE /DRV-WID
number drive

/JUNCTION /JUN-WID
flag
```

2 QUALIFIERS

/DRIVE

The Drive number is simply a unique integer number assigned to a drive (roadway, entry...) to distinguish it from another drive. Drive number associates a center node with a drive. A drive can be made up of many center nodes.

<table>
<thead>
<tr>
<th>.6</th>
<th>.7</th>
<th>.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MDOES</td>
<td>DRIVE NUMBER</td>
</tr>
<tr>
<td></td>
<td>1,2,3</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4,5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>6,7,8</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>.4</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>.2</td>
<td>.3</td>
</tr>
</tbody>
</table>
NOTE: The term DRIVE is used to designate an ENTRY or a ROADWAY.

/JUNCTION

Junction flag defines the connectivity pattern of two intersecting drives.

A flag of '0' indicates that there is no junction at that node point.

A flag between '1' and '99' indicates that there is a new drive starting at the node point.

A flag greater than '100' indicates that the intersection is formed by two existing drives.

<table>
<thead>
<tr>
<th>NODES</th>
<th>JUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3,6,8</td>
<td>0</td>
</tr>
<tr>
<td>2,4</td>
<td>7</td>
</tr>
<tr>
<td>5,7</td>
<td>105</td>
</tr>
</tbody>
</table>

/DRV-WID

Each center node is associated with a width parameter. This determines the width of a drive at that node.

See the diagram below:
This parameter defines the openings at the intersections. If the node has no intersection then junction width is 0. In the diagram below, nodes 1, 3, 4, 5, 7, 9 and 11 have no intersections, hence the junction width parameter is zero for these nodes.

See the diagram below
1 STORE

This option is used to store the designed layouts. User has the option to store the layout in either the scratch files x.dat, y.dat, z.dat or permanently through the DIRECTORY option.

See /OLD in INPUT for additional information

```
<table>
<thead>
<tr>
<th>.8</th>
<th>NODES</th>
<th>JUNC-WIDTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 3, 4, 5, 7, 9, 11</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>2, 6</td>
<td>350</td>
</tr>
<tr>
<td>1.6</td>
<td>8, 10</td>
<td>200</td>
</tr>
</tbody>
</table>
```

\[ .1 .2 .3 .4 .5 \]

1 R-P

This module allows the user to define the room and pillar sections. This module is accessed after the initial layout is developed. User identifies the section to be designed through four corner points and inputs the necessary design parameters.

```
R-P

AREA /VARATM /START /NEW
identi point pillar

/MAIN /DESIGN /CHANGE /SURVEYOR
entry geometry input
```

2 QUALIFIERS
User specifies the orientation of the main entrance through this option. User can select any of the four sides along which the main entrance is oriented.

See the diagram below:

```
[2]
    /    /
   /     /
  /      /
 [1] 0, 0, 0 [3] (e.g. main entry along side [1])
  \      \        \        
   \      \        \        
    \      \        \        
    0, 0, 0          0, 0
[4]
```

User has the choice of designing all the entries and rows of pillars to be of the same constant dimensions or can specify the widths of each entrance and each row of pillars.

see the fig. below.
Pillar length, width, cross-cut and staggering of pillars are specified in this menu. Initially all the pillars are designed rectangular, but their shape be changed through /CHANGE option.

[3] (e.g. first row of pillars are wider than the second row of pillars)
/START

This option is helpful, in case the room and pillar section is irregular in shape. Any one of the corner points can be specified for start of the pillars.

See the fig. below

User enter either 1, 2, 3 or 4 for starting point for the pillars.

/CHANGE

Even though the pillars are designed to be rectangular or square initially, through this option the user can change the shape of one pillar, a set of pillars or all the pillars in that section. User inputs the side to be changed and the angular offset of the line from the current position.

See the fig. below.
Line [3] is changed by pivoting at point 4 and an offset of 15 degrees from the initial position.

This option is used to drawing pillars through cursor or keyboard. User specifies the number of points and the co-ordinates of the pillar. These manually defined pillars are stored along with those designed by the program. This option is useful to draw irregular pillars.

User can update the mine plans through this options. After the pillars have been mined, the user supplies the surveyor's notes for the pillars in a section. This information can be either entered interactively, or through a data file. The data file option is useful if large number of pillars have to be updated.
/INTERACTIVE

The actual mined co-ordinates of the pillars are entered by the user. The points can be entered through

1. Cursor
2. By absolute co-ordinates and
3. Through a reference point,
   length and azimuth from the reference point.

/DATA-FILE

This option is used to update a large number of pillars. The user inputs the section identification, pillar identification, and the offsets from the center span points (left offset, right offset, etc...). Once the data file is created outside the program, the user specifies the file name to this module for updating the pillars.

1 AREA

This option helps the user in identifying the area for longwall or room and pillar design. Depending on the complexity of the layout the user has the following options to define the area for design.

```
AREA
  /CORNER
  /EDG-LEN
  CURSOR
  /CENTER

2 QUALIFIERS

/CORNER

Corner nodes are the SNAP-ON nodes which have to be specified during the second stage of the plot option.

See CONTINUE
/EDG-LEN

User specifies the two CORNER nodes for the base line and the length of the section from this base line. A +ve or -ve direction has to be specified. A section is developed which is based on the two corner nodes and a length perpendicular to this base.

```
20
 length
 base <------------>

21
```

/.CENTER

User specifies two (for room and pillar) or four (for longwall) center nodes for defining the section for design. The program displays certain key corner points at these node positions. User then selects the desired points amongst these temporary points for design. These temporary numbers are written in green.

```
.8 .9 .10 .11

User-Inputs
For room and pillar - 8 and 5
For Longwall - 5, 8, 10 and 7
```

.5 .6 .7
Longwall design is done through this module. Identification of the area for LW design is very similar to the R-P design. This module optimizes the LW design through user interaction.

1 LW

AREA /VARATN /START
ident1 | point
/BASE /DESIGN /START-SEC

2 QUALIFIERS

/BASE

Base is an imaginary base line (usually one of the sides of a four sided polygon), from which the panels start. Any of the four sides can be selected to be the base for designing the panels.

See fig. below.

Sides [1], [2], [3] or [4] can be chosen as a base line.

See fig. below.
(e.g. Side [1] is chosen as the base line)

/VARATN

User has the option to design either with constant face panels and development widths or specify the widths of individual panels and development widths through this option.

/DESIGN

User specifies the orientation of the panels relative to the base line, the width of the panels and the width of developments through this option.

/START

The design can be started from either end of the specified side. For e.g. if the BASE line is [2], user can start the design from either point 2 or point 3. (see /BASE)

/START-SEC

This option is used to design the LW section, starting with either a development or with a longwall panel. Depending on whether it is a development or longwall panel, the design starts from the specified start point (see /START).

1 QUALITY

This module is linked to the geological database of the property. User has the following options through which he/she can determine the coal quality at a point or of an area (four sided polygon).

QUALITY

/POINT

/AREA

quality

/MODE

/MODE-MODE

quality

2 QUALIFIERS

/POINT
This option is used for retrieving the information of a point through either co-ordinate or through cursor keys.

/NODE

User specifies the CENTER node at which the quality has to be determined.

/AREA

Four points are identified through the cursor, and the resultant quality is the weighted average of the blocks to which the points belong.

/NODE-NODE

The two CENTER nodes between which the quality is to be determined are specified by the user.

1 SYMB

The symbols are user defined and site specific. There are symbols for fan, doors, air crossings, etc... User inputs the scale factor and the co-ordinates at which the origin of the symbol is to be positioned.

1 SIMUL

This module is used for simulating the mining operation of the designed bleeders, developments, longwall sections and other room and pillar sections.

SIMUL

<table>
<thead>
<tr>
<th>/CREATE</th>
<th>/OUTPUT</th>
<th>/START</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>point</td>
<td></td>
</tr>
</tbody>
</table>

/GRID /DESIGN /START-SEC

2 QUALIFIERS

/CREATE

User initially creates four data files before initiating the simulation.
CALENDER: This file contains the year, month and the number of working days in that year and the month. This file is common to all the sections of the property.

/SECTION

SECTION: User specifies the Section ID, the average shifts, advance, cuts for that section. The calculations are based on these figures, except on certain times wherein the information is used from the /EXCEPTION file.

/EXCEPTION

EXCEPTION: Certain time of the year, the shifts, etc. might be different from the average figures. This file stores the section ID and the exception values for that section in the specified times.

/RELATION

RELATION: This file stores the starting dates of the sections. If a section's start date depends (sequential or concurrent) to a previously defined section, an appropriate information is coded in this data file. The program automatically takes this relationship into consideration, to simulate that section.

I TERMINATE

Ends the current session. User should make sure that all the necessary files are saved before terminating the session.
APPENDIX B
SUBROUTINES TO INTERFACE TERMINAL ROUTINES WITH FORTRAN

THE SUBROUTINES ARE CALLABLE FROM FORTRAN
CREATED BY SRIDHAR CHALUVADI ON 09-19-85
FOR SPECIAL INVESTIGATION
THESE ROUTINES ARE 4100 - STYLE PARAMETERS
UPDATED FOR THESIS WORK. 03-20-1986

SUBROUTINE TO OPEN A NEW SEGMENT

SUBROUTINE SGOPEN(K)
    CHARACTER*3 SOUT
    INTEGER ESC,S,0
    DATA ESC,S,0/27,33,79/
    CALLING THE SUBROUTINE FOR CONVERTING K
    CALL INT2(K,SOUT)
    WRITE(*,1)ESC,S,0,SOUT
1 FORMAT(1X,2A2,A3)
    RETURN
END

SUBROUTINE TO FILL THE PATTERN WITH THE DESIRED COLOR/SHAPE

SUBROUTINE FILPTN(K)
    INTEGER ESC,M,P
    CHARACTER*3 SOUT
    DATA ESC,M,P/27,77,60/
    CALLING THE SUBROUTINE FOR CONVERTING K
    CALL INT2(K,SOUT)
    WRITE(*,1)ESC,M,P,SOUT
1 FORMAT(1X,2A2,A3)
    RETURN
END

SUBROUTINE TO BEGIN A PANEL

SUBROUTINE BGNPANEL(X,Y,SOUN)
    INTEGER X,Y,SOUN,ESC,L,P
    DATA ESC,L,P/27,76,99/
    CHARACTER*5 A
    CHARACTER*3 B
    CALLING THE SUBROUTINE FOR CONVERTING SOUN
    CALL INT2(SOUN,B)
    CALLING THE SUBROUTINE FOR CONVERTING X,Y

CALL HIY(x,y,a)
WRITE(*,10)ESC,L,P,A,B
10 FORMAT(1X,3A2,A5,A3)
RETURN
END

------------------------------------------------------------------------
 SUBROUTINE TO DRAW FROM THE PRESENT POSITION TO X,Y
------------------------------------------------------------------------

 SUBROUTINE TDRAW(X,Y)
 INTEGER X,Y,ESC,L,G
 CHARACTER*5 A
 DATA ESC,L,G/27,75,71/
 CALLING SUBROUTINE FOR CONVERTING X,Y TO CHARACTER FORM
 CALL HIY(X,Y,A).
 RETURN
END

------------------------------------------------------------------------
 SUBROUTINE TO END THE SEGMENT AND THE PANEL
------------------------------------------------------------------------

 SUBROUTINE ENDSMNT
 INTEGER ESC,L,C
 DATA ESC,S,C/27,75,71/
 WRITE(*,100)ESC,S,C
100 FORMAT(1X,3A2)
 RETURN
END

------------------------------------------------------------------------
 SUBROUTINE TO END THE THE PANEL
------------------------------------------------------------------------

 SUBROUTINE ENDPANEL
 INTEGER ESC,L,E
 DATA ESC,L,E/27,75,71/
 WRITE(*,100)ESC,L,E
100 FORMAT(1X,3A2)
 RETURN
END

------------------------------------------------------------------------
 SUBROUTINE TO MAKE THE DIALOG AREA TRANSPARENT
------------------------------------------------------------------------

 SUBROUTINE DINDEX
 INTEGER ESC,L,I,COLOR(3)
 CHARACTER*3 ICOLOR(3)
DATA ESC,L,1/27,76,73/
DATA COLOR,3/0,0/
DO 10 I=1,3
10 CALL INTEGR1(COLOR(I),COLOR(I))
WRITE(*,15)ESC,L,1,COLOR(I),COLOR(I),COLOR(I)
FORMAT(1X,3A2,3A3)
RETURN
END

C
---------------------------------------------
C SUBROUTINE TO CLEAR THE SCREEN
---------------------------------------------
SUBROUTINE PAGE
INTEGER ESC,FF
DATA ESC,FF/27,12/
WRITE(*,10)ESC,FF
10 FORMAT(1X,3A2)
RETURN
END

C
---------------------------------------------
C SUBROUTINE TO SET code tek THROUGH PROGRAM
---------------------------------------------
SUBROUTINE MTEK
INTEGER ESC,PERCENT,BANG,
CHARACTER*3 SCUT
DATA ESC,PERCENT,BANG/27,37,33/
DATA TTEK/0/
CALL INTEGR1(TTEK,SOUT)
WRITE(*,10)ESC,PERCENT,BANG,SOUT
10 FORMAT(1X,3A2,A3)
RETURN
END

C
---------------------------------------------
C SUBROUTINES FOR CREATING CHARACTER INFORMATION
---------------------------------------------
C
---------------------------------------------
C SUBROUTINE FOR FIXING THE TEXT PRECISION
---------------------------------------------
SUBROUTINE TXTPR
INTEGER ESC,M,0
C USING THE STROKE PRECISION
DATA ESC,M,0/27,77,81/
DATA IPRE/2/
CHARACTER*3 SCUT
CALL INTEGR1(SCUT)
WRITE(*,10)ESC,M,0,SOUT
10 FORMAT(1X,3A2,A3)
SUBROUTINE FOR FIXING THE TXT SIZE

SUBROUTINE TXTSIZ
INTEGER ESC,M,C,D(I)
CHARACTER*2 D1(I)
DATA 0/30,30,10/
DATA ESC,M,C/27,77,67/
DO 10 II=1,3
10 CALL INTEGR(DII),DII)
WRITE(*,15)ESC,M,C,DII(1),DII(2),DII(3)
:5 FORMAT(1X,3A2,3A3)
RETURN
END

SUBROUTINE FOR TXT COLOR

SUBROUTINE TXTCOL(INDEX)
INTEGER ESC,M,T,INDEX
DATA ESC,M,T/27,77,84/
CHARACTER*3 SOUT
CALL INTEGR(INDEX,SOUT)
WRITE(*,10)ESC,M,T,SOUT
10 FORMAT(1X,3A2,3A3)
RETURN
END

SUBROUTINE FOR LINE COLOR

SUBROUTINE LINECOL(INDEX)
INTEGER ESC,M,L,INDEX
DATA ESC,M,L/27,77,76/
CHARACTER*3 SOUT
CALL INTEGR(INDEX,SOUT)
WRITE(*,10)ESC,M,L,SOUT
10 FORMAT(1X,3A2,3A3)
RETURN
END

SUBROUTINE FOR ACTUAL TEXT PLACEMENT

SUBROUTINE TXTPLT(S)
CHARACTER*40 S
CHARACTER*50 STRING
CHARACTER*243 DARRAY
INTEGER ESC,L,T
DATA ESC,L,T/27,73,24/
WRITE(*,10)ESC,L,T,S
10 FORMAT(1X,3A2,4X)
RETURN
END

C---------------------------------------------------------------
C   SUBROUTINE FOR MOVING THE BEAM POSITION WITHOUT DRAWING
C---------------------------------------------------------------
SUBROUTINE MOVE(X,Y)
INTEGER ESC,L,F,X,Y
DATA ESC,L,F/27,73,70/
CHARACTER*5 A
CALL HI(Y,X,A)
WRITE(*,10)ESC,L,F,A
10 FORMAT(1X,3A2,4X)
RETURN
END

C---------------------------------------------------------------
C   SUBROUTINES FOR ENABLING THE GIN, INKING, RUBBERANDING,....
C---------------------------------------------------------------

C---------------------------------------------------------------
C   SUBROUTINE FOR ENABLING THE GIN WITH LOCATOR
C---------------------------------------------------------------
SUBROUTINE GINABLE(INUMBER)
INTEGER ESC,I,E,FC,INUMBER
CHARACTER*3 A1,A2
DATA ESC,I,E,FC/27,73,69,3/
CALL INTEGR(FC,A1)
CALL INTEGR(INUMBER,A2)
WRITE(*,10)ESC,I,E,A1,A2
10 FORMAT(1X,3A2,4X)
RETURN
END

C---------------------------------------------------------------
C   SUBROUTINE FOR ENABLING THE GININKING FEATURE
C---------------------------------------------------------------
SUBROUTINE GININK
INTEGER ESC,I,FP,IM
CHARACTER*3 A1,A2
DATA ESC,I,FP,IM/27,73,0,2/
CALL INTEGR(FP,A1)
CALL INTEGR(IM,A2)
WRITE(*,10)ESC,I,I,A1,A2
SUBROUTINE FOR ENABLING THE SINE RUBBER ANDING THE FEATURE

SUBROUTINE SINE_RUBBER
INTEGER ESC, I, R, FP, IMODE
DATA ESC, I, R, FP, IMODE / 27, 73, 82, 0, 2 /
CHARACTER*5 A1, A2
CALL INTESP(FP, A1)
CALL INTESP(IMODE, A2)
WRITE(*, 10) ESC, I, R, A1, A2
10 FORMAT(1X, 3A2, 2A3)
RETURN
END

SUBROUTINE FOR REPORTING THE COORDS OF THE POINT IN GIN MODE

SUBROUTINE GINRPT
INTEGER ESC, I, P, FP
CHARACTER*5 SOUT
DATA ESC, I, P, FP / 27, 73, 80, 6 /
CALL INTESP(FP, SOUT)
WRITE(*, 10) ESC, I, P, SOUT
10 FORMAT(1X, 3A2, A3)
RETURN
END

SUBROUTINE FOR CONVERTING THE TEK XY COORD TO HOST FORMAT

SUBROUTINE XYTOI(STR, X, Y)
INTEGER X, Y
CHARACTER*5 STR
DIMENSION I(15)
X=0
Y=0
DO 10 II=1, 5
10 J(II)=ICHAR(STR(II+1))
C DECODING THE HIGH BITS OF Y
C READING THE LEAST 5 BITS OF J(1)
CALL MVBITS(J(1), 0, 5, Y, 7)
C DECODING THE EXTRA BITS OF Y
CALL MVBITS(J(2), 2, 2, Y, 0)
C DECODING THE HIGH BITS OF Y
CALL MVBITS(J(3), 0, 5, Y, 3)
C DECODING THE EXTRA BITS OF X
CALL MVBITS(J(2),0,E,X,0)
C DECODING THE HIGH BITS OF X
CALL MVBITS(I(5),0,S,Y,2)
RETURN
END

C---------------------------------------------------------------------
C SUBROUTINE FOR GENERATING THE MARKER
C---------------------------------------------------------------------
SUBROUTINE SETMARK(MTYPE)
INTEGER ESC,M,MTYPE
CHARACTER*3 SGUT
DATA ESC, M/27,77/
CALL INTEGER(MTYPE,SGUT)
WRITE(*,10)ESC,M,MTYPE,SGUT
10 FORMAT(1X,3AE,A3)
RETURN
END

C---------------------------------------------------------------------
C SUBROUTINE FOR PLACING THE MARKER
C---------------------------------------------------------------------
SUBROUTINE PLCMARK(X,Y)
INTEGER X,Y,ESC,L,H
CHARACTER*5 A
DATA ESC, L,H/27,76,72/
CALL HIY(X,Y,A)
WRITE(*,10)ESC,L,H,A
10 FORMAT(1X,3AE,A5)
RETURN
END

C---------------------------------------------------------------------
C SUBROUTINE FOR SETTING THE VIEW DISPLAY CLUSTER
C---------------------------------------------------------------------
SUBROUTINE DISCLUST(ICOUNT,IARRAY)
INTEGER IARRAY(*),ICOUNT,ESC,R,G
DATA ESC,R,G/27,68,61/
CHARACTER IARRAY(*)
C CALLING THE CONVERSION OF ARRAY FOR TEK MODE
CALL ARRAY(ICOUNT,IARRAY,ICOUNT,IARRAY)
WRITE(*,10)ESC,R,G,IARRAY
10 FORMAT(1X,3AE,A<ICOUNT>)
RETURN
END

C
SUBROUTINE SURSET(I=COUNT,1,ARRAY)
INTEGER I=ARRAY(4),OCOUNT,ESC,R,D
DATA ESC,R,3/27,92,66/
CHARACTER D=ARRAY(243)
C CALLING THE CONVERSION OF ARRAY FOR TEK MODE
CALL ARRAY(I=COUNT,1,ARRAY,OCOUNT,D=ARRAY)
10 FORMAT(1X,3AE,4<OCOUNT>)
RETURN
END

C

C SUBROUTINE FOR SETTING THE SURFACE VISIBILITY

SUBROUTINE SURVIS(I=COUNT,1,ARRAY)
INTEGER I=ARRAY(4),OCOUNT,ESC,R,l
DATA ESC,R,1/27,92,73/
CHARACTER D=ARRAY(343)
C CALLING THE CONVERSION OF ARRAY FOR TEK MODE
CALL ARRAY(I=COUNT,1,ARRAY,OCOUNT,D=ARRAY)
10 FORMAT(1X,3AE,4<OCOUNT>)
RETURN
END

C

C SUBROUTINE FOR SELECTING THE VIEW FOR DRAWING

SUBROUTINE SelVIEW(NVIEW)
INTEGER ESC,R,C
DATA ESC,R,C/27,92,67/
CHARACTER*3 OUT
CALL INTEGER(NVIEW,OUT)
WRITE(*,10)ESC,R,C,OUT
10 FORMAT(1X,3AE,4)
RETURN
END

C

C SUBROUTINE FOR SETTING THE VIEW ATTRIBUTES

SUBROUTINE VIEWATTR(I=ATTR)
INTEGER I=ATTR,ESC,R,A
DATA ESC,R,A/27,92,66/
15 FORMAT(1X,3A2,3A3)
RETURN
END

C------------------------------------------------------------------------
C SUBROUTINE FOR DEFINING THE KEY MACRO
C THIS IS FOR SET KEY EXECUTE CHARACTER
C------------------------------------------------------------------------
SUBROUTINE KEYEXEC(ICHARA)
  CHARACTER OUT*3
  INTEGER ESC,K,Y
  DATA ESC,K,Y/27,75,97/
  CALL INTEG(ICHARA,OUT)
  WRITE(*,10)ESC,K,Y,OUT
10  FORMAT(1X,3A2,A3)
RETURN
END

C------------------------------------------------------------------------
C SUBROUTINE FOR DEFINING THE MACRO
C------------------------------------------------------------------------
SUBROUTINE DEFCMACRO(CHARA,LENTH,STR)
  CHARACTER STR*80,AARRAY*243,OUT*3
  INTEGER LENTH,OCCOUNT,CHARA
  INTEGER ESC,K,D
  DATA ESC,K,D/27,75,68/
  CALL INTEG(ICHARA,OUT)
  CALL INTEG(LENTH,STR,OCCOUNT,AARRAY)
  WRITE(*,10)ESC,K,D,OUT,AARRAY
10  FORMAT(1X,3A2,A3,A*(OCCOUNT))
RETURN
END

C------------------------------------------------------------------------
C SUBROUTINE FOR CONVERTING CHARACTER ARRAY INTO ARRAY
C------------------------------------------------------------------------
SUBROUTINE ADPECT(LENTH,STR,OCCOUNT,AARRAY)
  CHARACTER STR*60,AARRAY*243
  INTEGER*4 LENTH,OCCOUNT
  DIMENSION IA(60)
  DO 10 IJ=1,LENTH
       IA(IJ)=ICHARA(STR(IJ:IJ))
10  print *, ia', ia(ij)
  CALL ARRAY(LENTH,IA,OCCOUNT,AARRAY)
  WRITE(*,15)AARRAY
SUBROUTINE gtext1(k,str)
INTEGER ESC,l,t,k
BYTE STR(80),OUT(80)
character*3 parm
data esc,l,t/27,76,04/
DO 50 JK=1,K
50 OUT(JK)=STR(JK)
call integer(k,parm)
write(*,100) esc,l,t,parm,out(JK), JK=1,K
100 format(1x,a2,2a2,a3,<K),A1)
return
end

SUBROUTINE gsize1(N)
INTEGER ESC,M,CX
INTEGER*4 C(3)
CHARACTER*243 CA
DATA ESC,M,CX/27,77,67/
GO TO (10,15,20,30)N
10 C(1)=19
C(2)=27
C(3)=4
GO TO 35
15 C(1)=39
C(2)=59
C(3)=8
GO TO 35
20 C(1)=50
C(2)=80
C(3)=10
GO TO 35
30 C(1)=15
C(2)=20
C(3)=10
35 CALL ARRAY(3,C,M,CA)
write(*,100)ESC,M,CX,CA(2:3),CA(4:5),CA(6:7)
100 format(1x,3a2,3a2)
return
end
CHARACTER*5 A,B
DATA ESC,R,W/27,32,37/
C CALLING SUBROUTINE FOR CONVERTING X,Y TO CHARACTER FORM
CALL HIY(XL,YL,A)
CALL HIY(XU,YU,B)
WRITE(*,10)ESC,R,W,A,B
10 FORMAT(1X,3A2,2A5)
RETURN
END
SUBROUTINE FOR PLANNING THE R/P SECTION

SUBROUTINE SIMPLIFR

<table>
<thead>
<tr>
<th>DIMENSION X(4), Y(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>character ANS, REDY, FNAME<em>10, AREA</em>10, FNE<em>10, AREA</em>6</td>
</tr>
<tr>
<td>CHARACTER*80 CHVAL(10)</td>
</tr>
<tr>
<td>REAL VALUES(10)</td>
</tr>
<tr>
<td>COMMON /FILEST/IOLD</td>
</tr>
</tbody>
</table>

COMMON AREA TO IDENTIFY WHETHER THE SUB IS CALLED FROM LW DESIGN

| COMMON /LW/LWST, ARE, IPD |
| data REDY/*' ' |
| DATA X,Y*0.0 |
| DATA LWST/0 |
| VALUES(1)=120.0 |
| VALUES(2)=20.0 |
| VALUES(3)=65.0 |
| VALUES(4)=0.0 |
| CHVAL(1)="120.0" |
| CHVAL(2)="20.0" |
| CHVAL(3)="65.0" |
| CHVAL(4)="0.0" |

C PRINT *, 'CHOOSE ONE OF THE FOLLOWING OPTIONS'

C CALL MAINSCN(1,CHVAL)

C CHVAL(1)="<1> DESIGN NEW R/P SECTION" |
| CHVAL(2)="<2> VIEW DESIGNED R/P SECTION" |
| CHVAL(3)="<3> RETURN TO MAIN MENU" |
| CHVAL(4)="ENTER YOUR OPTION" |

CALL OPTIONS(4,CHVAL,IDCHOICE)

C CHVAL(1)="YES" |
| CHVAL(2)="NO" |
| CHVAL(3)="NO" |

C CALL AVALSCN(2,CHVAL)

C IF(CHVAL(1).EQ. 'YES') IDCHOICE=1 |
| IF(CHVAL(2).EQ. 'YES') IDCHOICE=2 |

IF(IDCHOICE .EQ. 3) RETURN

IF (IDCHOICE .EQ. 2) THEN

IOLD=1 |
| LWST=0 |

C PRINT *, 'INITIAL', IOLD, LWST |

CALL SRIPLR(X,Y, VALUES, CHVAL) |
ELSE |
| IOLD=0 |
LWST=0
C       PRINT *, 'INITIAL', IDLD, LWST
       CALL SRIPLR(IX, Y, VALUES, CHVAL)
ENDIF
RETURN
END
DUA0:(SRIICDALPLR.FOR;67, type? [N]:)
SUBROUTINE SRIPLR(MPX,MPY,VALUES,CHVAL)
C
MPX AND MPY ARE THE CORNER COORDINATES OF THE SELECTED AREA
REAL MPX(4),MPY(4),ANGSLP(4),SLP(4),INTRE(4),
$PX(4),PY(4)
CHARACTER*2 SI
CHARACTER*80 CHVAL(15)
REAL VALUES(10)
DIMENSION PINTX(300,50),PINTY(300,50)
1,PLRX(50),PLRYD(50),NID(4)
1,PLRX(300,4),PLRY(300,4),SXP(4),SYP(4),PLRX(50),PLRY(50)
DIMENSION CENSLP(7),CENFAWTH(7),CENPCLEN(7),CENCNS(7),
ICENCI(7),CENC2(7),CX(300,2),CY(300,2)
C
DIMENSION STATEMENT FOR STORING THE ENTRY WIDTHS
DIMENSION ENTRY(20)
C
THIS COMMON STATEMENT IS USED TO EXTRACT THE CORNER POINTS
FROM THE MAIN PROGRAM
COMMON /base/npoint,1,xa(300),ya(300),mpt(300),nflag(300),
1ndt(300),ndtt(300),x(2000),y(2000),mpt(2000),ip11
C
COMMON FOR OFFSETTING THE START FROM THE INITIAL START POINTS
COMMON /OFFSET/WIDTH12
C
DATA STATEMENT FOR STORING THE NUMBE OF ENTRIES
DATA IDNTY/L/
DATA PI/3.14159267/
INTEGER FUPANSI,IAARFLAG(4)
C
COMMON FOR FINING THE STATUS FROM Lw DEFINITION
C
ETHER A OLD FILE HAS TO BE OPENED OR THE NEW FILE HAS TO BE CREATED
COMMON /FILEST/10LD
C
COMMON AREA TO IDENTIFY WHETHER THE SUB IS CALLED FROM Lw DESIGN
COMMON /LW/LWST,ARE,IPD
CHARACTER ANS,redy,fname*10,ARE*6,FME*10,AREA*6,BORP,DUMB*4
COMMON /ifipat/icolor,ipattern,ifileg,lu
COMMON /PLR/PLFPI,PLFPA,PLFRS
C
DATA ifileg,lu/0,15/
C
THE BELOW DIMENSION IS USED FOR STORING THE VARIOUS CENTER LINES
C
FOR THE R/P SECTION
COMMON /CENIN/SLPZ,FAWTH,FCLEN,CONS,C1,C2,
ICENSLP,SLP,INTRE,CENFAWTH,CENPCLEN,
ICENCNS,CENC1,CENC2,OFFDIS,CX,CY,IPERFE,IBPERFE,ICROSS
DATA DUMB'/DAT'
C
THIS COMMON AREA IS DEFINED FOR WRITING ON TO THE DATA FILE
C
CHECKING FOR DRAWING THE OLD FILE
CALL PROMPT
PRINT *,'DESIGN BLEEDER OR DEVELOPMENT (BLEEDER=1 DEVEL=0)'
CALL YES_NO(IBORP)
C
READ(*,2)IBORP
C 2 FORMAT(A1)
    IF(IORD .EQ. 1)IORD='B'
    IF(IORD .EQ. 0)IORD='P'
    IF((IOLD .EQ. 1) .AND. (LWST .EQ. 0))THEN
        IF(IORD .EQ. 'P')THEN
            OPEN(UNIT=12,FILE='RP.DAT',STATUS='OLD')
        ELSE
            OPEN(UNIT=12,FILE='BP.DAT',STATUS='OLD')
        ENDIF
    CALL PROMPT
    IF(LWST .EQ. 0)THEN
        PRINT *, 'ENTER THE AREA NUMBER (AREA1...)
        READ(*,108)AREA
        PRINT *, 'ENTER THE R/P OR SLEEPER ID NUMBER'
        INP=999
        READ *,IPD
    ENDIF
    READ(12,505,END=600)AREA,IPN,FNAME
    IF((AREA .EQ. AREA) .AND. (IPD .EQ. IPN))THEN
        OPEN(UNIT=15,FILE=FNAME,STATUS='OLD',ERR=600)
        READ(15,*)MP1,MPY
        READ(15,*)OFFDIS
        READ(15,*)CENSLP
        READ(15,*)CENFANWDTH
        READ(15,*)CENPCLEN
        READ(15,*)CENC
        READ(15,*)CENC2
        READ(15,*)IPC
        DO 25 I=1,IPC-1
            READ(15,*)((PLRX(I1,I2),PLRY(I1,I2)),I2=1,4)
        25 CLOSE(UNIT=15)
        CLOSE(12)
        GOTO 561
    ELSE
        GOTO 24
    ENDIF
    IPATTERN=0
    C THIS PORTION ASKS THE USER FOR THE CORRECT SEQUENCE OF THE
    C POINTS ON THE POLYGON
    IF(LWST .EQ. 1)THEN
        CALL CHRSTZ(3)
        CALL MOVEA(MPX(1),MPY(1))
        CALL OUTN(1)
        CALL MOVEA(MPX(1),MPY(1))
        DO 525 IABC=2,4
        CALL DRAWA(MPX(IABC),MPY(IABC))
        CALL OUTN(IABC)
CONTINUE
CALL DRAWA(MPX(1),MPY(1))
CALL SELL
CALL PROMPT
PRINT *, '------------------------------------'
PRINT *, 'ENTER THE POINT SEQUENCE STARTING FROM'
PRINT *, 'LOWER LEFT HAND CORNER (CLOCK WISE)'
PRINT *, '------------------------------------'
PRINT *, '
C READ *,I1,I2,I3,I4
PRINT *, 'ENTER THE LOWER LEFT POINT'
CALL INPUT(SIN)
I1=GIN
PRINT *, 'ENTER THE UPPER LEFT POINT'
C READ *,I2
CALL INPUT(SIN)
I2=GIN
PRINT *, 'ENTER THE UPPER RIGHT POINT'
C READ *,I3
CALL INPUT(SIN)
I3=GIN
PRINT *, 'ENTER THE LOWER RIGHT POINT'
C READ *,I4
CALL INPUT(SIN)
I4=GIN
DO 522 IABC=1,4
   SXP(IABC)=MPX(IABC)
   SYP(IABC)=MPY(IABC)
522   MPX(1)=SXP(I1)
   MPY(1)=SYP(I1)
   MPX(2)=SXP(I2)
   MPY(2)=SYP(I2)
   MPX(3)=SXP(I3)
   MPY(3)=SYP(I3)
   MPX(4)=SXP(I4)
   MPY(4)=SYP(I4)
GOTO 531
ENDIF
   IF (IDOLD .NE. 1) THEN
   CONTINUE
C CHVAL(1)='ENTER THE OPTION FOR AREA IDENTIFICATION'
C CALL MAISCN(1,CHVAL)
C CHVAL(1)='(1) CORNERS BY CORNER NODES'
C CHVAL(2)='(2) CORNERS BY CURSOR'
C CHVAL(3)='(3) CORNERS BY EDGE AND LENGTH'
C CHVAL(4)='(4) CORNERS BY CENTER NODES'
C CHVAL(5)='(5) RETURN TO MAIN MENU'
C CHVAL(6)='ENTER YOUR OPTION'
C CALL OPTIONS(6,CHVAL,IOPT)
CALL AREADEF(MPX,MPY,2)

IF(MPX(1).EQ. -9999.)RETURN
ENDIF

CALL MOVEA(MPX(1),MPY(1))
CALL OUTM(1)
CALL MOVEA(MPX(1),MPY(1))
DO 31 ICKL=1,4
    CALL DRAWA(MPX(ICKL),MPY(ICKL))
    CALL OUTM(ICKL)
    CALL MOVEA(MPX(ICKL),MPY(ICKL))
31 CONTINUE
CALL DRAWA(MPX(1),MPY(1))
CALL TSEND
CALL PROMPT

print *, 'DO YOU WANT TO REDRAW THE SECTION (YES=1 NO=0) ?'
READ (*) ,IANS
IF(AANS .EQ. 'Y') GOTO 533
CALL YES_NO(AANS)
IF(AANS .EQ. 'N') GOTO 533

CALL RPWINDOW
CALL CHRSIZ(3)

CALCULATING THE SLOPES, INTERCEPTS OF THE LINES 1_2, 2_3, 3_4 AND 4_1

i flagship=0
I PERFECT=0
DO 10 I=1,4
    K=I+1
    IF (K .EQ. 5) K=1
    SLP(I)=SLOPE(MPX(I),MPY(I),MPX(K),MPY(K))
10 MAKING THE SLOPES TO BE WITHIN LIMITS OF SINGLE PRECISION

IF(ABS(SLP(I)) .GT. 99999.9) SLP(I)=999.
IF(ABS(SLP(I)) .LT. .00001) SLP(I)=.0001
PRINT *, 'SLP(I)', SLP(I)
IF(SLP(I).LT.0.) THEN
  SLP(I) = SLP(I)
  ANGSLP(I) = ATAND(SLP(I))
  SLP(I) = SLP(I)
ELSE
  ANGSLP(I) = ATAND(SLP(I))
ENDIF
IF((ABS(SLP(I)).GE.999.1.OR.(ABS(SLP(I)).LE..001))
  $IPERFECT$IPERFECT+1
10 CONTINUE
PRINT *, 'IPERFECT', IPERFECT
C TRANSFERRING IN CASE OF DRAWING OLD PILLARS
IF(OLD .EQ. 1) GOTO 121
C INITIALIZING FOR THE MAIN ROAD CENTER LINES
5 ICROSS=0
  IENTRY=1
C INITIALIZING FOR DRAWING THE CENTER LINES THROUGH SRICENTER ROUTINE
IND=0
C ENQUIRING FOR THE PANEL ORIENTATION W.R.T HORIZONTAL
C ENQUIRING FOR THE BASE LINE
C PRINT *, ' ENTER ONE OF THE FOLLOWING FOR THE MAIN ENTRANCE'
C PRINT *, ' ' 1 - BASE LINE 1_2'
C PRINT *, ' ' 2 - BASE LINE 2_3'
C PRINT *, ' ' 3 - BASE LINE 3_4'
C PRINT *, ' ' 4 - BASE LINE 4_1'
C READ *, IBASE
CHVAL(1)='CHOOSE THE MAIN ENTRANCE ORIENTATION'
CALL MAINSCN(1,CHVAL)
CHVAL(1)='<(1) LINE 1 - 2'
CHVAL(2)='<(2) LINE 2 - 3'
CHVAL(3)='<(3) LINE 3 - 4'
CHVAL(4)='<(4) LINE 4 - 1'
CHVAL(5)='<ENTER YOUR OPTION'
CALL OPTIONS(5,CHVAL,IBASE)
IF( ICROSS .EQ. 0) IPERFECT=IBASE
C CALL MENUSCN(4,CHVAL)
C CHVAL(1)='YES'
C CHVAL(2)='NO'
C CHVAL(3)='NO'
C CHVAL(4)='NO'
C CALL AVALSCN(4,CHVAL)
C IF(CHVAL(1) .EQ. 'YES') IBASE=1
C IF(CHVAL(2) .EQ. 'YES') IBASE=2
C IF(CHVAL(3) .EQ. 'YES') IBASE=3
C IF(CHVAL(4) .EQ. 'YES') IBASE=4
ANGLE=ATAN2(SLPAZ)
ELSE
CHVAL(1)="ENTER THE STARTING POINT FOR PILLARS"
CALL MAINSCN(1,CHVAL)
CHVAL(1)="<1> CORNER 1"
CHVAL(2)="<2> CORNER 2"
CHVAL(3)="<3> CORNER 3"
CHVAL(4)="<4> CORNER 4"
CHVAL(5)="ENTER YOUR OPTION"
CALL OPTIONS(5,CHVAL,IBASE)
CALL MENUSCN(4,CHVAL)
CHVAL(1)="YES"
CHVAL(2)="NO"
CHVAL(3)="NO"
CHVAL(4)="NO"
CALL AVALSCN(4,CHVAL)
IF(CHVAL(1) .EQ. 'YES') IBASE=1
IF(CHVAL(2) .EQ. 'YES') IBASE=2
IF(CHVAL(3) .EQ. 'YES') IBASE=3
IF(CHVAL(4) .EQ. 'YES') IBASE=4
PRINT *, 'ENTER THE STARTING POINT FOR THE PILLARS'
PRINT *, '-------------------------------'
PRINT *, ' 1 - STARTING POINT IS CORNER 1'
PRINT *, ' 2 - STARTING POINT IS CORNER 2'
PRINT *, ' 3 - STARTING POINT IS CORNER 3'
PRINT *, ' 4 - STARTING POINT IS CORNER 4'
PRINT *, '-------------------------------'
READ *,IBASE
C1=KNCTC(SLPAZ,MPX(IBASE),MPY(IBASE))
IF (ICROSS .EQ. 0) THEN
  IF (IBASE .EQ. 1) THEN
    C2=KNCTC(SLPAZ,MPX(3),MPY(3))
    C3=KNCTC(SLPAZ,MPX(4),MPY(4))
    DEFDIST IS THE DISTANCE USED IN CASE OF DEFAULT DESIGN
    DEFDIST=DIST(MPX(1),MPY(1),MPX(4),MPY(4))
    DEFDIST1=DIST(MPX(2),MPY(2),MPX(3),MPY(3))
  ENDIF
  IF (IBASE .EQ. 2) THEN
    C2=KNCTC(SLPAZ,MPX(1),MPY(1))
    C3=KNCTC(SLPAZ,MPX(4),MPY(4))
    DEFDIST=DIST(MPX(1),MPY(1),MPX(2),MPY(2))
    DEFDIST1=DIST(MPX(4),MPY(4),MPX(3),MPY(3))
  ENDIF
  IF (IBASE .EQ. 3) THEN
    C2=KNCTC(SLPAZ,MPX(1),MPY(1))
    C3=KNCTC(SLPAZ,MPX(2),MPY(2))
    DEFDIST=DIST(MPX(1),MPY(1),MPX(4),MPY(4))
    DEFDIST1=DIST(MPX(2),MPY(2),MPX(3),MPY(3))
  ENDIF
ENDIF
IF(IBASE .EQ. 4) THEN
  C2=XTNTCT(SLPAZ,MPX(3),MPY(3))
  C3=XTNTCT(SLPAZ,MPX(4),MPY(4))
  DEFDIST=DIST(MPX(1),MPY(1),MPX(2),MPY(2))
  DEFDIST1=DIST(MPX(4),MPY(4),MPX(3),MPY(3))
ENDIF
IF(DEFDIST1 .LT. DEFDIST) DEFDIST=DEFDIST1
ELSE
IF (IBASE .EQ. 1) THEN
  C2=XTNTCT(SLPAZ,MPX(3),MPY(3))
  C3=XTNTCT(SLPAZ,MPX(4),MPY(4))
  C4=XTNTCT(SLPAZ,MPX(2),MPY(2))
  IF(ABS(C1-C4) .GT. ABS(C1-C3)) C3=C4
ENDIF
IF (IBASE .EQ. 2) THEN
  C2=XTNTCT(SLPAZ,MPX(3),MPY(3))
  C3=XTNTCT(SLPAZ,MPX(4),MPY(4))
  C4=XTNTCT(SLPAZ,MPX(1),MPY(1))
  IF(ABS(C1-C4) .GT. ABS(C1-C3)) C3=C4
ENDIF
IF (IBASE .EQ. 3) THEN
  C2=XTNTCT(SLPAZ,MPX(1),MPY(1))
  C3=XTNTCT(SLPAZ,MPX(2),MPY(2))
  C4=XTNTCT(SLPAZ,MPX(4),MPY(4))
  IF(ABS(C1-C4) .GT. ABS(C1-C3)) C3=C4
ENDIF
IF(IBASE .EQ. 4) THEN
  C2=XTNTCT(SLPAZ,MPX(2),MPY(2))
  C3=XTNTCT(SLPAZ,MPX(1),MPY(1))
  C4=XTNTCT(SLPAZ,MPX(3),MPY(3))
  IF(ABS(C1-C4) .GT. ABS(C1-C3)) C3=C4
ENDIF
ENDIF
C WRITE(*,35)
C 35 FORMAT(1X,' ENTER THE PANEL ORIENTATION ANGLE ----> ',A)
C READ *,AZANG
CC MAKING THE ANGLE ABSOLUTE
C AZANG=ANGLSP(IBASE)-AZANG
C IF(ABS(AZANG)) .GE. 90. THEN
C IF(AZANG .LT. 0.) THEN
C  AZANG=180.+AZANG
C ELSE
C  AZANG=180.-AZANG
C ENDIF
C ENDIF
C PRINT *,AZANG,AZANG
C SLPAZ=TANH(AZANG)
C CALCULATING THE STARTING AND ENDING LINES
C C2=MPY(ISTART)-SLPAZ*MPX(ISTART)
C C1=MPY(IEND)-MLPAZ+MPX(IEND)
C
C INQUIRING FOR THE FACE WIDTH FOR THE LONGWALL AREA
C ADJUSTING FOR THE MAXIMUM DISTANCE
C IF(ABS(C1-C3) .LE. ABS(C1-C2))C2=C3
C IF(I CROSS .EQ. 0) THEN
C CHVAL(1)=DESIGN CRITERIA'
C CALL MAINSCN(1,CHVAL)
C CHVAL(1)="(1) BASED ON ENTRIES'
C CHVAL(2)="(2) BASED ON DIMENSIONS'
C CHVAL(3)="(3) RETURN TO MAIN MENU'
C CHVAL(4)=" ENTER YOUR OPTION'
C CALL OPTIONS(4,CHVAL,IDESIGN)
C IF(IDESIGN .EQ. 3) RETURN
C
C CALL MENUSCN(2,CHVAL)
C CHVAL(1)="YES'
C CHVAL(2)="NO'
C CHVAL(3)="NO'
C CALL AVALSCh(2,CHVAL)
C IF(CHVAL(1) .EQ. 'YES')IDESIGN=1
C IF(CHVAL(2) .EQ. 'YES')IDESIGN=2
C
C CALL SETUP(1)
C CALL ERASESCRN(2)
C CALL DRAWMOVE(10)
C CALL SETUP(0)
C PRINT *, ' CHOOSE ONE FROM THE FOLLOWING'
C PRINT *, ' -----------------------------'
C PRINT *, ' 1 - DESIGN BASED ON NUMBER OF ENTRIES'
C PRINT *, ' 2 - DESIGN BASED ON SPECIFIED DIMENSIONS'
C PRINT *, ' -----------------------------'
C READ *, IDESIGN
C WRITE(*,30)
C 30 FORMAT(1X,' ENTER THE MAIN ENTRANCE WIDTH -----> ',$)
C READ *,FAW
C CALL PROMPT
C PRINT *, ' WIDTH OF THIS SECTION IS ----->',SFDIST
C PRINT *, ' PRESS RETURN TO CONTINUE'
C READ (*,30)CHVAL(1)
C 30 FORMAT(A1)
C VALUES(1)=120.
C VALUES(2)=60.
C VALUES(3)=65.
C VALUES(4)=0.
C CALL RFWND
C CHVAL(1)=DESIGN PARAMETERS'
C CALL MAINSCN(1,CHVAL)
C CALL DESMENU
C CHVAL(1)="[A] PILLAR LENGTH'
C CHVAL(2)="[B] CROSS CUT'
C CHVAL(3)="[C] PILLAR WIDTH"
CHVAL(4)='OFFSET DISTANCE'
CALL MENUSCN(4,CHVAL,35)
CHVAL(1)='120.00'
CHVAL(2)='20.09'
CHVAL(3)='65.00'
CHVAL(4)='0.00'
CALL MENUSCN(4,CHVAL,20)
IRGN=7
DO 1JK=1,4
   CALL BLANKS(1,IRGN,55,10)
   IRGW=IRGW+2
END DO
CALL DVALSCN(4,CHVAL,VALUES)
PMLEN=VALUES(1)
PWDTCH=VALUES(2)
PCLCH=VALUES(3)
POFDF=VALUES(4)
CHVAL(1)='PILLAR & ROOM WIDTH VARIATION DESIGN'
CALL MAINTCN(1,CHVAL)
CHVAL(1)='<1> NO VARIATION ACROSS ROWS'
CHVAL(2)='<2> VARIATION ACROSS ROWS'
CHVAL(3)='<3> RETURN TO MAIN MENU'
CHVAL(4)='< ENTER YOUR OPTION'
CALL OPTIONS(4,CHVAL,IVAR)
IF(IVAR .EQ. 3)RETURN
C   PRINT *, 'ENTER THE PILLAR LENGTH ALONG THE MAIN'
C   READ *,PMLEN
C   PRINT *, 'ENTER THE CROSS ROAD WIDTH'
C   READ *,PWDTCH
C   PRINT *, 'ENTER THE PILLAR LENGTH ALONG CROSS'
C   READ *,PCLCH
C   PRINT *, 'ENTER THE OFFSET DISTANCE BETWEEN TWO ROWS'
C   READ *,POFDF
C   MAKING THE POFDF VERY SMALL IN CASE OF ZERO INPUT
C   IF(POFDF .EQ. 0.)POFDF=2.
C   CALL PROMPT
C   IF (IDEIGN .EQ. 1)THEN
   PRINT *, 'ENTER NUMBER OF ENTRIES'
   READ *,NENTRIES
   CALL INPUT(GIN)
   NENTRIES=GIN
   FAWDTCH= ((DEFDIST-(FLOAT(NENTRIES-1)*PCLCH))/FLOAT(NENTRIES))
   PRINT *, 'ENTRY WIDTH IS ',FAWDTCH
   ENTRY(1ENTRY)=ABS(FAWDTCH)
   ELSE
   PRINT *, 'ENTER THE FIRST ENTRY WIDTH'
   READ *,FAWDTCH
   CALL INPUT(GIN)
   FAWDTCH=GIN
   ENTRY(1ENTRY)=ABS(FAWDTCH)
ICOUNT DETERMINES WHICH IS THE FIRST POINT OF THE PANEL ETC...
PXX, PNY STORE THE INDIVIDUAL PANELS CORNER POINTS

IF THE ENDING LINE SLOPED IS LESS THAN THE BEGINING LINE
THE WIDTHS ARE MADE NEGATIVE

IF (C2 .LT. C1) THEN
FAWIDTH=FAWIDTH
FLCLEN=FLCLEN
IF(ICROSS .EQ. 1)PPOFD=-PPOFD
ENDIF
I=1
IOF=1
CONS=ABS(COSD(ANGLE))
IF(CONS .EQ. 0.)CONS=.0001
WIDTH=FAWIDTH/(CONS)
OFFDIS=PPOFD/(CONS)

PRINT *, ' DO YOU WANT THE INITIAL OFFSET FOR CROSS ROADS (Y/N)'
READ (*.9Y)ANS
C 89 FORMAT(AI)
C IF(ANS
C MAKING DFAC SMALL TO GET THE STARTING CROSS ROAD OFFSET
C ASKING FOR A OFFSET DISTANCE FROM THE BEGINING
IF(ICROSS .EQ. 1) THEN
PRINT *, ' ENTER THE OFFSET DISTANCE FROM THE CORNER'
READ *,DFACT
CALL INPUT(GIN)
DFACT=GIN
WIDTH=DFACT*(WIDTH/ABS(FAWIDTH))
WIDTH12=WIDTH
END IF
IMAIN=1
IPLFLG=0
IND=IND+1
C CALLING FOR CENTER LINES
CALL CENINFO(IND)
IND=IND+1
CALL CENINFO(IND)
C CALLING PILLAR CENTER LINES
CALL CENINFO(6)
IF(ICROSS .EQ. 1) THEN
CALL CENINFO(5)
CALL SRICENTER(S,MPX,MPY)
CALL CENINFO(7)
ENDIF
CALL TSEND
77 CONTINUE
ic=i-6
ICOUNT=1
RIJK=I
DO 50 ICK=1,4
   IFLAG1=0
   IFLAG2=0
   IPCK=ICK+1
   IF(IPCK .EQ. 5) THEN
      IPCK=1
   ENDF
C TAKING CARE OF PERFECT FIGURES
IF(IPERFECT .EQ. 4) THEN
   PRINT *, ' IN THE PERFECT LOOP'
IF(ICROSS .EQ. 0) THEN
   IF(((BASE.EQ.1).OR.((BASE.EQ.3))) THEN
      PRINT *, ' BASE LINE 1 OR 3'
      PY(1)=MPY(1)
      PY(2)=MPY(2)
      PX(1)=MPX(1)+ABS(WIDTH)*CONS
      IF(PX(1) .GE. MPX(4)) GO TO 200
      PX(2)=PX(1)
      ELSE
      PRINT *, ' BASE LINE 2 OR 4'
      PX(1)=MPX(2)
      PX(2)=MPX(3)
      PY(1)=MPY(1)+ABS(WIDTH)*CONS
      IF(PY(1) .GE. MPY(2)) GO TO 200
      PY(2)=PY(1)
   ENDF
   ELSE
   IF(((IPERFECT.EQ.2).OR.((IPERFECT.EQ.4))) THEN
      PRINT *, ' BASE LINE 1 OR 3'
      PY(1)=MPY(1)
      PY(2)=MPY(2)
      PX(1)=MPX(1)+ABS(WIDTH)*CONS
      IF(PX(1) .GE. MPX(4)) GO TO 200
      PX(2)=PX(1)
      ELSE
      PRINT *, ' BASE LINE 2 OR 4'
      PX(1)=MPX(1)
      PX(2)=MPX(3)
      PY(1)=MPY(1)+ABS(WIDTH)*CONS
      IF(PY(1) .GE. MPY(2)) GO TO 200
      PY(2)=PY(1)
   ENDF
   ENDF
ICOUNT=3
GO TO 79
ENDIF
if(iick .eq. 1) then
  if((yi .ge. mpy(ick)).and.(yi .le. mpy(ipck))) iflagl=1
endif
if(iick .eq. 2) then
  IF((xi .GE. MPx(ick)).AND.(xi.LE.MPx(ipck))) IFLAG1=1
endif
if(iick .eq. 3) then
  if((yi .le. mpy(ick)).and.(yi.ge.ipck))) iflagl=1
endif
if(iick .eq. 4) then
  if((xi .le. mpx(ick)).and.(xi.ge.ipck))) iflagl=1
endif
IF((IFLAG1 .EQ. 1)) THEN
  ITRACK=ITRACK+1
  IARFLAG(I_COUNT)=ICK
  PX(I_COUNT)=XI
  PY(I_COUNT)=YI
  I_COUNT=ICOUNT+1
  CONTINUE
endif
C
C DRAWING THE LINES WITH THE ABOVE COORDINATES
C
C PRINT *, "PX(1),PY(1),PX(2),PY(2),PX(1),PY(1),PX(2),PY(2)
C CALL MOVEA(PX(1),PY(1))
C CALL DRAWA(PX(2),PY(2))
call tsend
IF(ICROSS .EQ. 0) THEN
  PLRX(I)=PX(1)
  PLRY(I)=PY(1)
  PLRXD(I)=PX(2)
  PLRYD(I)=PY(2)
C
C STORING THESE COORDINATES FOR LATER USE
I=I+1
C
C CHECKING FOR THE DIMENSION
C
C**************************************************************************
C**************************************************************************
if(1 .ge. 50) then
  print *, ' reduce the section length'
  return
endif
C**************************************************************************
C**************************************************************************
ENDIF
IF(ICROSS .EQ. 1) THEN
  W1=0.
  IGF=1
  DXX=PX(1)
  DYY=PY(1)
  DXX1=PX(2)
  DYY1=PY(2)
IF (IGF .GE. 3) THEN
  PX(1)=CX(I,1)
  PY(1)=CY(I,1)
  PX(2)=CX(I,2)
  PY(2)=CY(I,2)
ENDIF

IF (IGF .EQ. 4) IGF=0
ELSE
  PX(1)=DXX
  PY(1)=DYY
  PX(2)=DXX1
  PY(2)=DYY1
ENDIF

IF (IPERFECT .EQ. 4) THEN
  PRINT *, 'IN THE PERFECT LOOP AFTER MAIN'
ENDIF

IF (IPERFECT .EQ. 1) OR (IPERFECT .EQ. 3) THEN
  PRINT *, 'BASE LINE IS 2 OR 4'
  PRINT *, '****PLRX(J1)', PLRX(J1)
  XP=PLRX(J1)
  YP=PY(1)
ELSE
  PRINT *, 'BASE LINE IS 1 OR 3'
  PRINT *, '****PX(1)', PX(1)
  XP=PX(1)
  YP=PLRX(J1)
ENDIF

PRINT *, '-----------------------'
PRINT *, 'ICROSS XP,YP', XP, YP
GO TO 2222
END IF

SLOPE1=SLOPE(PLRX(J1), PLRY(J1), PLRXD(J1), PLRYD(J1))
SLOPE2=SLOPE(PX(1), PY(1), PX(2), PY(2))

IF (ABS(SLOPE1) .LE. .1) THEN
  XP=PX(1)
  YP=PLRY(J1)
ENDIF

PRINT *, 'PERPENDICULAR XP,YP', XP, YP
GO TO 2222

CINT1=XTCT(SLOPE1, PLRX(J1), PLRY(J1))
CINT2=XTCT(SLOPE2, PX(1), PY(1))
CALL COORD(SLOPE1, SLOPE2, WI, CINT1, CINT2, XP, YP)

DIST1=SQRT((PX(1)-PX(1)**2+(PY(1)-PY(1)**2)
DIST2=SQRT((PX(2)-XP)**2+(PY(2)-YP)**2)
DIST3=SQRT((PX(1)-XP)**2+(PY(1)-YP)**2)

IF (DIST2 .LT. DIST1) .AND. (DIST3 .LT. DIST1) THEN
  PINTX(I,J1)=XP
  PINTY(I,J1)=YP
ELSE
  PINTX(I,J1)=0.
  PINTY(I,J1)=0.
ENDIF
55 CONTINUE
   I=I+1
C CHECKING FOR THE DIMENSION
C******************************************************************************
   IF(I .GE. 300) THEN
      PRINT *, 'REDUCE THE SECTION LENGTH'
      RETURN
   ENDIF
C******************************************************************************
   ENDIF
   GOTO 45
ENDIF
ENDIF
50 CONTINUE
C CHECKING FOR THE INITIAL PIECE
45 PX(3)=PX(2)
   PY(3)=PY(2)
   PX(4)=PX(1)
   PY(4)=PY(1)
   IARPRE=IARFLAG(1)
   IARPRE1=IARFLAG(2)
   IF(WIDTH .NE. 0.) THEN
      IF(ITRACK .EQ. 2) THEN
         A=ABS(C1-C2)
      ENDIF
   ENDIF
C CALCULATING THE PANEL DISTANCE FROM THE CORNER
C IF(IMAIN .EQ. 1) THEN
   IMAIN=0
C TAKING INTO ACCOUNT THE CHANGES IN WIDTH ACROSS ROWS
C IF(IVAR .EQ. 2 .AND. ICROSS .EQ. 0) THEN
   CALL BELL
   CALL PROMPT
   IVARROW=1/2
   WRITE(*,51) IVARROW
C FORMAT(1X,'ENTER THE WIDTH OF PILLARS FOR ROW ','I2','--')
51 FORMAT(1X,'ENTER THE WIDTH OF PILLARS FOR ROW ','I2','--')
C READ *, WIDTH
   CALL INPUTT(GIN)
   WIDTH=GIN
   IF(PCLEN .LT. 0) WIDTH=-WIDTH
   WIDTH=WIDTH+(WIDTH/CONS)
   ELSE
   WIDTH=PCLEN
   WIDTH=WIDTH+(PCLEN/CONS)
   ENDIF
   ELSE
   IMAIN=1
   IF(IVAR .EQ. 2 .AND. ICROSS .EQ. 0) THEN
CALL BELL
CALL PROMPT
IVARROW=I/2+1
WRITE(*,52)IVARROW
CALL INPUT(GIN)
VWIDTH=GIN
IF(FAWIDTH .LT. 0)VWIDTH=-VWIDTH
WIDTH=WIDTH+(VWIDTH/CON5)
ELSE
WIDTH=WIDTH+(FAWIDTH/CON5)
VWIDTH=FAWIDTH
ENDIF
ENDIF
IF((ABS(C2-C1)) .LT.(ABS(WIDTH)))GOTO 200
IF((ICROSS .EQ. 0).AND.(IMAIN .EQ. 0))THEN
ENTRY(ENTRY)=ABS(VWIDTH)
ENTRY=ENTRY+1
ENDIF
GO TO 77
200 CONTINUE
IF((ICROSS .EQ. 0))THEN
ANGLE=ATAND(SLPAZ)
ANGLE=ANGLE+69.9999
PRINT *, 'ENTER VALUES FOR THE CROSS ROADS'
PRINT *, ---------------------'
SLPAZ=TAND(ANGLE)
ICROSS=1
FAWIDTH=PWDTH
PCLEN=PMLEN
IMCOUNT=I-1
PRINT *, '
GOTO 15
ENDIF
IMCOUNT=I-1
C FILLING THE PILLARS WITH PATTERN -E
IPATTERN=0
IPC=1
DO 115 JI=1,IMCOUNT,2
IF((JI) .GE. IMCOUNT)GOTO 121
IK=1
119 IF((IK) .GE. IMCOUNT)GOTO 115
SKP(1)=PINTX(IK,JI)
SYP(1)=PINTY(IK,JI)
SKP(2)=PINTX(IK+1,JI)
SYP(2)=PINTY(IK+1,JI)
SKP(3)=PINTX(IK+1,JI+1)
SYP(3)=PINTY(IK+1,JI+1)
SKP(4)=PINTX(IK,JI+1)
DO 117 JIII=1,4
   IF((SXP(JIII) .EQ. 0.) .AND. (SYP(JIII) .EQ. 0.)) THEN
      IX=IX+2
      GOTO 119
   ENDIF
117 CONTINUE
   IX=IX+2
   DO 126 ICK=1,4
      PLRX(IPC,ICK)=SXP(ICK)
      PLRY(IPC,ICK)=SYP(ICK)
   END DO
   IPC=IPC+1
   GOTO 119
115 CONTINUE
121 CONTINUE
C
   DRAWING THE PILLARS
   CALL TXTCOL(2)
   CALL PROMPT
   PRINT *, 'PILLAR CENTER LINES (YES=1 NO=0)'
   READ(*,104)ANS
   CALL YES_NO(ANS)
   CALL SRICENTER(1,MPX,MPY)
   IF(ANS .EQ. 1) THEN
      IF(IDOLD .NE. 1) CALL SRICENTER(2,MPX,MPY)
      IF(IDOLD .NE. 1) CALL SRICENTER(3,MPX,MPY)
      CALL TSEND
   ENDIF
   IDYES=0
   CALL PROMPT
   PRINT *, 'ID NUMBERS FOR PILLARS (YES=1 NO=0)'
   READ(*,104)ANS
C
   IF(ANS .EQ. 'Y') IDYES=1
   CALL YES_NO(ANS)
   IF(ANS .EQ. 1) IDYES=1
   IPATTERN=-3
   CALL CHRSIZ(4)
   DO 124 ICK=1,IPC-1
      DO 126 IDK=1,4
         SXP(IDK)=PLRX(ICK,IDK)
         SYP(IDK)=PLRY(ICK,IDK)
      IF(ICK .EQ. 1) THEN
         CALL MOVEA(SXP(IDK),SYP(IDK))
         CALL OUTK(IDK)
         CALL MOVEA(SXP(IDK),SYP(IDK))
      ENDIF
      124 CONTINUE
      CALL PATTERN(4,SXP,SYP)
   IF(IDYES .EQ. 1) THEN
CHY=(SYP(1)+SYP(2)+SYP(3)+Str(4))/4.
CALL MOVEA(CHY,CHY)
CALL OUTN(IK)
CALL MOVEA(SXP(1),SYP(1))
ENDIF
124 CONTINUE
C SECTION TO MODIFY THE PILLAR GEOMETRY
156 CONTINUE
CHVAL(1)='PILLAR MODIFICATION MENU'
CALL MAINSCN1,CHVAL)
CHVAL(1)='1' CHANGE GEOMETRY'
CHVAL(2)='2' REDESIGN/RETURN'
CHVAL(3)='3' NEW PILLAR/SURVEYOR INPUT'
CHVAL(4)='4' PLOT THE PILLARS'
CHVAL(5)='ENTER YOUR OPTION'
CALL OPTIONS(5,CHVAL,IPCH)
C CHVAL(1)='YES'
C CHVAL(2)='NO'
C CHVAL(3)='NO'
CALL AVALSCH(3,CHVAL)
C IF(CHVAL(1) .EQ. 'YES')IPCH=1
C IF(CHVAL(2) .EQ. 'YES')IPCH=2
C IF(CHVAL(3) .EQ. 'YES')IPCH=3
CZZZZ
PRINT *,
C PRINT *, PILLAR MODIFICATIONS'
C PRINT *,-----------------------
C PRINT *, 1 - CHANGE THE GEOMETRY'
C PRINT *, 2 - SAVE/REDESIGN'
C PRINT *, 3 - NEW PILLAR/CHANGE OLD PILLAR'
C READ *,IPCH
IF(IPCH .EQ. 4)GO TO 121
IF(IPCH .EQ. 3)THEN
PRINT *, 'PLEASE WAIT'','
CALL SRICENTER(6,MPX,MPY)
CALL SRICENTER(7,MPX,MPY)
CHVAL(1)='NEW PILLAR/SURVEYOR UPDATE'
CALL MAINSCH1,CHVAL)
CHVAL(1)='1' DRAW A NEW PILLAR'
CHVAL(2)='2' SURVEYOR INPUT'
CHVAL(3)='ENTER YOUR OPTION'
CALL OPTIONS(3,CHVAL,idesign)
IF(idesign .EQ. 1)THEN
CALL NEWPILLAR(SXP,SYP)
IPN=IPN
IPC=IPC+1
END IF
IF(idesign .EQ. 2)THEN
CHVAL(1)='SURVEYOR UPDATE'
CALL MAINSCN1,CHVAL)
CHVAL(3) = 'ENTER YOUR OPTION'
CALL OPTIONS(3,CHVAL,IDESGN1)
IF (IDESGN1 .EQ. 1) THEN
     PRINT *, 'ENTER THE PILLAR ID - FOR UPDATING'
     READ *, IPN
     CALL INPUT (GIN)
     IPN = GIN
     CALL NEWPILLAR (SXP, SYP)
     ELSE
     PRINT *, 'NOT AVAILABLE NOW'
     GO TO 156
     END IF
     END IF
     DO 152 ILK = 1, 4
     PLRX(IPN, ILK) = SIPI(ILK)
     152
     PLRY(IPN, ILK) = SYPI(ILK)
     GO TO 156
     ENDIF
     IF (IPCH .EQ. 1) THEN
     CHVAL(1) = 'CHOOSE THE LINE TO BE CHANGED'
     CALL MAINSCN(1, CHVAL)
     CHVAL(1) = '<1> LINE 1 - 2'
     CHVAL(2) = '<2> LINE 2 - 3'
     CHVAL(3) = '<3> LINE 3 - 4'
     CHVAL(4) = '<4> LINE 4 - 1'
     CHVAL(5) = 'ENTER YOUR OPTION'
     CALL OPTIONS(5, CHVAL, ILINE)
     C
     CHVAL(1) = 'YES'
     CHVAL(2) = 'NO'
     C
     CHVAL(3) = 'NO'
     C
     CHVAL(4) = 'NO'
     C
     CALL AVALSCN(4, CHVAL)
     C
     IF (CHVAL(1) .EQ. 'YES') ILINE = 1
     IF (CHVAL(2) .EQ. 'YES') ILINE = 2
     IF (CHVAL(3) .EQ. 'YES') ILINE = 3
     IF (CHVAL(4) .EQ. 'YES') ILINE = 4
     WD = 0.
     C
     PRINT *, 'CHOOSE ONE FROM BELOW'
     PRINT *, '------------------------'
     C
     PRINT *, '1 - CHANGE LINE 1_2'
     C
     PRINT *, '2 - CHANGE LINE 2_3'
     C
     PRINT *, '3 - CHANGE LINE 3_4'
     C
     PRINT *, '4 - CHANGE LINE 4_1'
     C
     READ*, ILINE
     C
     CHVAL(1) = 'CHOOSE THE PIVOT POINT'
     CALL MAINSCN(1, CHVAL)
     IF (ILINE .EQ. 1) THEN
     CHVAL(1) = '<1> ABOUT POINT 1'
```c
CHVAL(1)='YES'
CHVAL(2)='NO'
CALL AVALSCN(2,CHVAL)
IF(CHVAL(1) .EQ. 'YES') IPV=3
IF(CHVAL(2) .EQ. 'YES') IPV=4
PRINT *,' ILINE 3 SELECTED'
PRINT *,' 3 - ABOUT PT 3'
PRINT *,' 4 - ABOUT PT 4'
READ*, IPV
IF(IPV .EQ. 3) THEN
PRINT *,' ABOUT PT 3'
ICR1=4
ICR2=1
ELSE
PRINT *,' ABOUT PT 4'
ICR1=3
ICR2=2
ENDIF
ENDIF
IF (ILINE .EQ. 4) THEN
CHVAL(1)='(1) ABOUT POINT 4'
CHVAL(2)='(2) ABOUT POINT 1'
CHVAL(3)='ENTER YOUR OPTION'
CALL OPTIONS(3,CHVAL,IPV)
IF(IPV .EQ. 1) IPV=4
IF(IPV .EQ. 2) IPV=1
CALL MENUSCN(2,CHVAL)
CHVAL(1)='YES'
CHVAL(2)='NO'
CALL AVALSCN(2,CHVAL)
IF(CHVAL(1) .EQ. 'YES') IPV=4
IF(CHVAL(2) .EQ. 'YES') IPV=1
PRINT *,' 4 - ABOUT PT 4'
PRINT *,' 1 - ABOUT PT 1'
READ*, IPV
IF(IPV .EQ. 4) THEN
ICR1=1
ICR2=2
ELSE
ICR1=4
ICR2=3
ENDIF
ENDIF
CALL PROMPT
PRINT *,' ENTER THE OFFSET ANGLE'
READ *, OFANGLE
CALL INPUTT(SIN)
ofangle=SIN
BLPi=SLOPE(PLRX(1,IPV),PLRY(1,IPV),PLRX(1,ICR1),...
SLP11=TAND(ANG1)
CHVAL(1)="CHOOSE ONE OF THE OPTIONS"
CALL MAINSCN(1,CHVAL)
CHVAL(1)="<1> CHANGE ALL PILLARS"
CHVAL(2)="<2> CHANGE ONE PILLAR"
CHVAL(3)="<3> CHANGE A SET OF PILLARS"
CHVAL(4)="ENTER YOUR OPTION"
CALL OPTIONS(4,CHVAL,IPCH)
CALL MENUSCN(2,CHVAL)
CHVAL(1)="YES"
CHVAL(2)="NO"
CALL AVLSCN(2,CHVAL)
IF(CHVAL(1) .EQ. 'YES')IPCH=1
IF(CHVAL(2) .EQ. 'YES')IPCH=2
PRINT *, 'PLEASE CHOOSE ONE FROM THE FOLLOWING'
PRINT *, '-------------------------------------'
PRINT *, '
PRINT *, 1 - CHANGE ALL PILLARS'
PRINT *, 2 - CHANGE ONE PILLAR'
READ *,IPCH
CALL PROMPT
IF(IPCH .EQ. 1 .OR. IPCH .EQ. 2)THEN
  ISTART=1
  IEND=IPCH-1
END IF
IF (IPCH .EQ. 2)THEN
  PRINT *, 'ENTER THE ID OF THE PILLAR TO BE CHANGED'
  CALL INPUT(DUMY)
  IPID=DUMY
ENDIF
IF(IPCH .EQ. 3)THEN
  PRINT *, 'ENTER THE STARTING PILLAR ID'
  CALL INPUT(DUMY)
  ISTART=DUMY
  PRINT *, 'ENTER THE ENDING PILLAR ID'
  CALL INPUT(DUMY)
  IEND=DUMY
END IF
DO 150 ICK=1,IPCH-1
DO 150 ICK=ISTART,IEND
IF((IPCH .EQ. 2) .AND. (ICK .NE. IPID))GOTO 150
CP1=XTNCT(SLP11,PLRX(ICK,IPY),PLRY(ICK,IPY))
SLP12=SLOPE(PLRX(ICK,ICR1),PLRY(ICK,ICR1),
  IPRX(ICK,ICR2),PLRY(ICK,ICR2))
CP2=XTNCT(SLP12,PLRX(ICK,ICR2),PLRY(ICK,ICR2))
CALL COORD(SLP11,SLP12,WD,CP1,CP2,X1,Y1)
PLRX(ICK,ICR1)=X1
150 CONTINUE
   IF(PATTERN=0)
   CALL PATTERN(4,MPX,MPY)
   GOTO 121
ENDIF
C PART FOR WRITING ON TO A DATA FILE
C CALL PROMPT
C PRINT *,' DO YOU WANT TO STORE THIS INFORMATION (YES=1 NO=0)?
C READ (*,104)ANS
C 104 FORMAT(A1)
   CALL YES_NO(ANS)
   IF(ANS .EQ. 'I') THEN
C OPENING THE DIRECTORY FILE FOR STORING THE NEW FILE
   IF(ILD .EQ. 0) THEN
     IF(BORP .EQ. 'P') THEN
       OPEN(UNIT=12,FILE='RP.DAT',STATUS='UNKNOWN',
            ACCESS='APPEND',ERR=600)
     ELSE
       OPEN(UNIT=12,FILE='BP.DAT',STATUS='UNKNOWN',
            ACCESS='APPEND',ERR=600)
     ENDIF
   ENDIF
C READING THE FILE TO THE END IF OLD FILE
C CALL PROMPT
115 CONTINUE
   IF(LWST .EQ. 0) THEN
     PRINT *,' ENTER THE AREA NAME (AREA1..)'
     READ (*,108)ARE
108 FORMAT(A6)
     PRINT *,' ENTER THE R/P OR BLEEDER ID NUMBER'
C READ *,'IPD
   CALL INPUT(A15)
   IPD=GIN
ENDIF
C CONSTRUCTING A FILE NAME
   FNAME(1:2)=ARE(1:2)
   FNAME(3:4)=BORP(1:1)
   ENCODE(2,102,S1:IPD)
102 FORMAT(I2)
   FNAME(5:6)=S1
   FNAME(7:10)=DUMB(1:4)
   WRITE(*,106)FNAME
106 FORMAT(A10)
   WRITE(12,505)ARE,IPD,FNAME
505 FORMAT(A6,13,A10)
   CLOSE(12)
ENDIF
C PRINT *,' WRITING DATA ON TO THE FILE'
C OPEN(UNIT=15,FILE=FNAME,STATUS='NEW')
   WRITE(15,*)MPX,MPY
WRITE(*,*)CENFANDTH
WRITE(*,*)CENPCLEN
WRITE(*,*)CENCCONS
WRITE(*,*)CENCI
WRITE(*,*)CENC2
WRITE(*,*)IPC
DO 20 I1=1,IPC-1
20 WRITE(*,*)(PLRRX(I1,I2),PLRAY(I1,I2)),I2=1,4

call prompt
PRINT *, 'INFORMATION FOR SIMULATION'
PRINT *, 'ENTER THE PILLAR LENGTH FOR SIMULATION'
C READ *, PMLEN
CALL INPUTIN(GIN)
PMLEN=GIN
C WRITE(*,*)ENTRY,(ENTRY(IEN),IEN=1,IENTRY),PMLEN,PWDTH
C CHANGING THE DIMENSION INTO ENTRIES
ENTRY=0.
DO IEN=1,IENTRY-1
ENTRY=ENTRY+ENTRY(IEN)
END DO
AENTRY=(DEFDIST-ENTRY)/(IENTRY)
C PRINT *, ' AFTER TRANSFORMATION'
C WRITE(*,*)ENTRY,AENTRY,PMLEN,PWDTH
WRITE(*,*)ENTRY,AENTRY,PMLEN,PWDTH
CLOSE(I5)
C ENDIF
C CREATING A DIRECTORY FOR THE PILLAR AND BLEEDERS

IF(DORP .EQ. 'P') THEN
  OPEN(UNIT=4,STATUS='UNKNOWN',ACCESS='KEYED',
  ORGANIZATION='INDEXED',
  FORM='FORMATTED',KEY=(1:6:CHARACTER),FILE='PILDIREC',
  RECL=132)
ELSE
  OPEN(UNIT=4,STATUS='UNKNOWN',ACCESS='KEYED',
  ORGANIZATION='INDEXED',
  FORM='FORMATTED',KEY=(1:6:CHARACTER),FILE='BLEDIREC',
  RECL=132)
END IF
READ(4,700,KEY=FNAME(1:6),ERR=750)
C CHVAL(I1),MPI,MPY,IENTRY,AENTRY,PMLEN,PWDTH
700 FORMAT(A6,8F10.2,I2,3F7.2)
PRINT *, ' THIS FILE NAME ALREADY EXISTS'
PRINT *, ' DO YOU WANT TO OVER WRITE THIS FILE (YES=1 NO=0)?'
CALL YES_NO(IANS)
IF(IANS .EQ. 1) THEN
  REWRITE(4,700)FNAME(1:6),MPI,MPY,IENTRY,AENTRY,PMLEN,PWDTH

GO TO 110
END IF
GO TO 610
WRITE(4,700)FNAME(1:6),MPX,MPY,ENTRY,AVENTRY,PLEN,FWIDTH
END IF
CALL PROMPT
GOTO 610
600 PRINT *," ERROR IN FILE"
CLOSE(15)
CLOSE(12)
GOTO 612
610 PRINT *," DO YOU WANT TO REDESIGN THE R/P SECTION (YES=1 NO=0)"
READ (*,104)IANS
WRITE(*,*)'ENTRY',ENTRY,(ENTRY(IEN),IEN=1:IENTRY)
CALL YES_NO(IANS)
IF(IANS .EQ. 1)THEN
IPATTERN=0
ICOLOR=2
CALL LINCLR(2)
CALL PATTERN(4,MPX,MPY)
GOTO 5
ENDIF
612 CONTINUE
CLOSE(4)
RETURN
END

C--------------------------------------------------------
C SUBROUTINE FOR DRAWING THE CENTER LINES FOR THE R/P SECTIONS
C THIS SUBROUTINE STORES THE COORDINATES OF THE CENTRE LINES
C--------------------------------------------------------
SUBROUTINE SRICENTER(IND,MPX,MPY)
REAL MPX(4),MPY(4),SLP(4),INTRE(4),PX(2),PY(2)
DIMENSION CENSLP(7),CENFWIDTH(7),CENPCLLEN(7),CENCONS(7),
\CENC1(7),CENC2(7),CX(300,2),CY(300,2)
COMMON /CENIN/SLP=AZ,FWIDTH,PCLLEN,CONS,C1,C2,
\CENSLP,SLP,INTRE,FENFWIDTH,CENPCLLEN,
\CENCONS,CENC1,CENC2,OFFDIS,CX,CY,IPERFECT,IPERFECT,ICROSS
C OFFDIS WIDTH
COMMON /OFFSET/WIDTH
C THIS ROUTINE STORES THE CENTRE LINE COORDS
C ROUTINE CENTDRAW DRAWS THE ACTUAL LINES PASSED THROUGH CX,CY
IF((IND .EQ. 2) .OR. (IND .EQ. 4))THEN
WIDTH=(CENFWIDTH(IND)+CENPCLLEN(IND))/(2.*CENCONS(IND))
ELSE
WIDTH=CENFWIDTH(IND)/(CENCONS(IND)**4)
ENDIF
C IF(IND .EQ. 5)WIDTH=(CENFWIDTH(IND)/(CENCONS(IND)))+OFFDIS
C IF(WIDTH .EQ. 0.) WIDTH = (WIDTH/ABS(IFPO18))
ENDIF
I=1
IY=3
CONTINUE
IA=0
ITRACK =0
LT=1-6
ICOUNT =1
IIX=I
DO 50 ICK=1,4
IFLAG=0
IFLAGB =0
IPX=ICK+1
IF(IPX .EQ. 5) THEN
   IPIX=1
ENDIF
IF(IPERFECT .EQ. 4) THEN
  C PRINT *, ' IN THE PERFECT LOOP'
  IF(1CROSS .EQ. 0) THEN
     IF((IBASE.EQ.1) .OR. (IBASE.EQ.3)) THEN
       C PRINT *, ' BASE LINE 1 OR 3'
       PX(1) = MPX(1)
       PX(2) = MPX(2)
       PX(1) = PX(1) + ABS(WIDTH) * CONS
       PX(2) = PX(1)
       ELSE
       C PRINT *, ' BASE LINE 2 OR 4'
       FX(1) = MPX(1)
       PX(2) = MPX(3)
       PX(1) = PX(1) + ABS(WIDTH) * CONS
       PX(2) = PX(1)
       ENDIF
     ELSE
     IF((IBPERFECT.EQ.2) .OR. (IBPERFECT.EQ.4)) THEN
       C PRINT *, ' BASE LINE 1 OR 3'
       PX(1) = MPX(1)
       PX(2) = MPX(2)
       PX(1) = PX(1) + ABS(WIDTH) * CONS
       PX(2) = PX(1)
       ELSE
       C PRINT *, ' BASE LINE 2 OR 4'
       FX(1) = MPX(1)
       PX(2) = MPX(3)
       PX(1) = PX(1) + ABS(WIDTH) * CONS
       PX(2) = PX(1)
       ENDIF
  ENDIF
50 CONTINUE
ELSE
    CALL COORDCLOSEP(IND),SLP(Ick),WIDTH,CEMLE(IND)
    IF(INTE(Ick),A1.Y1)
    ENDIF
    IF(IXK.EQ.1) THEN
        IF(A1.GE.NPY(Ick)).AND.(A1.LE.NPY(Ick)) IFLAG=1
    ENDIF
    IF(IXK.EQ.2) THEN
        IF(A1.GE.MPY(Ick)).AND.(A1.LE.MPY(Ick)) IFLAG=1
    ENDIF
    IF(IXK.EQ.3) THEN
        IF(A1.LE.NPY(Ick)).AND.(A1.GE.NPY(Ick)) IFLAG=1
    ENDIF
    IF(IXK.EQ.4) THEN
        IF(A1.LE.MPY(Ick)).AND.(A1.GE.MPY(Ick)) IFLAG=1
    ENDIF
    IF(IFLAG.EQ.1) THEN
        ITRACK=ITRACK+1
        PX(IICOUNT)=A1
        PY(IICOUNT)=A1
        IICOUNT=IICOUNT+1
    ELSE
        CONTINUE
    ENDIF
ENDIF

checking for dimension
    IF(A1.GE.300) THEN
        PRINT *, 'REDUCE THE SECTION LENGTH'
        PRINT *, 'ERROR IN DEFINITION'
        STOP
    ENDIF
    CX(I,1)=PX(I)
    CY(I,1)=PY(I)
    CX(I,2)=PX(I)
    CY(I,2)=PY(I)
ENDIF

DRAWING THE LINES WITH THE ABOVE COORDINATES
    CALL MOVEA(PX(1),PY(1))
    IF(INE.EQ.6).OR.(INE.EQ.7) THEN
        CALL DRAWA(PX(2),PY(2))
    ELSE
        CALL DASHA(PX(2),PY(2),3)
    ENDIF
    CALL TEMO
    IF(INE.EQ.5) THEN
        IF(INAME.EQ.1) THEN
            INAME=0
            WIDTH=WIDTH-CEMALE(IND),CEMLE(IND))
SUBROUTINE FOR DRAWING PILLAR MANUALLY

DIMENSION (X(4),Y(4))
CHARACTER*1 ANG
CHARACTER*20 CHVAL(15)
REAL VALUES(10)
COMMON /INPUT/CHVAL,VALUES

CONTINUE
II=1
CHVAL(1)="PILLAR DEFINITION FOR POINT - 1/ANG"
CALL MAIN(1,CHVAL)
CHVAL(2)="(1) POINT THROUGH CURSOR"
CHVAL(3)="(2) POINT THROUGH REFERENCE POINT"
CHVAL(4)="(3) RETURN TO MAIN MENU"
CHVAL(4)="(4) LINE 4 - I"
CHVAL(4)="ENTER YOUR OPTION"
CALL OPTIONS(4,CHVAL,IBASE)
IF(IBASE LE 3) RETURN
IF(IBASE LE 1) THEN
   CALL VCURSOR(RED,1,1,1,1)
   ELSE
   CALL REFERENCE(1,1,1,1)
PRINT *, "ENTER THE LENGTH OF THE EXTENSION1000000."
CALL INPUT(1000000)

CONTINUE
II=1
CHVAL(1)="PILLAR DEFINITION FOR POINT - 1/ANG"
CALL MAIN(1,CHVAL)
CHVAL(2)="(1) POINT THROUGH CURSOR"
CHVAL(3)="(2) POINT THROUGH REFERENCE POINT"
CHVAL(4)="(3) RETURN TO MAIN MENU"
CHVAL(4)="(4) LINE 4 - I"
CHVAL(4)="ENTER YOUR OPTION"
CALL OPTIONS(4,CHVAL,IBASE)
IF(IBASE LE 3) RETURN
IF(IBASE LE 1) THEN
   CALL VCURSOR(RED,1,1,1,1)
   ELSE
   CALL REFERENCE(1,1,1,1)
PRINT *, "ENTER THE LENGTH OF THE EXTENSION1000000."
CALL INPUT(1000000)
CALCULATING THE ABSOLUTE CO-ORDINATES:
  X(IJ) = EXLENGTH*COS(IANGLE) + X(I)
  Y(IJ) = EYLENGTH*SIN(IANGLE) + Y(I)
END IF
CALL MOVEA(X(IJ),Y(IJ))
CALL OUTN(IJ)
CALL MOVEA(X(IJ),Y(IJ))
call points(x(ij),y(ij))
CALL TSEND
PRINT *, 'IS THE CORRECT POINT PLACED (YES=1 NO=0)'
CALL YES_NO(IANS)
IF(IANS .NE. 1) GO TO 25
IF (I.J .EQ. 1) THEN
  CALL MOVEA(X(1),Y(1))
  PRINT *, '---',X(IJ),',Y---',Y(IJ)
ENDIF
IF (I.J .NE. 1) THEN
  CALL MOVEA(X(IJ-1),Y(IJ-1))
  CALL DRAWA(X(IJ),Y(IJ))
  CALL TSEND
ENDIF
CALL TSEND
END DO
CALL DRAWA(X(1),Y(1))
call tsend
PRINT *, 'DO YOU WANT TO DRAW THE PILLAR AGAIN (YES=1 NO=0)'
CALL YES_NO(IANS)
C READ (*.20)ANS
C 20 FORMAT(A1)
  IF(IANS .EQ. 1) THEN
    DO 20 IJ=1,4
      IF (I.J .EQ. 1) THEN
        CALL MOVEA(X(1),Y(1))
      ENDIF
      IF (I.J .NE. 1) THEN
        CALL DRAWA(X(IJ),Y(IJ))
      ENDIF
    20 CONTINUE
  ENDIF
ENDIF
CALL TSEND
RETURN
END

C SUBROUTINE FOR STORING THE CENTERLINES INFO
SUBROUTINE EININFO(NS)
SUBROUTINE PROMPT FOR WRITING INDIVIDUAL FRACKS

SUBROUTINE PROMPT
CALL SETUP(1)
CALL ERASESCREEN(2)
CALL ROWMOVE(20)
CALL SETUP(0)
RETURN
END

SUBROUTINE FOR OUTPUTTING THE TITLES FOR THE DESIGN VALUE INPUT

SUBROUTINE DESMENU
CHARACTER*80, STR(10)
CALL SETUP(1)
CALL REVERSE(0)
CALL REVERSE(4)
STR(1) = "DEFAULT"
CALL SUBMENU(4, 20, STR(1))
STR(1) = "VALUES"
CALL SUBMENU(5, 20, STR(1))
STR(1) = "VARIABLES"
CALL SUBMENU(4, 35, STR(1))
STR(1) = "USER INPUT"
CALL SUBMENU(4, 55, STR(1))
CALL REVERSE(0)
RETURN
END

THIS SUBROUTINE HANDLES THE OPTION TYPE I/O ROUTINES FOR THE UNDERGROUND MINING PROGRAM. ALL THE ROUTINES ARE TENTATIVE, DEPENDENT.

DEVELOPED BY SRIKAR CHALUVADI ON DEC 4, 1985 FOR THESIS WORK

SUBROUTINE OPTIONS(INA, STR,IVAL)
BYTE 81(30)
CHARACTER*80 STR(30); STRING(11; JPT(1)
INTEGER COLUMN(3))
CHARACTER*1 AND
DATA START, COLUMN(5, 30)
DATA STR, " "/
DO 11 I=1, NA
   WRITE(*,5) STR(I)
   FORMAT(5)
   CALL COUNTCHAR(ACCOUNT (I) + STR(I))
   PRINT *, ACCOUNT (ACCOUNT, I)
   11 CONTINUE

END DO
C  READ *, IJUM
CALL SETUP(1)
C  CONTINUE
ISTART = 5
ICOLUMN = 20
CALL ROWMOVE(ISTART)
CALL ERASESCREEN()
CALL REVERSE(0)
C  CALL REVERSE(*):
C  CALL REVERSE(*):
C  WRITING THE STRINGS AT A SPACING OF 2 ROWS STARTING FROM ISTART
DO 10 JJ = 1, NW
ENCOD(WCOUNT(IJ), 10, 51, STR(IJ))
10 FORMAT(6(A(WCOUNT(IJ))))
IF (IJ .NE. NW) THEN
   CALL ROWMOVE(ISTART)
   ISTART = ISTART + 1
ELSE
   ISTART = ISTART + 4
   CALL ROWMOVE(ISTART)
ENDIF
   CALL HORMOV(ICOLUMN, WCOUNT(IJ), 31)
END DO
C  CALLING FOR THE BLANK SPACE FOR THE OPTIONS TO BE ENTERED
ICOLUMN = ICOLUMN + WCOUNT(NW) + 3
C  CHECKING FOR THE NUMBER OF BLANKS NEEDED
IF (NW - 1) .LE. 9 THEN
   ISLANK = 1
ELSE
   ISLANK = 2
ENDIF
12 CALL BLANKS(ISTART, ICOLUMN, ISLANK):
READ (*, 14) OPT(1)
14 FORMAT(20)
IF (OPT(1) .EQ. 'O' .OR. OPT(1) .EQ. '.') GOTO 12
CALL CHR_NUM(OPT(1), IIVAL)
IVAL = IIVAL
IF (IVAL .GE. NW) .OR. (IVAL .EQ. -99999) GOTO 12
IF (IVAL .LE. 0) GO TO 12
ICOL = 20
ICOL = IVAL - 1
IFLAM = 0
CALL REVERSE(1):
   CALL REVERSE(1):
ENCOD(WCOUNT(IVAL), 15, 51, STR(IVAL))
15 FORMAT(6(A(WCOUNT(IVAL))))
CALL ROWMOVE(1)
CALL HORMOV(ICOLUMN, WCOUNT(IVAL), 31)
C  PRINTING THE DOUBLE CHECK QUESTION

SUBROUTINE FOR PRINTING THE BLANK SPACE IN REVERSE

SUBROUTINE BLANKS1(ISTART,ICOLUMN,IBLANKS)
CHARACTER*10 BLANK
BYTE S1(4)
IF(IBLANKS .EQ. 1)BLANK=' '*
IF(IBLANKS .EQ. 2)BLANK='  '*
IF(IBLANKS .EQ. 3)BLANK='   '*
IF(IBLANKS .EQ. 4)BLANK='    '*
IF(IBLANKS .EQ. 5)BLANK='     '*
ENCOD(EIBLANKS+1,X0,X1)S1BLANK

CALL REVERSE(7)
CALL ROMove(ISTART)
CALL HORMOV((ICOLUMN-1),S1)
RETURN
END

SUBROUTINE BLANKS2(ISTART,ICOLUMN,IBLANKS)
CHARACTER*10 BLANK
BYTE S1(4)
IF(IBLANKS .EQ. 1)BLANK='  '*
IF(IBLANKS .EQ. 2)BLANK='   '*
IF(IBLANKS .EQ. 3)BLANK='     '*
IF(IBLANKS .EQ. 4)BLANK='       '*
IF(IBLANKS .EQ. 5)BLANK='         '*
ENCOD(EIBLANKS+6,X0,X1)S1BLANK

CALL REVERSE(7)
CALL ROWMOVE(ISTART)
CALL HORMOV(ICOLUMN,1BLANKS,SI)
CALL REVERSE(0)
RETURN
END

C-----------------------------
C ROUTINE FOR WRITING THE TEXT FOR PARMAATER INPUT
C-----------------------------
SUBROUTINE MENSCH(IH,STR,ICOLUMN)
BYTE SI(80)
CHARACTER*80 STR(10),STRING
INTEGER WCOUNT(10)
DATA ISTART,IT /
ISTART=7
DO 10 II=1,NW
  WRITE(*,5)STR(II)
  FORMAT(A80)
  CALL COUNTCHAR(WCOUNT(II),STR(II))
  PRINT *,", WCOUNT",WCOUNT(II)
10   END DO
  READ *,IDUM
  CALL SETUP(1)
  CALL REVERSE(1)
  CALL REVERSE(7)
  CALL REVERSE(4)
  CALL REVERSE(3)
  WRITE THE STRINGS AT A SPACING OF 1 ROWS STARTING FROM ISTART
  II=1,NW
  WRITE(*,10)COUNT(II),10,SI,STR(II)
10   FORMAT(A80)
    CALL ROWMOVE(ISTART)
    CALL HORMOV(ICOLUMN,WCOUNT(II),SI)
    ISTART=ISTART-2
  END DO
  RETURN
END

C-----------------------------
C ROUTINE FOR WRITING AT THE EXACT LOCATIONS
SUBROUTINE SUBMENU(IROW,ICOLUMN,STR)
BYTE SI(80)
CALL COUNTCHAR(ICOUNT,STR(1))
CALL HORMOV(ICOLUMN,ICOUNT,SI)
10 FORMAT(AICOUNT)
    CALL ROWMOVE(IROW)
    CALL HORMOV(ICOLUMN,ICOUNT,SI)
    RETURN
END
SUBROUTINE FOR CONVERTING CHARACTER DATA TO INTEGER DATA
SUBROUTINE CHR_NUM(STR,IVAL)
CHARACTER*80 STR(1), STRING
DATA STRING/''/
CALL COUNTCHAR(1COUNT, STR(1))
DECVAL=0.
IVAL=0.
ITCOUNT=1COUNT
IMINUS=0
STRING=STR(1)
IVAL=0.
I=1
10 INTER=-99999
WRITE(*,12)STRING
C 12 FORMAT(1X,'STRING ',13<1COUNT>)
IF(STR(1COUNT),EQ,'0')IINTER=0
IF(STR(1COUNT),EQ,'1')IINTER=1
IF(STR(1COUNT),EQ,'2')IINTER=2
IF(STR(1COUNT),EQ,'3')IINTER=3
IF(STR(1COUNT),EQ,'4')IINTER=4
IF(STR(1COUNT),EQ,'5')IINTER=5
IF(STR(1COUNT),EQ,'6')IINTER=6
IF(STR(1COUNT),EQ,'7')IINTER=7
IF(STR(1COUNT),EQ,'8')IINTER=8
IF(STR(1COUNT),EQ,'9')IINTER=9
IF(STR(1COUNT),EQ,'+')THEN
IMINUS=1.
IINTER=0.
ELSE IF(STR(1COUNT),EQ,'/')THEN
PRINT '*,' IN THE DECIMAL VALUE
IDECL=ITCOUNT-1COUNT
IVAL=IVAL/10.*<IDECL>
IVAL=0.
1COUNT=1COUNT-1
I=1.
IF(1COUNT .GE. 1)THEN
GO TO 10
ELSE
IVAL=DECL
IVAL=IVAL-IMINUS
IVAL=-IVAL
RETURN
END IF
END IF
ELSE IF(IVAL .EQ. -99999)THEN
IVAL=-99999
RETURN
END IF
IVAL=IVAL-INTER*I**(I-1)
I=I-
ICOUNT=ICOUNT-1
IF ICOUNT .GE. 13 TO 10
RVAL=IVAL-DEIVAL
IF I(2) .EQ. 1,IVAL=-RVAL
C PRINT *,IVAL,DEIVAL,RVAL I5,T,IVAL,DEIVAL,RVAL
RETURN
END
SUBROUTINE FOR TAKING CARE OF AZIMUTH CONVENTION

IF(ANGIN .GE. 180.) ANGOUT = 360. - ANGIN
IF(ANGIN .LT. 0.) ANGOUT = 360. + ANGIN
RETURN
END
C SUBROUTINE FOR TAKING CARE OF AZIMUTH CONVENTION
C FOR AEP MINES
C CONVERTS THE TRUE ANGLES INTO THE AZIMUTH FOR AEP
C DEVELOPED BY SRIDHAR CHALUVADI
C-------------------------------------------------------------------
SUBROUTINE AZIMUTH(ANGIN,ANGOUT)
IF(ANGIN.GT.360.)ANGIN=ANGIN-360.
IF((ANGIN.GE.0.) .AND. (ANGIN.LE.90.) ) THEN
   ANGOUT=ANGIN-90.
   ANGOUT=-ANGOUT
   RETURN
ENDIF
IF((ANGIN.GE.90.) .AND. (ANGIN.LE.180.) ) THEN
   ANGOUT=90.-ANGIN
   RETURN
ENDIF
IF((ANGIN.GE.180.) .AND. (ANGIN.LE.270.) ) THEN
   ANGOUT=-ANGIN+90.
   RETURN
ENDIF
ANGOUT=50.-ANGIN
RETURN
END
THIS SUBROUTINE IS USED FOR STORING PLOT FILE INFORMATION
AND LAYOUT DETAILS
THIS SUBROUTINE IS CALLED FROM THE MAIN PROGRAM "POLUT.".
WRITTEN BY BRIDHAR CHALUVADI JAN ZURBAG AS PART OF THESE WORK

THIS PROGRAM STORES THE FILES THAT ARE USED IN THIS PACKAGE
OPTIONS WILL BE GIVEN TO ADD AND DELETE THE FILES
CREATED FOR THIS PROGRAM

SUBROUTINE MAINDIREC(FNAMECUR,IVER,NSTAT)
NSTAT=0 FOR READ OPERATIONS
NSTAT=1 FOR WRITE OPERATIONS
CHARACTER*80 CHVAL(15)
CHARACTER*10 FName(15),ENC(DLMS)
CHARACTER*8 FNAMECUR(1)
CHARACTER*8 ENC,IVER(10),IVER(1)
COMMON /INPUT/CHVAL,VALUES

MAXIMUM NUMBER OF VERSIONS ARE LIMITED TO 10
MAXIMUM NUMBER OF MAIN FILES ARE LIMITED TO 15

CALL SETUP(0)
CALL DCLA
CALL DCLN(32)
CALL SETUP(11)
CALL ERASCREEN(2)
CALL MESSAGE(1)
OPEN UNIT=1,STATUS='OLD',ACCESS='READ',ORGANIZATION='INDEXED',
  FORM='FORMATTED',KEY=(1,1,CHARACTER),FILE='MAINDIREC.DAT',
  RECLEN=120,ERR=9999)
CALL MESSAGE(0)

IF NSTAT EQ 1 THEN THE DIRECTORY IS UPDATED
IF NSTAT .NE. 1 THEN
  WRITE(*,1)FNAMECUR(11)
  FORMAT (1,FNAMECUR(11))
  CALL MESSAGE(1)
  READ 1,X,Y=NAMECUR(1),ERR=9999)FILE=1,COM
  PRINT *,"REWRITING THE RECORD IN MAIN CUR".
  REWRITE(1,0)FNAMECUR(1),IVER(1)
  CALL SETUP(1)
  CALL MESSAGE(0)
RETURN
END IF

IF INSTAT=0 THEN THE DIRECTOR IS JUST DISPLAYED

IF (INSTAT.LT.0) THEN
CALL MESSAGE(1)
   CHVAL(I)="MAIN DIRECTORY (NO. OF VERSIONS):"
   CALL MAINDIR(I,CHVAL)
   I0=1
   READING THE RECORDS SEQUENTIALLY
   CONTINUE
   READ(3,100,END=10)FILENAME(I),VER(I)
100 FORMAT(A6,A8)
   IF(I0.LE.9) THEN
      ENCOD(I,14,ENCD(I))
   ELSE
      ENCOD(E2,14,ENCD(I))
   END IF
   ENCOD(E2,14,ENCD(I))
   CALL COUNTCHAR(I,FILENAME(I))
   ENCD(I)=FILENAME(I)
   ENCD(I)=ENCD(I+1:10)
   CHVAL(I)=("<"/ENCD//">")//FILENAME(I)
   ENCD(I)=CHVAL(I)
   CALL COUNTCHAR(I,CHVAL(I))
   PRINT *,"COUNT":IC
   CHVAL(I)=ENCD(I+1:10)//max.NER="//VER(I)
   CHVAL(I)=ENCD(I+1:10)//"//VER(I)://"
   I0=I0+1
   GO TO 5

AT THE END OF FILE, SCREEN MENU IS CREATED

10 CHVAL(I)="ENTER YOUR CHOICE"
   CALL OPTIONS(I,CHVAL,OCT)
   CALL SETUP(I)
   CALL MESSAGE(3)
   PRINT *,"ENTER THE VERSION NUMBER"
   READ *,IB,IVER(I)
   FORMAT(A8)
   PRINT *,"CALLING THE LOADDIR"
   CALL LOADDIR(FILENAME(I),IVER(I),I,STAT)
RETURN
END IF

*** CONTINUE
IF(INSTAT.LT.0) THEN
PRINT *, 'MAINDIR.DAT IS NOT FOUND'
CLOSE(3)
RETURN
ELSE
PRINT *, 'CREATING A NEW MAINDIR.DAT'
CLOSE(3)
CALL SETUP(1)
CALL MESSAGE(1)
OPEN(UNIT=5,STATUS='NEW',ACCESS='KEYED',ORGANIZATION=
  'INDEXED',FORM='FORMATTED',KEY=100,CHARACTER)
FILE=MAINDIR.DAT', FEOL=160
WRITE(5,100)FILENAME(1),IVER(1)
CALL SETUP(1)
CALL MESSAGE(0)
RETURN
END IF

IN CASE THE RECORD IS NOT FOUND FOR UPDATING
A NEW RECORD IS CREATED
CALL SETUP(1)
CALL MESSAGE(1)
WRITE(5,100)FILENAME(1),IVER(1)
PRINT *, 'A NEW RECORD IS WRITTEN IN MAINDIR'
CALL SETUP(1)
CALL MESSAGE(0)
RETURN
END

This subroutine keeps track of the information associated with each file and version. It is used to retrieve and create new files etc. All the information is stored in indexed files created by Erichar Chaluvadi for M.B. Thesis Work.

SUBROUTINE COALDIR(FNAME,IVER,NSTAT)

CHARACTER*80 CHVAL(15),FNAME(1),IVER(1),ENCOL,CHVAL1
REAL VALUES(100),SWIN(4)

CALL INPUT(CHVAL,VALUES)
CALL WINDOW(SWIN(1),SWIN(4))
VALUES(1)=SWIN(1)
VALUES(2)=SWIN(2)
VALUES(3)=SWIN(3)
VALUES(4)=SWIN(4)
CHVAL(1) = "DIRECTORY INFORMATION"
CALL MENUSCN(1, CHVAL)
IF (NSTAT .EQ. 1) CALL DEVENU
CHVAL(1) = "FILE NAME"
CHVAL(2) = "VERSION"
CHVAL(3) = "DATE"
CHVAL(4) = "DRAWN BY"
CHVAL(5) = "CHECKED BY"
CHVAL(6) = "LEGEND"
CHVAL(7) = "REMARKS"
CHVAL(8) = "FILE NAME"
IF (NSTAT .EQ. 1) THEN
   CALL MESSAGE(1)
   CALL MENUSCN(6, CHVAL, 35)
ELSE
   CALL SETUP(1)
   CALL MESSAGE(1)
   CALL REVERSE(0)
   CALL MENUSCN(6, CHVAL, 15)
END IF

IF (NSTAT .NE. 0) THE INFORMATION IS JUST DISPLAYED
   BUT THE BOUNDARY DEFINITION CAN BE CHANGED

OPEN UNIT=5, STATUS=OLD, ACCESS='KEYED', ORGANIZATION='INDEXED',
   FORM=FORMATTED, KEY=(11,16 CHARACTER), FILE=SLOEIR.jpg,
   RECFM=CD, EOB=999
ENCI=NAME(1) / IVEN(11)
READ 5, 39, KEY=ENCI, ERR=999, ENCI, CHVAL(1), CHVAL(2), CHVAL(3),
   CHVAL(4), CHVAL(5), CHVAL(6), CHVAL(7), CHVAL(8), VALUE(10)
ENCI=NAME(1) / IVEN(11)
READ 5, 39, KEY=ENCI, ERR=999, ENCI, CHVAL(1), CHVAL(2), CHVAL(3),
   CHVAL(4), CHVAL(5), CHVAL(6), CHVAL(7), CHVAL(8), VALUE(10)

FORMAT(448, 34, 34, 14, 999)
GO TO 50
   IN CASE THE ID IS NOT FOUND THE FILE IS CLOSED AND IS
   OPENED AGAIN IN APPEND MODE
   CHVAL(1) = "CRE"
   CHVAL(2) = "C"
   CHVAL(3) = "1/12/1995"
   CHVAL(4) = "CRE"
   CHVAL(5) = "DEM"
   CHVAL(6) = "NEW LW"
   CHVAL(7) = "TRIAL"
   CHVAL(8) = "MEIS6"
IF (NSTAT .EQ. 1) THEN
    WRITING BACK THE INFORMATION IN CASE OF
    NSTAT # 1
   CALL MENUSCN(6, CHVAL, 0)
   CALL VALUE(11, CHVAL)
   CALL SETUP(1)
CALL MESSAGE(1)
  PRINT *, 'REWRITING IN CIADIR'
  FNAME(1)=CHVAL(1)
  IVER(1)=CHVAL(2)
  ENCI=NAME(1)/IVER(1)
  READ(556,KEY=ENC1,ERR=998)CHVAL(1),CHVAL(2),CHVAL(3)
  CHVAL(4)=CHVAL(1),CHVAL(5)=CHVAL(2),CHVAL(6)=VALUES(11),I1=1
  READ(556,KEY=ENC1,ERR=998)CHVAL(1),CHVAL(2),CHVAL(3)
  CHVAL(4),CHVAL(5)=CHVAL(1),CHVAL(6),CHVAL(7)
  CHVAL(8),CHVAL(9)=VALUES(11),I1=1
CLOSE(15)
CALL MAINDIRECTORY(CHVAL(1),CHVAL(2),STAT)
RETURN
ELSE
  CALL SETUP(1)
  CALL REVERSE(0)
  CALL MENU(18,CHVAL(42))
  CALL MESSAGE(0)
  CALL SETUP(1)
  CALL SETUP(1)
  CALL ROWNO(30)
  WRITE(*,15)
  FORMAT(1X,'PRESS RETURN TO CONTINUE')
  READ(*,1)CHVAL(1)
  FORMAT(11)
  CHVAL(1)="WORKING AREA DEFINITION"
  CALL MAIN(1),CHVAL
  CALL USEMENU
  CHVAL(1)="LOWER LEFT X"
  CHVAL(2)="LOWER LEFT Y"
  CHVAL(3)="UPPER RIGHT X"
  CHVAL(4)="UPPER RIGHT Y"
  CALL MENU(14,CHVAL,55)
  IRON=
DO I2K=1,4
  CALL ROWNO(IRON)
  CALL NHORNO(20,7,VALUES(I2K))
  CALL BLANKS(1,IRON,55,10)
  IRON=IRON+2
END DO

CALL NVALEC(14,CHVAL,VALUES)
  SN=VALUES(1)
  SNK=VALUES(2)
  SNK2=VALUES(3)
  SNK3=VALUES(4)
  SNW=VALUES(1)
FORM = 'FORMATTED', RECL=130, CHS=130, FILE='SUBDIR.JAT',
& RECL=130,
  CALL SETUP(1)
  CALL REVERSE(0)
  CALL NAMECH(1,CHVAL(1))
  FNAM(1)=CHVAL(1)
  IVER(1)=CHVAL(2)
  ENCO=FNAM(1),IVER(1)
  WRITE(5,555,ERR=999)ENCO,CHVAL(1),CHVAL(2),CHVAL(3),
&      CHVAL(4),CHVAL(5),CHVAL(6),
&      CHVAL(7),CHVAL(8),VALUES(10),IC=1

CLOSE(5)
CALL MAINDIR(CHVAL(1),CHVAL(2),NSTAT)
RETURN
ELSE
  PRINT *, 'SUBDIR.JAT NOT FOUND'
RETURN
END IF
9995 PRINT *, 'UNABLE TO WRITE ON TO THE DIRECTORY'
  PRINT *, 'STORE THE FILE THROUGH THE OVERRIDE OPTION'
RETURN
130 CONTINUE
STOP
END
SUBROUTINE AREedef(MF1,MF2,MARKS)
CHARACTER*80, CHVAL(10), RED*1
DIMENSION N10(4)
REAL VALUES(10), MF1(4), MF2(4)
COMMON /INPUT/CHVAL, VALUES
COMMON /BASE/MARKS, X(300), Y(300), NOT(300), H(300), T(300), RRT(1000), NRT(1000), X(300), Y(300)
COMMON /DEBUG/CHVAL, RED*1
CONTINUE
CHVAL(1)="ENTER THE OPTION FOR AREA IDENTIFICATION"
CALL MAISO(1,CHVAL)
CHVAL(2)="(1) CORNERS BY CORNER NO."
CHVAL(3)="(2) CORNERS BY CURSOR"
CHVAL(4)="(3) CORNERS BY EDGE AND LENGTH"
CHVAL(5)="(4) CORNERS BY CENTER NO."
CHVAL(6)="(5) RETURN TO MAIN MENU"
CHVAL(7)="(6) ENTER YOUR OPTION"
CALL OPTIONS(6,CHVAL,IOPT)
CHVAL(1)="YES"
CHVAL(2)="NO"
CHVAL(3)="N"nnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn....
READ(*) (NID(IN),IN=1,4)
DO IN=1,4
  MPX(IN)=X(NID(IN))
  MPY(IN)=Y(NID(IN))
END DO
END IF
ELSE
  PRINT *,
  IF (IOPF .EQ. 2) THEN
    CALL SELF
    CALL PROMPT
    PRINT *, '---------------------------------------------------'
    PRINT *, 'ENTER THE FOUR CORNER POINTS WITH THE CURSOR'
    PRINT *, 'PRESS ANY KEY TO REGISTER THE POINT'
    PRINT *, 'ENTER THE POINTS STARTING AT THE LOWER LEFT HAND'
    PRINT *, '---------------------------------------------------'
    DO 22 ICKA=1,4
      CALL VOURS(RED,MPX(ICKA),MPY(ICKA))
    CONTINUE
  PRINT *, 'X ---',MPX(ICKA),'Y ---',MPY(ICKA)
ENDIF
C
C NEW OPTION FOR THE AEP MINES
IF (IOPF .EQ. 3) THEN
  CALL PAREDEF(MPX,MPY)
ENDIF
C
IF (IOPF .EQ. 4) THEN
  PRINT *, '---------------------------------------------------'
  PRINT *, 'ENTER THE FOUR POINTS IN A/
  PRINT *, 'CLOCK-WISE DIRECTION, STARTING FROM /
  PRINT *, 'LOWER LEFT HAND CORNER'
  PRINT *, '---------------------------------------------------'
  DO 22 ICKA=1,4
    PRINT *, 'ENTER THE POINT',ICKA
    CALL NODE_ID(MPX(ICKA),MPY(ICKA))
  END DO
  CALL NODE_ID(MPX,MPY,NPOINTS)
ENDIF
C
C SETTING THE LINE COLOR TO B
CALL LINESOL(3)
CALL MOVEA(MPX(1),MPY(1))
DO ICKA=2,4
  CALL DRAWA(MPX(ICKA),MPY(ICKA))
END DO
CALL DRAWA(MPX(4),MPY(4))
CALL TSEND
CALL LINESOL(1)
PRINT *, 'IS THE CORRECT AREA IDENTIFIED? (YES=1, NO=0)'
CALL YES_NO(IANS)
IF (IANS .EQ. 0) THEN
  CALL MOVEA(MPX(1),MPY(1))
ENDIF
DO ICX=1,4
       CALL DRAWA(MF*ICX+MF*ICX)
END DO
CALL DRAWA(HPX(1),HPY(1))
CALL TENEND
GO TO 1
END IF
RETURN
END
SUBROUTINE RDISPLAY
REAL MPX(4), MPY(4)
REAL VALUES(10)
CHARACTER*15 CHVAL(15), FNAME*10
COMMON /INPUT/CHVAL, VALUES
COMMON /WINDOW/THIN(4), SHIN(4)
PRINT *, 'ENTER THE OPTION (SLEEPER=1, DEV=0)'
CALL INPUT(SIN)
IF(SIN.EQ.0) THEN
  WHEN IFIRST=0 THEN THE INFORMATION IS NOT DISPLAYED
  WHEN IFIRST=1 THEN THE PROPER SECTION IS DISPLAYED FOR SUBSEQUENT USE IN THE SIMULATION ETC...
  IFIRST=0
  IDCOUNT IS USED FOR KEEPING TRACK OF THE SECTIONS FOR LATER IDENTIFICATION BY THE USER
  CALL TXTCOL(2)
  CALL LINEECOL(2)
  IDCOUNT=0
  CONTINUE
  IDCOUNT=IDCOUNT+1
  IF(IFIRST.EQ.1) THEN
    OPEN UNIT=4, STATUS='UNKNOWN', ACCESS='KEYED',
    & ORGANIZATION='INDEXED',
    & FORM='FORMATTED', KEY='(1,1,1,CHARACTER)', FILE='SLEDIREC'
    & REC=132
    OPEN UNIT=4, STATUS='OLD', FILE='SLEDIREC', ERR=500
  ELSE
    OPEN UNIT=4, STATUS='UNKNOWN', ACCESS='KEYED',
    & ORGANIZATION='INDEXED',
    & FORM='FORMATTED', KEY='(1,1,1,CHARACTER)', FILE='FILIDIREC'
    & REC=132
    OPEN UNIT=4, STATUS='OLD', FILE='FILIDIREC', ERR=500
  END IF
  REA(4,700,MPX,MPY,CHVAL,ENTRY,PLEN,PACT
  FORMAT('B10.0,E18.13,E12.7,12')
  IF IFIRST.EQ.0 OR IDCOUNT.EQ.0 THEN
    IF IFIRST.EQ.1 THEN CALL LINECOL(3)
    PRINT *, MPX, MPY
    CALL SHAPE(MPX, MPY)
    CALL TEXT...
  ELSE...
  END IF
CALL CENTER(+,MPX,MPY,CX,CY)
CALL HOVERCX,CY)
IF((CX.LT.SWIN2).AND. CY.LT. SWIN4) .AND.
1 (CX.GT. SWIN1) .AND. CY.GT. SWIN3)THEN
   CALL OUTN(1COUNT)
   CALL TSEND
ENDIF
END IF
GO TO 5
C
ERROR IN READ STATEMENT SIGNALS THE END OF THE DATA
C AT THIS POINT THE FILE IS AGAIN READ AND THE USER SPECIFIED
C SECTION IS PASSED ON TO THE SIMULATION
C
750 CONTINUE
   IF(IFIRST.EQ. 0) THEN
      CLOSE(+)
      PRINT *, 'ENTER THE SECTION SEQUENCE NUMBER'
      CALL INPUT(SIN)
      ISECTION=SIN
      IFIRST=1
      ICOUNT=0
      GO TO 5
   END IF
   PRINT *, 'IS THE CORRECT AREA IDENTIFIED (YES=1 NO=0)?'
   CALL YES_NO(IANS)
   IF(IANS.EQ. 0) THEN
      CALL LINECOL(2)
      ISECTION=0
      IFIRST=0
      ICOUNT=0
      CLOSE(+)
      GO TO 5
   END IF
   CLOSE(+)
   GO TO 650
800 PRINT *, 'ERROR IN THE DIRECTORY'
850 CONTINUE
RETURN
END
This subroutine is the main control program from which the simulation programs are run. The user creates the data in the following way before running the simulation program:

- simsect.dat: Section ID, avg figures, op shifts, advance, cut
- simyear.dat: Global year, month, and the working days
- simrel.dat: Section, month, shifts, cuts, advance...
- simrelat.dat: Gives the section relationships, say essential...

The program creates indexed files from above three files to be used during the simulation.

Subroutine finalsim!

Character*50 chval(1), chval(1), chval(1), chval(1), chval(1), chval(1), chval(1), chval(1), chval(1)

Main Menu for the Simulation

Continue

chval(1) = "underground mine production simulation menu"
call mainmenu(chval)

chval(1) = "Mine calendar of work days"
chval(2) = "Average production data for sections"
chval(3) = "Expansion production data for sections"
chval(4) = "Inter-relationship information of sections"
chval(5) = "Define sections for simulation"
chval(6) = "Simulate defined sections"
chval(7) = "Display simulation results"
chval(8) = "View simulation directory"
chval(9) = "Return to main menu"
chval(10) = "Enter your option"

call options(10, chval, chval)

For creating the calendar option

if (chval(1) == "Mine calendar of work days")
    call sectfile(chval)

   Creating a new file

if (chval(2) == "")
    print "Enter the area name"
READ (*,10) AREA
10 FORMAT (A1)

CREATING A FILE WITH THE INPUTNAME=DAY

FILE=AREA//DAY

OPEN UNIT=1, STATUS="NEW", ACCESS="KEYED", ORGANIZATION="INDEXED",
   FORM="FORMATTED", KEY=1:CHARACTER, FILE=FILE, REC=1321

PRINT *, 'ENTER THE DATA IN THE FOLLOWING FORMAT:'
PRINT *, 'TERMINATE WITH 0S IN ALL FIELDS:'
PRINT *, 'YEARS HOW DAYS'
PRINT *, '---------------------'
PRINT *, '1986 01 20 (EXAMPLE)'

READ (*,*,ERR=111) YEAR,IMONTH,IDAY

IROW=7
VALUES(1)=0.
VALUES(2)=0.
VALUES(3)=0.

CHVAL(1)="1. CALENDAR MENU ENTER 0 0 TO STOP."
CALL MAINE(I,CHVAL)
CALL DELENE
CHVAL(1)="2."
CHVAL(2)="3."
CHVAL(3)="4."
CALL MEDIT(1,CHVAL,5)
DO 10 IK=1,3
   CALL ROWMOV(IROW)
   CALL WDATE(IK,IROW,VALUES(1))
   CALL BLANK(1,IROW,5,10)
   IROW=IROW+2
   END DO

CALL DVIDE(I3,CHVAL,VALUES)
   YEAR=VALUES(1)
   IMONTH=VALUES(2)
   IDAY=VALUES(3)

IF (YEAR.NE.0) THEN
   ENCODE(+E,2,CHR_YEARS/YEAR

       FORMAT (A1)
   ENCODE(+E,2,CHR_MONTH/IMONTH

       FORMAT (A1)
FORMAT(2),
CH1_DUMY=CH1_YEAR//CH1_MONTH
PRINT *, 'CH1_DUMY IN WRITING'
WRITE(*,100)/CH1_DUMY 110
WRITE(4,130,ERR=1020)/CH1_DUMY(110),IDAY
FORMAT(A6,13)

30 TO 15
ELSE
CLOSE(4)
GO TO 5
END IF

Handling all the other cases in SUB-MENU

PRINT *, 'Enter the area name'
READ(*,101)AREA
(LFILE=AREA/IDAY
OPEN(UNIT=4,STATUS='OLD',ACCESS='I',ORGANIZATION='UNIFORM',\ FORM=FORMATTED,KEY=11BCHARACTER,FILE=FILE,\ REC=10B,ERR=1010)
PRINT *, '
PRINT *, 'Year Men Days'
PRINT *, '-------------------'
PRINT *, '1936 1 20  (example)' REA1.setX,ERR=111/YEAR,MONTH,IDAY
IROW=7
VALUES(1)=0,
VALUES(2)=0,
VALUES(3)=0,

CHVAL(1)= 'calendar modification menu'
CALL MAINEN(1,CHVAL)
CALL DESMENU
CHVAL(1)= 'year'
CHVAL(2)= 'month'
CHVAL(3)= 'working days'
CALL MENUSEN(3,CHVAL,3E)
DO 17 X=1,3
    CALL ROWMOV(IROW),
    CALL NOROWMOV(N1,X,VALUES(1)),
    CALL BLANKS(101,VALUES)
    IR=IR+X
17   CONTINUE

CALL RAWEM(3,CHVAL,VALUES)
YEAR=VALUES(1)
IF (ACTION .EQ. 1) THEN
  PRINT *, 'ENTER THE AREA NAME:'
  READ (*,1) IAREA

  CREATING A FILE WITH THE INPUTNAME

  IF (IAREA .NE. 'SE')
    \text{(Choose a file unit and status)}
    \text{with the 'NEW' access,}
    \text{organizing in 'INDEXED',}
    \text{formatted, with key 1 being character,}
    \text{and file 1.)

  \text{PRINT} *, 'ENTER THE DATA IN THE FOLLOWING FORMAT:'
  \text{PRINT} *, '(TERMINATE WITH 0 & IN ALL FIELDS)'
  \text{PRINT} *, '--------------------------------------------'
  \text{PRINT} *, '--------------------------------------------'
  \text{PRINT} *, 'SEE SHIFTS/DAY ADVANCE/CUT CUTE/SHIFT'
  \text{PRINT} *, '---------------------------------------'
  \text{PRINT} *, '  2  3  12  1  4 (EXAMPLE)'

  \text{READ} (*) ! ISED, SHIFTS, ADVANCE, CUTS

  IRW=7
  VALUES(1)=0.
  VALUES(2)=0.
  VALUES(3)=0.
  VALUES(4)=0.

  CHVAL(1)= 'SECTION DATA (ENTER 0 & 0 TO STOP)'
  CALL MAINSON(1,CHVAL)
  CALL SEEMENU
  CHVAL(1)= 'SECTION ID'
  CHVAL(2)= 'SHIFTS/DAY'
  CHVAL(3)= 'ADVANCE/CUT'
  CHVAL(4)= 'CUTS/SHIFT'
  CALL MENUED(1,4,CHVAL,SE)
  DO J=1,4
    CALL ROWMOVE(IFCM)
    CALL HROMOY(20,7,VALUES J)
    CALL BLANKS: IROW,ES,1,0
    IRW=IFCM+2
  END DO

  CALL EVALSON(4,CHVAL,VALUES)
  ISED=VALUES(1)
  SHIFTS=VALUES(2)
  ADVANCE=VALUES(3)
  CUTS=VALUES(4)
  IF (ISED .NE. 0) THEN
    \text{Encode 'SED' into the 'SED' format.}
    \text{WRITE(4,*) ! ISED, SHIFTS, ADVANCE, CUTS}
    \text{FORMAT(4,*) ! P,F3.0,P,E1.0,E3.0,}
    \text{GO TO 40}
  END IF

END
ELSE
CLOSE(4)
GO TO 6
END IF

HANDLING ALL THE OTHER CASES IN SUB-MAIN

PRINT *, 'ENTER THE AREA NAME'
READ (*) 10, AREA
FILE=AREA, 'SEE'
OPEN(UNIT=4, STATUS='OLD', ACCESS='KEYED', ORGANIZATION='INDEXED',
     FORM='FORMATTED', RECORD=10, FILE=FILE, UNREAD=10, ERR=10)
PRINT *, 'ENTER THE DATA IN THE FOLLOWING FORMAT:'
PRINT *, ' (TERMINATE WITH \ E IN ALL FIELDS)'
PRINT *, '----------------------------------------'
PRINT *, 'SEED SHIFTS/DAY ADVANCE/CUTS/SHIFTS'
PRINT *, '----------------------------------------'
READ (*) 11, ISEED, SHIFTS, ADVANCE, CUTS
ICH=7
VALUES(1)=0.
VALUES(2)=0.
VALUES(3)=0.
VALUES(4)=0.

CHVAL1 = 'SECTION DATA : ENTER \ 0 \ 0 \ TO STOP'
CALL MAINSON1(CHVAL1)
CALL DESIN
CHVAL1 = 'SECTION (1)'
CHVAL2 = 'SHIFTS/DAY'
CHVAL3 = 'ADVANCE/CUTS'
CHVAL4 = 'CUTS/SHIFTS'
CALL MAINSON1(CHVAL1, CHVAL2, CHVAL3, CHVAL4)
DO 10, I=1,4
   CALL ROWMOVE(IROW)
   CALL NORMOV(20,7,VALUES(I), I)
   CALL SPACE(11, IROW, 50)
   IROW=IROW+2
END DO

CALL VALUES(4), CHVAL, VALUES(1)
VALUES(2)=0.
VALUES(3)=0.
VALUES(4)=0.
EXEC (2, ISEED, ISEED)
READ(4,*) 10, ISEED, ERR=10)
CHRI_DUMMY(11)=DUMMY;DUMMY;DUMMY;

ADDING A NEW RECORD

IF I.ACTION EQ. 2 THEN
  PRINT *,' THIS DATA ALREADY EXISTS'
  PRINT *,' USE THE MODIFY OPTION TO CHANGE THE DATA'
  CLOSE(*);
  GO TO 5
END IF

MODIFYING A RECORD

IF I.ACTION EQ. 3 THEN
  REWRITE(*,0,ERR=1080);SHIFTS;ADVANCE;CLTS
  CLOSE(*);
  GO TO 5
END IF

NULLIFYING A RECORD (DELETING)

IF I.ACTION EQ. 4 THEN
  CHR1_DUMMY=’0’
  SHIFTS=0.
  ADVANCE=0.
  OUTS=0.
  DELETE(4,ERR=611)
  CLOSE(4);
  GO TO 5
END IF

WRITE(*,0,ERR=1083);CHR1_DUMMY(11);SHIFTS;ADVANCE;OUTS
CLOSE(*);
GO TO 5
END IF

IF I.CPT EQ. 3 THEN
  OPT(1)=’EXCEPTION’
  CALL EDFILE/OPT(1):I.ACTION

CREATING A NEW FILE

IF I.ACTION EQ. 1 THEN
  PRINT *,’ ENTER THE FILE NAME:
  READ *,FILE

CREATING A FILE WITH THE NAME (FILE)

FILE=FILE
OPEN(FILE=FILE,STATUS='NEW',ACCESS='READ',FORM=’FORMATED’,I-E=1111,CHARACTER)

L    READ=108;
      PRINT *, 'ENTER THE DATA IN THE FOLLOWING FORMAT:
      PRINT *, 'TERMINATE WITH * & E IN ALL ENDBS:
      PRINT *, '-----------------------------------------------
      PRINT *, 'DAY MONTH YEAR DAYShifts/Day ADVANCE CUTS/SHIFT:
      PRINT *, '-----------------------------------------------
      PRINT *, '4 1986 3 15 3 18 4 (EXAMPLE:)
      READ *,* YER=111 ISED=1 YEAR, MONTH, DAYS, Shifts, ADVANCE, CUTS
      IROW=7
      VALUES(1)=0.
      VALUES(2)=0.
      VALUES(3)=0.
      VALUES(4)=0.
      VALUES(5)=0.
      VALUES(6)=0.
      VALUES(7)=0.

      CHVAL(1)=" EXCEPTIOVALUES( ENTER * 0 0 TO STOP):
      CALL MAINENCH1, CHVAL)
      CALL DESMENU
      CHVAL(1)="ISED"
      CHVAL(2)="YEAR"
      CHVAL(3)="MONTH"
      CHVAL(4)="DAYS"
      CHVAL(5)="Shifts/Day"
      CHVAL(6)="ADVANCE/CUT"
      CHVAL(7)="CUTS/SHIFT"
      CALL MENUSCH1,CHVAL,3E1
      DC IROW=1,7
      CALL ROWMOVE(IROW)
      CALL REMOVE(IROW,10)
      CALL BLANK8(IROW,5,10)
      IROW=IROW+2
      END DO

      CALL CVALSCN(7,CHVAL,VALUES)
      ISED=VALUES(1)
      IYEAR=VALUES(2)
      IMONTH=VALUES(3)
      IDAYS=VALUES(4)
      IShifts=VALUES(5)
      ADVANCE=VALUES(6)
      ICUTS=VALUES(7)
      IF ISED,.NE.0 THEN
        WRITE(*,12)YER,ISED,_YEAR, MONTH,
               DAYS,Shifts,ADVANCE, CUTS
      12     FORMAT (I4,12,12,F2.2,F3.1)
      GO TO 55
ELSE
CLOSE(1)
GO TO 5
END IF

PRINT *, ' ENTER THE AREA NAME'
READ (*,0) AREA
FILE=AREA
OPEN(UNIT=4,STATUS="OLD",ACCESS="KEYED",ORGANIZATION="INDEXED",
FORM="FORMATTED",KEY=1+CHARACTER,FILE=FILE,
1 REC=132,ERR=1010)
PRINT *, ' ENTER THE DATA IN THE FOLLOWING FORMAT'
PRINT *, (TERMINATE WITH 0'S IN ALL FIELDS)'
PRINT *, -----------------------------'
PRINT *, SEQ YEAR MONTH DAYShifts DAY ADVANCE Cuts SHIFT'
PRINT *, ---------------------------------------------'
PRINT *, 1 1996 2 20 2 12 4 'EXAMPLE'
READ (*,*) ERR=811, ISEQ, IYEAR, IMONTH, IDAYS, IADV, ICUTS
VALUES(I1)=0.
VALUES(I2)=0.
VALUES(I3)=0.
VALUES(I4)=0.
VALUES(I5)=0.
VALUES(I6)=0.
VALUES(I7)=0.
END
C
CHVAL(1)=' EXCEPT VALUES ENTER 0 OR 1 TO STOP
CALL MAINCH(1,CHVAL)
CALL DESHEW
CHVAL(1)='SEQ'
CHVAL(2)='YEAR'
CHVAL(3)='MONTH'
CHVAL(4)='DAYS'
CHVAL(5)='SHIFTS/DAY'
CHVAL(6)='ADVANCE/CUT'
CHVAL(7)='CUTS/SHIFT'
CALL MENUSECN(7,CHVAL,3E)
DO 10 I3=1,7
   CALL READCON(ROW)
   CALL NTMPO(10,VALUES(I3))
   CALL BLANKS(ROW,VALUES(I3))
   IFOX(FOX)=0
10 CONTINUE
END 20
C
CALL CVALECN(7,CHVAL,VALUES)
VALUES(I1)=VALUES(I1)
IYEAR=VALUES(I3)
I. Month=VALUES(3)
2. Day=VALUES(1)
3. Shifts=VALUES(2)
4. Advance=VALUES(6)
5. Cuts=VALUES(7)
EXECUTE (4,5); READ (4,5); KEY = (5), ERR = 433

ADD A NEW RECORD

IF (ACTION = 2) THEN
PRINT *, 'THIS DATA ALREADY EXISTS'
PRINT *, 'USE THE MODIFY OPTION TO CHANGE THE DATA'
CLOSE (4)
GOTO 5
END IF

MODIFYING A RECORD

IF (ACTION = 3) THEN
REWRITE (4,5); ERR = 1080); SEQ = YEAR, MONTH,
$ IDAYS, SHIFTS, ADVANCE, CUTS
CLOSE (4)
GOTO 5
END IF

NULLIFYING A RECORD (DELETING)

IF (ACTION = 4) THEN
CHF1_DUMY = '0'
1: ERR = 0
1: MONTH = 1
SHIFTS = 0
ADVANCE = 0
CUTS = 0
DELETE (4, ERR = 11)
CLOSE (4)
GOTO 5
END IF

WRITE (4,5); ERR = 1080); SEQ = YEAR, MONTH,
$ IDAYS, SHIFTS, ADVANCE, CUTS
CLOSE (4)
GOTO 5
END IF

BUILDING THE RELATIONSHIP BETWEEN THE SECTIONS

IF (10 = 4) THEN
IF T1 = 'RELATIONSHIP'

CALL EDITFILE(I, ACTION)

CREATING A NEW FILE

IF ACTION == 1 THEN
    PRINT *, "ENTER THE NEW AREA NAME"
    READ (*, 10) IAREA

CREATING A FILE WITH THE INPUTNAME\RE

IFILE=IAREA\RE
OPEN(UNIT=4, STATUS=NEW, ACCESS="KEYED", ORGANIZATION="INDEXED"
& FORM="FORMATTED",KEY=11+CHARACTER);FILE=IFILE,
& REC=128)
PRINT +, 'ENTER THE DATA IN THE FOLLOWING FORMAT'
PRINT +, 'RELATION CODES'
PRINT +, '0 - SPECIFIED YEAR AND MONTH FOR START'
PRINT +, '1 - SEQUENTIAL TO THE RELATED SECTION'
PRINT +, '2 - CONCURRENT TO THE RELATED SECTION'
PRINT +, 'FOR ALL THE CASES THE OFFSET IS ADDED TO THE START'
PRINT +, '
PRINT +, 'SEQ REL-SEQ YEAR MONTH CODES OFFSET/DAYS'
PRINT +, '-----------------------------------------------'
PRINT +, '1 2 0 1966 2 0 16 (EXAMPLE)'
PRINT +, '1 4 2 0 0 1 16 (EXAMPLE)'
PRINT +, '2 4 0 0 2 2 (EXAMPLE)'
READ (*,ERR=50) ISEQ,IREL,ICYEAR,IMONTH,ICODE,IOFFSET

ERROR=1
VALUES(1)=0.
VALUES(2)=0.
VALUES(3)=0.
VALUES(4)=0.
VALUES(5)=0.
VALUES(6)=0.
VALUES(7)=0.

CHVAL(1)=DATES (0=SPECIFIED 1=BEIJIN 2=CONCURR, 0... TO STOP)
CALL MAINBCH1(CHVAL)
CALL DESCH1
CHVAL(1)=SECTION ID
CHVAL(2)=RELATED SECTION
CHVAL(3)=YEAR
CHVAL(4)=MONTH
CHVAL(5)=DAY
CHVAL(6)=RELATION
CHVAL(7)=INTEGER
CALL MAIBLEN1(CHVAL)
DO 10 I=1,7
CALL ROWMOV1(FRM)
CALL HORMO2((I0),VALUES,(10))
CALL BLNKM21(FROM, (10))
IRAWN=IRAWN+1

CALL QALECO1(I4, VALUES, I1)
IREL=VALUES(2)
LYEAR=VALUES(3)
IMONTH=VALUES(4)
iday=VALUES(5)
ICODE=VALUES(6)
OFFSET=VALUES(7)
IF (IREL .NE. 0.) THEN
   ENCOD4, 37, SEG1, IREL
   ENCOD4, 27, CHARS_REL, IREL
   WRIT4, 10, ERR=100, SEG1, CHARS_REL, I, YEAR, IMONTH,
   ICODE, OFFSET
   FORMAT(A4, A4, I4, I4, I4, 1E, 1F, 2)
GO TO 62
ELSE
   CLOSE(4)
GO TO 5
END IF
END IF

PRINT *, 'ENTER THE AREA NAME'
READ (*111), AREA
IFILE=AREA' (RE)
OPEN (UNIT=111, STATUS='OLD', ACCESS='DIRECT', FORM='FORMATTED', REC=512, ERR=1110)
PRINT *, 'ENTER THE DATA IN THE FOLLOWING FORMAT:'
PRINT *, ' ' RE A LATION CODES' *
PRINT *, ' 0 - SPECIFIED YEAR AND MONTH FOR START'
PRINT *, ' 1 - SEQUENTIAL TO THE RELATED SECTION'
PRINT *, ' 2 - CONCURRENT TO THE RELATED SECTION'
PRINT *, ' FOR ALL THE CASES THE OFFSET IS ADDED TO THE START'
PRINT *, ' ' VALUES (10)
PRINT *, ' ' SEED, REL-SEG YEAR, MONTH CODES OFFSET/CARDS' *
PRINT *, '.
PRINT *, ' ' EX=POLY
PRINT *, ' ' EX=POLY
PRINT *, ' ' EX=POLY
PRINT *, ' ' EX=POLY
READ (*111), ERR=1110, IREL, YEAR, IMONTH, ICODE, OFFSET
IRAWN=VALUES(1)
VALUES(2)=0.
VALUES(3)=0.
VALUES(4)=0.
VALUES(5)=0.
VALUES(6)=0.
VALUES(7)=0.

CHVAL(1)='DATES (0-SPECIFIED 1-SEQUENCE 2-CONCURRENT ... TO STOP)
CALL MAINSCH(1,CHVAL)
CALL DSECN
CHVAL(1)='SECTION-ID'
CHVAL(2)='RELATED SECTION'
CHVAL(3)='YEAR'
CHVAL(4)='MONTH'
CHVAL(5)='DAY'
CHVAL(6)='RELATION'
CHVAL(7)='OFFSET'
CALL MENSCH(7,CHVAL,35)
DO 13 I=1,7
   CALL RDMOV(1,IROW)
   CALL NDMOV(2:7,VALUES(I)).
   CALL BLANKS(1,IROW,25)
   IRCH=IROW+2
   ENDDO

CALL DVALSCH(7,CHVAL,VALUES)
ISED=VALUES(1)
IREL=VALUES(2)
IYEAR=VALUES(3)
IMTH=VALUES(4)
IDAY=VALUES(5)
ICODE=VALUES(6)
OFFSET=VALUES(7)
ENCODE(4:87,SED) ISED
ENCODE(4:87,CHAR_REL,IREL
READ(4:80,KEY=SED,ERR=585)
SEG=CHAR_DUMY,1DUMY,1DUMY,1DUMY,1DUMY

ADDING A NEW RECORD
IF ACTION .EQ. 2 THEN
   PRINT *,'THIS DATA ALREADY EXISTS'
   PRINT *,'USE THE MODIFY OPTION TO CHANGE THE DATA'
   CLOSE(1)
   GO TO 1
ENDIF

MODIFYING A RECORD
IF ACTION .EQ. 3 THEN
CLOSE(4)
GO TO 5
END IF

NULLIFYING A RECORD (DELETING)

IF (ACTION .EQ. 4) THEN
  CHA1_DUMY = '00'
  CHA2_DUMY = '00'
  IYEAR = 0
  IMONTH = 0
  ICODE = 0.
  OFFSET = 0.
  DELETE(4, EPR=611)
  CLOSE(4)
  GO TO 5
END IF

WRITE(4, 50, ERR=1020) SEQ, CHA1_DUMY, IYEAR, IMONTH,
     ICODE, OFFSET
  CLOSE(4)
  GO TO 5
END IF

OPTION FOR DEFINING THE SIMULATED SECTIONS

IF (OPT .EQ. 5) THEN
  AREA NAME IS SENT TO THE SUBROUTINE FOR STORING THE
  SECTION INFORMATION FOR FUTURE USE. THE INFORMATION IS STORED
  IN THE FILE AREA NAME /// 'SIM'
  CALL DEF_SEC (AREA NAME)
  GO TO 5
END IF

GO TO 611
1010 PRINT *, 'FILE DOES NOT EXIST. CHECK THE DIRECTORY'
  PRINT *, 'PRESS RETURN TO CONTINUE'
  READ(*,1012) CHVAL(11)
  GO TO 5
1020 PRINT *, 'ERROR WHILE WRITING ON TO THE FILE'
  PRINT *, 'CHECK THE DATA FIELDS'
  PRINT *, 'SECTION NAME MIGHT HAVE BEEN PREVIOUSLY DEFINED'
  PRINT *, 'PRESS RETURN TO CONTINUE'
  READ(*,1032) CHVAL(11)
  CLOSE(4)
  GO TO 5
1012 FORMAT(4I1)
111 PRINT *, 'ENTER NUMBERS'
   GO TO 45
211 PRINT *, 'ENTER NUMBERS'
   GO TO 45
311 PRINT *, 'ENTER NUMBERS'
   GO TO 45
411 PRINT *, 'ENTER NUMBERS'
   GO TO 10
511 PRINT *, 'ENTER THE NUMBERS AGAIN'
   CLOSE(4)
   GO TO 5
611 CONTINUE
   STOP
   END

SUBROUTINE FOR MANIPULATING THE CALENDAR DATA FILE
SUBROUTINE EDITFILE(OPT,IOPT)
CHARACTER*(15) CHVAL(IOPT),OPT(1)
REAL VALUE(IOPT)
CHVAL(1)= 'SETUP/EDIT'//OPT(1)
CALL MAINCN(1,CHVAL)
CHVAL(2)= '<1> CREATE A NEW'//OPT(1)//' FILE'
CHVAL(3)= '<2> ADD A NEW RECORD'
CHVAL(4)= '<3> CHANGE AN EXISTING RECORD'
CHVAL(5)= '<4> DELETE A RECORD'
CHVAL(6)= '<5> RETURN TO PREVIOUS MENU'
CHVAL(7)= '<6> ENTER YOUR OPTION'
CALL OPTIONS(IOPT,CHVAL,IOPT)
RETURN
END
SUBROUTINE ERIAREAS
CHARACTER*50 CHVAL(1:5), ANG*1
REAL VALUES(100)
REAL MAX(4), MFY(-1)
DIMENSION NID/4
COMMON /INPUT/VALUES, CHVAL
COMMON /FILES/COLOR, IPATTERN, INFILE, INC

THIS COMMON STATEMENT IS USED TO EXTRACT THE CORNER POINTS

FROM THE MAIN PROGRAM
COMMON /BASE/NOINT, INX(300), INY(300), NPT(300), NFLAG(300),
INTP(300), NEXTP(300), NT(300), NTT(300), TX(300), (EX), NCT(300), ID11
REAL, NDT, NDTT
CHARACTER*6 AREA, READ*1
DATE READ/"\"/

PRINT *, "'
PRINT *, 'LW AND R/P PLANNING'
PRINT *, '------------------------'
PRINT *, '1 - PLANNING A NEW AREA'
PRINT *, 'E - VIEW PLANNED AREA'
PRINT *, '3 - QUIT'
PRINT *, '------------------------'
READ *, CHOICE
CHVAL(1)="LW AND DEVELOPMENT PLANNING"
CALL MAINCH1(CHVAL)
CHVAL(2)="(1) DEFINE NEW AREA"
CHVAL(3)="(2) VIEW PLANNED AREA"
CHVAL(4)="(3) RETURN TO MAIN MENU"
CHVAL(5)="ENTER YOUR CHOICE"
CALL OPTIONS4(CHVAL, CHOICE)
IF (CHOICE .EQ. 1) GOTO 20
IF (CHOICE .EQ. 1) THEN
CONTINUE
END IF

E33 CONTINUE

533 CONTINUE
CHVAL(1)="ENTER THE OPTION FOR AREA IDENTIFICATION"
CALL MAINCH1(CHVAL)
CHVAL(1)="(1) CORNERS BY SNAP-IN HOLES"
CHVAL(2)="(2) CORNERS BY DIRECT"
CHVAL(3)="(3) CORNERS BY REFERENCE POINTS"
CHVAL(4)="ENTER YOUR CHOICE"
CALL OPTIONS4(CHVAL, IDFT)
IF (IDFT .EQ. 1) THEN
CALL SQL
READ(***,*,IN=1,CHID)
DO IN=1,4
MF Y1=(*,IN),
MF Y2=(*,IN),
END DO
END IF
IF (IDFT .EQ. 3) THEN
DO 10 IF=1,4
10 CONTINUE
CALL CURSXY(RAND,MIPX(IJ),MIPY(IJ))
PRINT *,' X=',MIPX(IJ),', Y=',MIPY(IJ)
IF (IJ .EQ. 1) THEN
  CALL MOVE(MIPX(IJ),MIPY(IJ))
ENDIF
IF (IJ .NE. 1) THEN
  CALL DRAW1(MIPX(IJ),MIPY(IJ))
ENDIF
CONTINUE
CALL DRAW1(MIPX(IJ),MIPY(IJ))
call tsend
ENDIF
IF (OPT .EQ. 3) THEN
  CALL PROMPT
  PRINT *,' ENTER THE FOUR CORNER NODE POINTS IN A' 
  PRINT *,' CLOCKWISE DIRECTION, STARTING FROM ' 
  PRINT *,' LOWER LEFT HAND CORNER' 
  PRINT *,' ' 
  PRINT *,'PRESS RETURN TO CONTINUE'
READ(*,2) CHVAL(I)
PRINT *,' ' 
DO IJ=1,4
  PRINT *,' ENTER THE CO-ORDINATES FOR POINT -------',IJ
  PRINT *,'PRESS RETURN TO CONTINUE'
  READ(*,10) CHVAL(IJ)
  CALL REFERENCE(MIPX(IJ),MIPY(IJ))
END DO
END IF
ICOLOR=3
CALL PATTERN(4,MIPX,MIPY)
ICOLOR=2
CALL SETUP(1)
CALL ERASER(2)
CALL ROWMOVE(I0)
CALL SETUP(0)
NODECENT=1
CALL AREADER(MIPX,MIPY,NODECENT)
CALL PROMPT
PRINT *,' DO YOU WANT TO REDEFINE THE AREA (YES=1, NO=0)?'
READ(*,104)RREAD
IF(RREAD .EQ. 1) THEN
  CALL PATTERN(4,MIPX,MIPY)
  GOTO 333
ENDIF
END IF
CALL SIMUL(MIPX,MIPY)
ELSE
SUBROUTINE SIMULA(MPX, MPY)

MPX and MPY are the corner coordinates of the selected area

REAL MPX(4), MPY(4), ANGLE(4), SLR(4), INTRE(4),
* PX(4), PY(4), FX(20,4), FY(20,4), PX2(80,4), FY2(80,4)
DIMENSION NMXE(3), IMSX(4), IMSY(4), DANG(4)
DATA PI/3.1415927/
INTEGER FUPA(4), IA(4)
character ANG, READ_FILE, FILENAME(10), AREAL, FILE(4)
COMMON /filename/letter, xfile, yfile, x
COMMON /FLR/IPFLR/IPFLA/IPFLA
COMMON /CHIPL/ CMPL/CMPL
REAL VALUE(10)
data xfile, yfile, 11, 0, 11

THE COMMON AREA IS DEFINED FOR WRITING INTO THE DATA FILE

COMMON /FILE/NIR, PIR, FFLY, FFLX, FFLX, PFL
COMMON /FILE/FILESTYPE, FILEST/IOLO
COMMON /FILE/NAME LAST, FILE, IPO
DATA 110/10/

CALL BASIC(1)

CALCULATING THE SLOPES, INTERCEPTS OF THE LINES _a \_y = \_b \_y = \_c

TAKE CARE WHEN THE SHAPE IS A "PERFECT" ONE

PERFECT = I

IF \_a < 1, 1 = \_a

IF \_a = 0, 1 = \_a

IF \_b = 0, 1 = \_b

IF \_c = 0, 1 = \_c

ENDDO
ENLARGING FOR THE PANEL ORIENTATION W.R.T. HORIZONTAL

ENLARGING FOR THE EASE LINE
CALL MOVEM(FPX(I),FPR(I)),
CALL OUTN(I),
CALL MOVEM(FPX(I+1),FPR(I+1)),
DO 31 IF(I-1.EQ.0),32,
CALL DRAW(FPX(1),FPR(1)),
CALL OUTN(1),
CALL MOVEM(FPX(N),FPR(N)),
CALL OUTN(N)
31 CONTINUE
CALL DRAW(FPX(I),FPR(I)),
CALL TEENO
CALL POMP
PRINT *, 'ENTER THE AREA NAME (AREA...)',READ(*,903)
903 FORMAT(5E15.10)
CALL CHRED(3)
CONTINUE
CALL POMP
PRINT *, 'ANGLES OF THE LINES ARE'
CALL ERASEREE(5)
CALL POMP
PRINT *, 'ORIENTATION OF THE SIDES'
CALL POMP
PRINT *, '------------------------------------------'
CALL SETUP(1),
CALL REVERSE(1),
CALL POMP
CALL PRINT(14)
DO 10 ID=1,4
   PRINT *,"LINE",ICK," ANGLE",CAMS(ICK)
10 END DO
CALL RChang(25)
CALL Reverse(7)
CALL Setup(1)
PRINT *,"PRESS RETURN TO CONTINUE"
READ(*,1101)CHVAL(1)
1101 FORMAT(1I1)
PRINT *,"ENTER ONE OF THE FOLLOWING FOR THE BASE LINE"
CHVAL(1)="ENTER THE OPTION FOR THE BASE LINE"
CALL MAIINAM(1)CHVAL:
CHVAL(1)="1", LINE 1 - 2"
CHVAL(2)="2", LINE 2 - 3"
CHVAL(3)="3", LINE 3 - 4"
CHVAL(4)="4", LINE 4 - 1"
CHVAL(5)="ENTER YOUR CHOICE"
CALL OPTIONS(5)CHVAL,ISBASE
CHVAL(1)="YES"
CHVAL(2)="NO"
CHVAL(3)="NO"
CHVAL(4)="NO"
CALL VALIDCH(4)CHVAL
IF(CHVAL(1).EQ.,"YES")ISBASE=1
IF(CHVAL(2).EQ.,"YES")ISBASE=2
IF(CHVAL(3).EQ.,"YES")ISBASE=3
IF(CHVAL(4).EQ.,"YES")ISBASE=4
PRINT *,"
PRINT *,"1 - BASE LINE 1_2"
PRINT *,"2 - BASE LINE 2_3"
PRINT *,"3 - BASE LINE 3_4"
PRINT *,"4 - BASE LINE 4_1"
READ(*,1125)ISBASE
CHVAL(1)="ENTER THE STARTING POINT ON THE LINE"
CALL MAIINAM(1)CHVAL:
CALL Setup(1)
CALL Reverse(10)
CALL RChang(10)
CALL Setup(10)
CALL Prompt
IF(ISBASE.EQ.,1)THEN
   PRINT *,"ENTER THE STARTING POINT ON LINE"
   READ *,ISTART
   CALL INPUT(2)
   ISTART=51
   IF(ISTART.EQ.,1)THEN
      IEND=2
   ELSE
      IEND=1
   END
ENDIF
ENDIF
IF (B,GE .,E0.3) THEN
  PRINT *, 'ENTER THE STARTING POINT ON E', B, '
  READ *, ISTART
  CALL INPUT (BIN)
  ISTART=BIN
  IF (ISTART .GE. 2) THEN
    IEND=3
  ELSE
    IEND=2
  ENDIF
ENDIF
ENDIF
IF (BASE .EQ. 3) THEN
  PRINT *, 'ENTER THE STARTING POINT ON G', B, '
  READ *, ISTART
  CALL INPUT (BIN)
  ISTART=BIN
  IF (ISTART .EQ. 3) THEN
    IEND=4
  ELSE
    IEND=3
  ENDIF
ENDIF
ENDIF
IF (BASE .EQ. 4) THEN
  PRINT *, 'ENTER THE STARTING POINT ON L', B, '
  READ *, ISTART
  CALL INPUT (BIN)
  ISTART=BIN
  IF (ISTART .EQ. 4) THEN
    IEND=1
  ELSE
    IEND=4
  ENDIF
ENDIF
ENDIF
CALCULATING THE END POINT

WRITE (*, 1000)
1000 FORMAT (1X, 'ENTER THE PANEL ORIENTATION ANGLE ----: ', A)
READ *, AIANG
MACH THE ANGLE ABSOLUTE
VALUES (1) = 0.0
VALUES (2) = 0.0
VALUES (3) = 35.0
CHVAL (1) = 100.0, PARAMETERS:
CALL XAANG (1), CHVAL
CALL DBEVP
CHANG (1) = "PANEL ORIENTATION"
CHVAL (1) = "A FACE WIDTH"
CHVAL (2) = "A DEVELOPMENT WIDTH"
CALL MENU Sơn13, CHVAL+33
CHVAL 1 = " F0.00"
CHVAL 2 = "300.00"
CHVAL 3 = "500.00"
CALL MENU Sơn13, CHVAL+33
IRD=#
DO IDY=1 TO 3
CALL BLA\$\$1(IRD, E\$\$1)
IRD=IRD+1
ENDDO
CALL OVALSON(I3, CHVAL, VALUES)
ADANG=VALUES(1)
CHECK NEW FOR CHECKING THE PERPENDICULARITY
CHECK=ADANG
FAWDH=VALUES(2)
FADTH=VALUES(3)
TAKING OUT THE APPROXIMATION
IF (ADANG .EQ. 90. .OR. (ADANG .EQ. 90. .AND. ADANG=ADANG+1))
ADANG=$ANGLA$ (ABE$ = ADANG
IF (ADANG=ADANG) .EQ. 90. .THEN
IF (ADANG .LT. 0. .THEN
ADANG=180.+ADANG
ELSE
ADANG=180.-ADANG
ENDIF
ENDIF
ENDIF
ENDDO
END\$\$}

CALCULATING THE STARTING AND ENDING LINES
CS=X(IStart)+BLAZ%A+X(IStart)
CI=MY(IEnd)+BLAZ%A+Y(IEnd)

INQUIRING FOR THE FACE WIDTH FOR THE LONGWALL AREA
PRINT *,' ENTER THE FACE WIDTH FOR THE LW AREA '
READ *, FAWDHT

FACHT DETERMINES WHICH IS THE FIRST POINT OR THE PANEL ETC...
PRINT *,' ENTER THE ROOF/FILLER WIDTH '
READ *, FAWDHT

IF THE ENDING LINE SLIPED IS LESS THAN THE BEGINNING LINE
THE WIDTHS ARE MADE NEGATIVE.
IF (CS<CI) .IS THEN
FAWDT=-FAWDT
FAWDT=-FAWHT
ENDIF
I=1
WIDTH=0.
TOTAL=0.
TPARE=0.
C IF ILFB IS MADE 0 INITIALLY SO THAT PANEL IS CALCULATED FIRST
IF ILFB=0
IPOINT=1
C PRINT * ;" CHOOSE ONE OF THE FOLLOWING"
C PRINT * ;'
C PRINT * ;' 1 - PANEL START'
C PRINT * ;' 2 - ROOM/PILLAR START'
C READ * ,IANS
CHVAL(1)="LONGWALL & DEVELOPMENT VARIATION DESIGN"
CALL MAINSCN(1,CHVAL)
CHVAL(1)="<1> NO VARIATION IN DESIGN"
CHVAL(2)="<2> VARIATION IN DESIGN"
CHVAL(3)="<3> RETURN TO MAIN MENU"
CHVAL(4)=" ENTER YOUR OPTION"
CALL OPTION(4,CHVAL,IVAR)
IF(IVAR .EQ. 3)RETURN
CHVAL(1)="ENTER THE OPTION FOR DESIGN"
CALL MAINSCN(1,CHVAL)
CHVAL(1)="<1> PANEL START'
CHVAL(2)="<2> DEVELOPMENT START'
CHVAL(3)="<3> RETURN TO MAIN MENU"
CHVAL(4)="ENTER YOUR CHOICE"
CALL OPTION(4,CHVAL,IANS)
C CHVAL(1)="YES"
C CHVAL(2)="NO"
C CALL AVALECN(2,CHVAL)
C IF(CHVAL(1) .EQ. "YES")IANS=1
C IF(CHVAL(2) .EQ. "YES")IANS=2
C IF(IANS .EQ. 3)RETURN
CALL PROMPT
IF(IANS .EQ. 1)THEN
IF(IVAR .EQ. 2)THEN
PRINT * ;" ENTER THE PANEL WIDTH FOR LW 1" CALL INPUT(XIN)
WIDTH=XIN
IF(FWIDTH .LT. 0)WIDTH=-WIDTH
ELSE
WIDTH=FWIDTH
END IF
WIDTH=WIDTH/COSEI(AZANS)
ELSE
IF(IVAR .EQ. 2)THEN
PRINT * ;" ENTER THE PANEL WIDTH FOR DEVELOPMENT 1" CALL INPUT(XIN)
WIDTH=XIN

IF (WIDTH .LT. 0) WIDTH = -WIDTH
ELSE
  WIDTH = WIDTH
ENDIF
    WIDTH = WIDTH / COSD(AZANG)
ENDIF
    IFLG = 1
    ENDIF
CONTINUE
ISTA = 0
ITRACK = 0
ICOUNT = 1
RIJK = 1
DO 50 ICK = 1, 4
  IFLAG1 = 0
  IFLAG2 = 0
  IPCX = ICK + 1
  IF (IPCX .EQ. 5) THEN
    IPCX = 1
  ENDIF
  CONS = COSD(AZANG)
  IF (IPERFECT .EQ. 4 .AND. A2CHECK .EQ. 90.) THEN
    PRINT *, '
    PRINT *, ' IN THE PERFECT LOOP'
  IF ([[IBASE .EQ. 1 .OR. (IBASE .EQ. 4)]) THEN
    IF (MPX(ISTART) .GT. MPX(END)) CONS = -CONS
    PRINT *, ' BASE LINE 2 OR 4'
    PY(1) = MPY(1)
    PY(2) = MPY(2)
    PX(1) = MPX(ISTART) + ABS(WIDTH) * CONS
    PX(2) = PX(1)
  IF (I .EQ. 1 .AND. IPCOUNT .EQ. 1) THEN
    CALL MOVEA(PX(1), PY(1))
    CALL DRAWA(PX(2), PY(2))
  CALL tsend
    PX(3) = MPX(ISTART)
    PY(3) = PY(2)
    PX(4) = PX(3)
    PY(4) = PY(1)
    IR = 1
    IP = 1
  ENDIF
ENDIF
END IF
IF ([[IBASE .EQ. 1 .OR. (IBASE .EQ. 3)]) THEN
  IF (MPX(ISTART) .GT. MPX(END)) CONS = -CONS
  PRINT *, ' BASE LINE 1 OR 3'
  PX(1) = MPX(1)
  PX(2) = MPX(2)
  PY(1) = MPY(ISTART) + ABS(WIDTH) * CONS
ENDIF

FY(2)=FY(1)
IF(1 .EQ. 1 .AND. ICOUNT .EQ. 1) THEN
CALL MOVEA(FX(1),FY(1))
CALL DRAWA(FX(2),FY(2))
call tsend
PX(3)=PX(2)
FY(3)=MPY(IstartDate)
PX(4)=PX(1)
FY(4)=MPY(IstartDate)
IF=1
IR=1
END IF
ENDIF
ICOUNT=3
ITRACK=2
AREA=SAREA(4,PX,FY)
GO TO 45
ENDIF
CALL COORD(SLPAZ,SLP(ICK)),WIDTH,CZ,INTRE(Icker),X1,Y1)
if(ICK .eq. 1) then
if((Y1 .GE. MPY(Icker)) .AND. (X1 .LE. MPX(Icker))) IFLAG=1
endif
if(ICK .eq. 2) then
IF((X1 .GE. MPX(Icker)) .AND. (X1 .LE. MPX(Icker))) IFLAG=1
endif
if(ICK .eq. 3) then
if((Y1 .LE. MPY(Icker)) .AND. (Y1 .GE. MPY(Icker))) IFLAG=1
endif
if(ICK .eq. 4) then
if((X1 .LE. MPX(Icker)) .AND. (X1 .GE. MPX(Icker))) IFLAG=1
endif
IF(IFLAG .EQ. 1) THEN
ITRACK=ITRACK+1
IAFLAG(ICOUNT)=ICK
PX(ICOUNT)=X1
FY(ICOUNT)=Y1
ICOUNT=ICOUNT+1
CONTINUE
endif
C DRAWING THE LINES WITH THE ABOVE COORDINATES
CALL MOVEA(FX(1),FY(1))
CALL DRAWA(FX(2),FY(2))
call tsend
C CALCULATING THE FIRST PIECE
IF(1 .EQ. 1) .AND. (ICOUNT .EQ. 1) THEN
IF=1
IR=1
CALL FIRST(SLPAZ,CZ,ISTART,PX,FY,MPY,SLP,INTRE,AREA)
GOTO 45
ENDIF
CALCULATING THE AREA

THIS PART TAKES CARE OF TRANSITION FROM ONE LINE TO THE OTHER

if (i .ne. 1) .and. (iarp .ne. iarflag(2)) .and.
  # (iarflag(2).eq.4) .and. (iarp .ne. i) then
  kn=abs(iarpre-iarflag(1))
  if ((abs(iarpre-iarflag(2)).eq.3) .and.
     # ((kn.eq.0) .or. (kn.eq.1))) then
    interchanging the coordinates
    dumx=px(3)
    dumy=py(3)
    px(3)=px(4)
    py(3)=py(4)
    px(4)=dumx
    py(4)=dumy
  endif
  AREA=SAREA(4,px,py)
  goto 45
endif

ENDIF

50 continue

WRITING ON THE LABELS

CHECKING FOR THE INITIAL PIECE

PX(2)=PX(1)
PY(2)=PY(1)
PX(4)=PX(1)
PY(4)=PY(1)
iarp=iarpflag(1)
iarpre=iarflag(2)

IF(WIDTH .NE. 0.) THEN
  IF(ITRACK .EQ. 2) THEN
    A=ABS(C1-C2)
    IF(IIPILG .EQ. 1) THEN
      PRINT *, ' AREA OF THE DEVELOPMENT', IIPCOUNT, ' IS', AREA
      IIPCOUNT=IIPCOUNT+1
      IIPILG=0
      TAREA=TAREA+AREA
    ELSE
      ID=1
      IF(AREA .EQ. 0.) ID=I-1
      PRINT *, ' AREA OF THE PANEL', ID, ' IS', AREA
      TAREA=TAREA+AREA
      IIPILG=1
      I=I+1
    ENDIF
  ENDIF
  ENDIF
ENDIF

CALCULATING THE PANEL DISTANCE FROM THE CORNER
RISK=1
IF (IIPILG .EQ. 0) THEN
IF(IVAR .EQ. 2) THEN
    PRINT *, 'ENTER THE PANEL WIDTH FOR NEXT LEVEL'
    CALL INPUT(WIN)
    WIDTH=WIN
    IF(FWIDTH .LT. 0) WIDTH=-WIDTH
    ELSE
        WIDTH=FWIDTH
    END IF
    WIDTH=WIDTH+WIDTH/CGSD(AZANG)
ENDIF

C CALCULATING THE ROOM/PILLAR DISTANCE FROM THE CORNER
IF(IPILFG .EQ. 1) THEN
    IF(IVAR .EQ. 2) THEN
        PRINT *, 'ENTER THE PANEL WIDTH FOR NEXT DEVELOPMENT'
        CALL INPUT(WIN)
        WIDTH=WIN
        IF(FWIDTH .LT. 0) WIDTH=-WIDTH
        ELSE
            WIDTH=FWIDTH
        END IF
        WIDTH=WIDTH+WIDTH/CGSD(AZANG)
    ENDIF
    IPILFG=1
ENDIF

IF((ABS(C1-C2)).LT.(ABS(WIDTH))) GOTO 200
CALL MOVEA(PX(1),PY(1))
CALL TCENTER(PX,PY)
    IF(IPILFG .EQ. 0) THEN
        CALL OUT(N(IR))
        IR=IR+1
    ELSE
        CALL OUT(N(IP))
        IF(AREA .NE. 0) IP=IP+1
    ENDIF
S0 TO 77

200 CONTINUE
IF(IPILFG .EQ. 1) IR=IR-1
IF(IPILFG .EQ. 0) IP=IP-1
FAREA=ABS(SAREA(4,MPX,MPY))
WAREA=FAREA+TAREA-TPAREA
PLAN=TAREA*100./FAREA
PARP=TPAREA*100./FAREA
FPAW=WAREA*100./FAREA
CALL PROMPT
PRINT *, ','
PRINT *, '------------' '---------' 'FAREA
PRINT *, 'TOTAL AREA' '---------' 'FAREA
PRINT *, 'L/A AREA' '---------' 'TAREA
PRINT *, 'R/P AREA' '---------' 'TPAREA
PRINT *, 'LEFT OVER AREA' '---------' 'WAREA
PRINT *, '----------------',
PRINT *, ' ' PRINT *, '{: PERCENTAGE VALUES' PRINT *, ' ' WRITE(*,1006)FPLW,PROP,FWA
1006 FORMAT(5X,'LW AREA ---','F5.1,','X9','I/5X,'R/P AREA ---','F5.1,','X9') C PART FOR WRITING ON TO A DATA FILE PRINT *, ' DO YOU WANT TO STORE THIS INFORMATION (YES=1 NO=0)' C READ(*,104)ANS C 104 FORMAT(A1) C IF(ANS.EQ.'Y') THEN CALL YES_NO(ANS) IF(ANS.EQ.'I') THEN C OPENING THE DIRECTORY FILE FOR STORING THE NEW FILE OPEN(LUNIT=12,FILE='LWARP.DAT',STATUS='UNKNOWN', ACCESS='APPEND',ERR=360) C READING THE FILE TO THE END IF OLD FILE PRINT *, ' ENTER THE FILE NAME' READ(*,105)NAME 105 FORMAT(A10) WRITE(12,505)NAME 505 FORMAT(A5,5A10) CLOSE(12) OPEN(UNIT=15,FILE=NAME,STATUS='NEW') WRITE(15,'(1X,1A,1X)') I,IP DO 520 IFK=1,IP 520 WRITE(15,'(1X,1A,1X)') (IFX(IFK,IKK),IFY(IFK,IKK)),IKK=1,4) DO 530 IFK=1,IP 530 WRITE(15,'(1X,1A,1X)') HR(IFK) 530 WRITE(15,'(1X,1A,1X)') (HR(IFK,IKK)),IKK=1,4) CLOSE(15) ENDIF GOTO 610 610 PRINT *, ' ERROR IN FILE' C ASKING FOR R/P DESIGN 610 PRINT *, ' ' VALUES(1)=120. VALUES(2)=60. VALUES(3)=65. VALUES(4)=0. CHVAL(1)='120.' CHVAL(2)='60.' CHVAL(3)='65.' CHVAL(4)='0.' PRINT *, ' DO YOU WANT TO DESIGN ANY R/P SECTION (YES=1 NO=0)' C READ(*,104)ANS C CALL YES_NO(ANS) C IF(ANS.EQ.'I') THEN
LST=1
CALL PROMPT
PRINT *, ' ENTER THE R/P ID FOR DESIGN'
READ *, IPD
CALL INPUT(GIN)
IPD=GIN

THE FOUR COORDINATES ARE SORTED ACCORDING TO THE
R/P DESIGN PROGRAM
IF(NR(IPD) .LE. 3) THEN
  FRX(IPD,4)=FRX(IPD,1);
  FRY(IPD,4)=FRY(IPD,1)
ENDIF
645 CONTINUE
WRITE(*,*)((FRX(IPD,IL),FRY(IPD,IL)),IL=1,4)
DO 650 ISORT=1,4
  DSRX(ISORT)=FRX(IPD,ISORT)
  DSRY(ISORT)=FRY(IPD,ISORT)
IF(FRX(IPD,4) .LT. DSRX(1)) THEN
  FRX(IPD,1)=DSRX(4)
  FRY(IPD,1)=DSRY(4)
  FRX(IPD,2)=DSRX(3)
  FRY(IPD,2)=DSRY(3)
  FRX(IPD,3)=DSRX(2)
  FRY(IPD,3)=DSRY(2)
  FRX(IPD,4)=DSRX(1)
  FRY(IPD,4)=DSRY(1)
  GOTO 645
ENDIF
IF(FRX(IPD,2) .LT. FRY(IPD,1)) THEN
  FRX(IPD,1)=DSRX(2)
  FRY(IPD,1)=DSRY(2)
  FRX(IPD,2)=DSRX(1)
  FRY(IPD,2)=DSRY(1)
  FRX(IPD,3)=DSRX(4)
  FRY(IPD,3)=DSRY(4)
  FRX(IPD,4)=DSRX(3)
  FRY(IPD,4)=DSRY(3)
  GOTO 645
ENDIF

CHECKING FOR THE ORDER AGAIN
SLSORT1=SLOPE(DSRX(3),DSRY(3),DSRX(1),DSRY(1))
CSORT1=KXCT(SLSORT1,DSRX(3),DSRY(3))
SLSORT2=SLOPE(DSRX(2),DSRY(2),DSRX(4),DSRY(4))
CSORT2=KXCT(SLSORT2,DSRX(2),DSRY(2))
IF(CSORT2 .LE. CSORT1) THEN
  FRX(IPD,1)=DSRX(4)
  FRY(IPD,1)=DSRY(4)
  FRX(IPD,2)=DSRX(1)
  FRY(IPD,2)=DSRY(1)
  FRX(IPD,3)=DSRX(2)

C  FRY(IFD,3)=DSRY(2)
C  FRX(IFD,4)=DSRX(3)
C  FRY(IFD,4)=DSRY(3)
C  GOTO 445
C  ENDIF
C  DO 445 IABC=1,4
C  DSRX(IABC)=FRX(IFD,IABC)
C  DSRY(IABC)=FRY(IFD,IABC)
C  I0L0=0
C  CALL SIPLR(DSRX,DSRY,VALUES,CHVAL)
C  PRINT *,"DO YOU WANT TO DESIGN ANOTHER R/P SECTION (YES=1 NO=0)?"
C  READ (*)0104ANS
C  CALL YES_NO(IANS)
C  GOTO 415
C  ENDIF
C  PRINT *,"DO YOU WANT TO REDESIGN THE LW AND R/P AREA (YES=1 NO=0)?"
C  READ (*)0104ANS
C  CALL YES_NO(IANS)
C  IF(IANS .EQ. 1) THEN
C  IPFIL=0
C  IPATTERN=0
C  CALL PATTERN(4,MPX,MPY)
C  GOTO 5
C  ENDIF
C  call finit(0,760)
C  RETURN
C--------------------------------------------------------
C  SUBROUTINE FOR CALCULATING THE FIRST PIECE
C--------------------------------------------------------
C  SUBROUTINE FIRST(GLPAZ,CE,ISTART,PX,PY,MPX,MPY,SLP,INTRE,AREA)
C  DIMENSION IFL(1),PX(4),PY(4),MPX(4),MPY(4)
C  REAL MPX(4),MPY(4),INTRE(4),SLP(4)
C  CHECKING FOR TRIANGULAR AREA
C  checking for triangular area, if exists.
C  PRINT *," ISTART IS",ISTART
C  WIDTH=0.
C  INTRE=0
C  DO 50 ICK=1,4
C  IFLAG=0
C  IFLAG=0
C  IPCK=ICK=1
C  IF(IPCK .EQ. 5) THEN
C  IPCK=1
C  ENDF
C  ICASE=ISTART-1
C  IF(ICASE .EQ. 0) ICASE=4
C  IF(ICK .EQ. ISTART) OR (ICK .EQ. ICASE) GOTO 50
C  CALL COORD(GLPAZ,SLP(ICK),WIDTH,CE,INTRE(ICK),X1,YP1)
SUBROUTINE DETERM
BYTE S1(2), S2(2)
INTEGER SDATE, SDAS, SDAE, IP1(15), SDATE1
CHARACTER*6 AREA(15), ARE
CHARACTER*10 FNAME
CHARACTER*1 TY(15), DUMP*, S1, S2
DIMENSION SJP(4), SYP(4), SDAS(15), SDAE(15), LABS(15), LABEL(15)
COMMON/ALL/MPX, MPY, IP, IR, FPX, FPY, FXY, FRX, NR
COMMON /DET1/ ARE, TYPE, IPD
COMMON /PANELS/IPAN, IPNO
COMMON /CHOICE/ICHOICE
COMMON /CUMUL/LABEL, CUMSHIFT, LABELMR, LABELMG, SDATE
COMMON /window/twin(4), swin(4)
COMMON /DET2/IDLU, IDETAIL, AREAFACT
DATA DUMP'/', 'DAT'/
DETAIL = 1
PRINT *, 'DETAILED SIMULATION'
PRINT *, '----------------------------------'
PRINT *, '1 - SPECIFIED START DATE'
PRINT *, '2 - START DATE RELATE TO PREVIOUS SECTIONS'
READ *, ICHILD
IF(ICHILD .EQ. 1) THEN
  PRINT *, 'ENTER THE STARTING DATE OF SECTION'
  PRINT *, 'ENTER THE MONTH(MM), YEAR(YYYY)'
  READ (**), LABELMR, LABELMG
  CONVERTING TO DAYS
  CALL YMDY(LABEL, SDATE, LABELMR, LABELMG, 1)
  SDAS(I) = SDATE
  LABS(I) = LABEL
ELSE
  PRINT *, 'CHOOSE ONE FROM THE FOLLOWING'
  WRITE(*, 35)
      FORMAT(1X, 3X, 'AREA', 7X, 'TYPE', 5X, 'NO. ')
  35      FORMAT(1X, 3X, ' ', 'A6', 8X, 'A1', 3X, 'I2')
  DO 1 I = 1, 1
      WRITE(*, 25) IDLU, AREAL, AREA(IDLU), TY(IDLU), IP1(IDLU)
  25  END DO
  READ *, INEXT
  PRINT *, '1
  PRINT *, 'CHOOSE ONE FROM BELOW'
PRINT *, '---------------------'
PRINT *, ' 1 - SEQUENTIAL TO THE AREA SELECTED'
PRINT *, ' 2 - CONCURRENT TO THE AREA SELECTED'
READ *, ICON
IF (ICON .EQ. 1) THEN
  SDATE=SDAE(INEXT)
  LABELY=LABE(INEXT)
ELSE
  SDATE=SDAS(INEXT)
  LABELY=LABS(INEXT)
END IF
PRINT *, 'CHECKING SDAS AND LABS', SDAS(INEXT), LABS(INEXT)
END IF

ICHoice=1
IF (TYPE .EQ. 'L') THEN
  IFLAG=1
ELSE
  IFLAG=0
END IF

C CONSTRUCTING A TEMPORARY DATAFILE

C THE FILE UNIT IS PASSED THROUGH THE COMMON /DET1/
FNAMEl1E2=ARE(2:3)
FNAMEl3=TYPE(1:1)
ENCODEl3=3,102,311P Dol
102 FORMAT(I3)
FNAMEl6=SI(6:3)
FNAMEl7=DISPLAY(1:4)
WRITE(*,106)FNAMEl7
106 FORMAT('FILE CONS',A10)
ILLU=1
IDETAIL=1
OPEN (UNIT=I, FILE=FNAMEl1E2, STATUS='NEW')
CALL INFO(IPD, I)
IPAN=IPD
IPNO=1

C----------------------------------------------

C INITIALIZING THE START DATE FOR THIS SECTION
SDAS(I)=SDATE
LABS(I)=LABELY
PRINT *, ' CHECKING I,SDAS,LABE',I,SDAS(I),LABS(I)

C----------------------------------------------

CALL ACTSIM(4, GAREA, SXF, SYF, IFLAG)
PRINT *, '-------CHECKING I, LABEL(I), SDAE(I)'
LABELI=LABELY
SDAE(I)=SDATE
PRINT *, ' I', LABELI, SDAE(I)
PRINT *, ' DO YOU WANT TO CONSIDER ANOTHER AREA (Y/N)?'
READ *, SDAE(I)
50 FORMAT(A1)
IF (ANS .EQ. 'Y') THEN
I=I+1
GOTO 5
ENDIF

C THIS PART OF THE PROGRAM IS USED FOR THE ANALYSIS
C OF THE ALREADY SIMULATED AREAS
C EACH OF THE DATA FILE IS CHECKED FOR THE CONDITIONS

CALL PLOTANS
PRINT *, 'ANALYSIS'
PRINT *, '-----------------------'
PRINT *, '1 - BY TARGET DATE'
PRINT *, '2 - BY TARGET TONNAGE'
PRINT *, '3 - QUIT'
PRINT *, '-----------------------'
READ *, ICHOICE
IF ( ICHOICE .EQ. 1 ) THEN
  PRINT *, 'ENTER THE MONTH AND THE YEAR'
  READ ( *, * ) LABELMO1, LABELYR
  CALL YMDYD ( LABEL1, SDATE1, LABELYR, LABELMO1, 1 )
  PRINT *, 'ENTERED YEAR, DATE', LABELYR, LABELMO1
ENDIF
IF ( ICHOICE .EQ. 2 ) THEN
  PRINT *, 'ENTER THE TONNAGE CUTOFF'
  READ *, TONS
ENDIF
IF ( ICHOICE .EQ. 3 ) THEN
  DO IA=1, 1
    CLOSE ( UNIT=IA, STATUS='DELETE' )
  END DO
  RETURN
ENDIF

C CHECKING THE DATAFILES
CUMPROD=0.

PRINT *, 'ENTER THE TONNAGE'
READ *, ANG1
CALL TXTPR
CALL GRUT ( ANG1 )
CALL SIZE ( 1, 1 )
DO 850 IA=1, 1
  REMIND ( UNIT=IA )
  750 READ ( IA, 800, END=650 ) SXP, SYP, LABEL, LABELMO, PTPROD
  PRINT *, 'YEAR', LABEL, 'MO', LABELMO
  CUMPROD=CUMPROD+PTPROD
  800 FORMAT ( BF7.1, 14, 1E, F10.0 )
C CALL YMDYD ( LABELYR, SDATE, LABEL, LABELMO, IDDAY )
  XD=XI-X2
C IF ( XD .EQ. 0. ) XD=.01
C SLOP2=(Y1-Y2)/XD
C IF ( SLOP2 .EQ. 0. ) SLOP2=.001
C SLOP1=4TAND ( SLOP2 )
C IF ( SLOP1 LT 0. ) SLOP1=SLOP1+180.
XIC=(SXP(1)+SXP(2)+SXP(3)+SXP(4))/4.
YMIN=SXP(1)
C CHECKING FOR THE LOWEST POSITION
DO MIN=2,4
IF(SXP(MIN) .LE. YMIN)YMIN=SXP(MIN)
END DO
YIC=YMIN
CALL MOVEA((X1+5.),YIC)
CALL YEAR(LABEL,LA)
PRINT *,' LABEL',LABEL,' LA',LA,' LABELMO',LABELMO
ENCOD(2,104,S11)LABELMO
ENCOD(2,104,S2)LA
FORMAT(12)
CALL SELPAT(IPATTERN)
C CALL FILMOD(0,1,0)
C CALL MOVEA(X1,YIC)
C CALL VBGPHL(X1,YIC,1)
call ENDPNL
ITIX=INT((XIC-SWIN(1))*4000./(SWIN(2)-SWIN(1)))
ITIY=INT((YIC-SWIN(3))*4000./(SWIN(4)-SWIN(3)))
IF(CHOICE .EQ. 1)THEN
IF(LABEL .LE. LABELY)THEN
  IF(LABEL .LT. LABELY)THEN
    PRINT *,' YEAR IS LESS'
    CALL PATTERN(4,SXP,SYP)
    CALL TEXT2PLQT(XIC,YIC,LABELMO,LA,AN1)
    CALL TMOVE(ITIX,ITIY)
    CALL GTEXT1(E,S11)
    CALL GTEXT1(E,S2)
  ELSE
    IF(LABELMO .LE. LABELMO1)THEN
      PRINT *,' MONTH IS LESS'
      CALL PATTERN(4,SXP,SYP)
      CALL TEXT2PLQT((XIC),YIC,LABELMO,LA,AN1)
      CALL TMOVE(ITIX,ITIY)
      CALL GTEXT1(E,S11)
      CALL GTEXT1(E,S2)
    ENDIF
  ENDIF
ENDIF
GOTO 750
ELSE
GOTO 850
ENDIF
ELSE
IF(CUMP3 .LE. TCNS)THEN
  CALL PATTERN(4,SXP,SYP)
  GOTO 750
ELSE
  CUMP3=0.
GOTO 850
ENDIF

ENDIF

550 CONTINUE
GOTO 55

C THIS SUBROUTINE IS USED FOR DETAILERED SIMULATION
C EACH AREA SPECIFIED BY THE USER IS SIMULATED COMPLETELY AND THE
C RESULTS ARE STORED IN A TEMPORARY FILE
C ANY NUMBER OF AREAS CAN BE SIMULATED, BY SPECIFYING THEIR
C RELATION TO THE PREVIOUSLY SIMULATED AREAS (PARALLEL, SEQUENTIAL)

C SUBROUTINE FOR STARTING THE DETAILERED SIMULATION
SUBROUTINE DETAILSIM(IAREA,SXP,SYP)
INTEGER IDATE
REAL MPX(4),MPY(4)
REAL FXX(50,4),FRX(50,4),FFX(50,4),FFY(50,4)
CHARACTER*6 ARE, TYPE*1, AREA*10, FINAME*6
COMMON /PLR/IPILPA,T,IPILG
common /window/twin(4),swin(4)
COMMON /IFRIPAT/ICOLOR,IPATTERN,IFILES
COMMON /ALL/MPX,MPY,IR,FXX,FRX,FFX,FFY,FRY,FR,IP
COMMON /DET/ARE,TYPE,IPD
COMMON /DET2/IDLU,IDEF,AREFACT
common /base/npoint,ixa(300),yaa(300),net(300),nflag(300),
Indt(300),ndtt(300),x(2000),y(2000),mpt(2000),ipil
DIMENSION PLRXX(200,4),PLRRY(200,4),SXP(4),SYP(4)
DIMENSION CENL(7),CENFWIDTH(7),CENPCLEN(7),CENCONS(7),
1 CENCI(7),CENC2(7)
DIMENSION IID(4)

C THIS COMMON PASSES INFORMATION TO ACTSIM
COMMON/SIME/PLRXX,PLRRY,CENL,CENFWIDTH,CENPCLEN
1,CENCONS,CENCI,CENC2,IPILG
PRINT *, "CHOOSE ONE OF THE FOLLOWING"
PRINT *, "---------------------------------
PRINT *, "1 - AREA BY NODES"
PRINT *, "2 - AREA BY SECTION DEFINITION"
PRINT *, "---------------------------------
READ *,IAREA
IF (IAREA.EQ.1) THEN
PRINT *, "ENTER THE SECTION ID"
READ *,IPD
TYPE='L'
PRINT *, "---------------------------------
PRINT *, "ENTER THE SEQUENCE STARTING FROM"
PRINT *, "LOWER LEFT HAND CORNER (CLOCK WISE)"
PRINT *, "---------------------------------
READ(++)((IID,IDUM),IDUM=1,4)
DO IDUM=1,4
SXP(IDUM)=XI(IID,IDUM)
END
GAREA=SAREA(4, SXP, SYP)
AREAFACT=1.
GOTO 44
ENDIF
PRINT *, 'ENTER THE AREA, TYPE AND THE ID NO. OF THE SECTION'
PRINT *, 'TYPE B= BLEEDER P=DEVELOPNT L=LONGWALL'
PRINT *, 'FOR E.G. AREA,B,2';
READ (*,10) ARE
READ(*,12) TYPE
READ(*,14) IPD
10 FORMAT(A6)
12 FORMAT(A1)
14 FORMAT(I3)
IPILND=IPD
IF(TYPE .EQ. 'L') THEN
PRINT *, 'CALLING PANDRAW'
CALL PANDRAW(ARE)
DO IDUM=1,4
SXP(IDUM)=FPX(IPD, IDUM)
SYP(IDUM)=FPY(IPD, IDUM)
END DO
GAREA=SAREA(4, SXP, SYP)
AREAFACT=1.
ELSE
IF(TYPE .EQ. 'B') THEN
FINAME='BP.DAT'
PRINT *, 'READING BP.DAT'
ELSE
FINAME='RP.DAT'
PRINT *, 'READING RP.DAT'
ENDIF
OPEN(UNIT=12, FILE=FINAME, STATUS='OLD')
READ(12,505, END=600) ARE, IPN, FINAME
505 FORMAT(A6, I3, A10)
IF((ARE .EQ. AREA) .AND. (IPD .EQ. IPN)) THEN
OPEN(UNIT=15, FILE=FINAME, STATUS='OLD', ERR=512)
READ(15,*) MPI, MPI
READ(15,*) OFFDIS
READ(15,*) CENSLO
READ(15,*) CENFWIDTH
READ(15,*) CENPCLN
READ(15,*) CENCONS
READ(15,*) CENCI
READ(15,*) ICENCE
READ(15,*) IPC
DO 25 II=1, IPC-1
25 READ(15,*) ((PLRX(II, 12), PLRY(II, 12)), II=1, 4)
GAREA=0.
SXP(IDUM)=(PLRX(IDUM,1,IDUM))
SYP(IDUM)=(PLRY(IDUM,1,IDUM))
END DO
SAREA=SAREA+SAREA1(4,SXP,SYP)
END DO
DO IDUM=1,4
SXP(IDUM)=MPX(IDUM)
SYP(IDUM)=MPY(IDUM)
END DO
TOTAAREA=SAREA1(4,SXP,SYP)
AREAFACT=(TOTAAREA-SAREA)/TOTAAREA
SAREA=TOTAAREA-SAREA
CLOSE(UNIT=15)
CLOSE(12)
ELSE
GOTO 24
ENDIF

C THIS SECTION OF THE PROGRAM GIVES THE OPTION
C TO START AT THE DESIRED SIDE
CONTINUE
IPATTERN=0
DO IDUM=1,4
MPX1(IDUM)=SXP(IDUM)
MPY1(IDUM)=SYP(IDUM)
END DO
CALL PATTERN(4,MPX,MPY)
CALL CHRSIZ(2)
CALL MOVEA(MPX1(1),MPY1(1))
CALL OUTN(1)
CALL MOVEA(MPX1(1),MPY1(1))
DO 31 ICKL=2,4
CALL DRAWA(MPX1(ICKL),MPY1(ICKL))
CALL OUTN(ICKL)
CALL MOVEA(MPX1(ICKL),MPY1(ICKL))
CONTINUE
CALL DRAWA(MPX1(1),MPY1(1))
CALL TSEND
PRINT *,
PRINT *,"CHOOSE ONE OF THE FOLLOWING SIDES"
PRINT *,"FOR THE SIMULATION START"
PRINT *,"-------------
PRINT *,1 - LINE 1-2"
PRINT *,2 - LINE 2-3"
PRINT *,3 - LINE 3-4"
PRINT *,4 - LINE 4-1"
PRINT *,"-------------
READ *,ICHG
SIPI(1)=MXI(1)
SIPI(1)=MPY1(1)
SIPI(2)=MXI(4)
SIPI(2)=MPY1(4)
SIPI(3)=MXI(3)
SIPI(3)=MPY1(3)
SIPI(4)=MXI(2)
SIPI(4)=MPY1(2)
ENDIF
IF(ICH0 .EQ. 2) THEN
SIPI(1)=MXI(2)
SIPI(1)=MPY1(2)
SIPI(2)=MXI(1)
SIPI(2)=MPY1(1)
SIPI(3)=MXI(4)
SIPI(3)=MPY1(4)
SIPI(4)=MXI(3)
SIPI(4)=MPY1(3)
ENDIF
IF(ICH0 .EQ. 3) THEN
SIPI(1)=MXI(3)
SIPI(1)=MPY1(3)
SIPI(2)=MXI(2)
SIPI(2)=MPY1(2)
SIPI(3)=MXI(1)
SIPI(3)=MPY1(1)
SIPI(4)=MXI(4)
SIPI(4)=MPY1(4)
ENDIF
RETURN

600 CLOSE(12)
PRINT *, 'DATA NOT FOUND'
RETURN

612 PRINT *, 'DATA FILE NOT FOUND'
RETURN
END
C THIS SUBROUTINE OPENS THE DATA FILE FOR WRITING THE PLOTTER OUTPUT
C A COMMON FLAG SETS UP THE WRITING IN THE DRAPLOT AND MOV PLOT ROUTINES
C DEVELOPED BY SRIDHAR CHALUVADI FOR INTERFACING WITH
C HI PLO TTER FOR THESIS WORK ON ICAMPS
SUBROUTINE PLOTANS
CHARACTER *1 ANS, FNAME*10
COMMON /PLO TTER/ I PLOT, PFACT
COMMON /POL/IPOL
DATA I PLOT/0/
CLOSE(UNIT=21)
PRINT *, ' ' PRINT *,' PLOTTER OUTPUT?' PRINT *,' ' ---------------?' PRINT *,' DO YOU WANT PLOTTER OUTPUT (YES=1 NO=0)?'
READ (*,10) ANS
C 10 FORMAT(A1)
C IF(ANS .EQ. 'Y') THEN CALL YES_NO(IANS)
IF(ANS .EQ. 'N') THEN
15 print *, ' ENTER THE SCALE FACTOR (4,5...) ' READ (*,*,err=15) PFACT IPLOT=1 PRINT *, ' ' PRINT *, ' ENTER THE FILE NAME' READ(*,20) FNAME
20 FORMAT(A10)
C PRINT *, ' DO YOU WANT LH/RP OUTLINES (Y/N)?' READ (*,10) ANS
C IF(ANS .EQ. 'Y') THEN IPOL=1
C ELSE IPOL=0 C ENDIF OPEN(UNIT=21,FILE=FNAME,STATUS='NEW') WRITE(21,113)
113 FORMAT(IX,'I: H A ') ENDIF RETURN END
C SUBROUTINE FOR SETTING THE PLOTTER OUTPUT
SUBROUTINE MOV PLOT(A,B)
INTEGER I1, I1
INTEGER WIN(4)
COMMON /WINDOW/ WIN, SWIN(4)
C PFACT IS USED FOR HI PLO TTER SCALING
COMMON/ PLO TTER/ I PLOT, PFACT
DATA PFACT/6./
IF(IPLLOT .EQ. 0) RETURN

C selecting the current session window definition

C PRINT *, 'ENTERED CO-ORDS', A, B
A(1)=A-(SWIN(1)
B(1)=B-(SWIN(3)

C PRINT *, 'SUBTRACTED CO-ORDINATES', A1,B1
X1=FIX((A1/PFACT)+.5)
Y1=FIX((B1/PFACT)+.5)

C PRINT *, 'FACTORED CO-ORDINATES', X1,Y1
WRITE(1,610)X1,Y1

610 FORMAT(1X,'A ',I7,' ',I7,' ') RETURN

END

SUBROUTINE DRAPLOT(A,B)
C PFACT IS USED FOR HI PLOTTER SCALING
INTEGER X1,Y1
COMMON/PLTTER/IPLLOT,PFACT
INTEGER TWIN(4)
COMMON /WINDOW/TWIN,SWIN(4)

C DATA PFACT/6.25/
IF(IPLLOT .EQ. 0) RETURN

C PRINT *, 'ENTERED CO-ORDS', A, B
A(1)=A-(SWIN(1)
B(1)=B-(SWIN(3)

C PRINT *, 'SUBTRACTED CO-ORDINATES', A1,B1
X1=FIX((A1/PFACT)+.5)
Y1=FIX((B1/PFACT)+.5)

C PRINT *, 'FACTORED CO-ORDINATES', X1,Y1
WRITE(1,620)X1,Y1

620 FORMAT(1X,'D ',I7,' ',I7,' ') RETURN

END

C SUBROUTINE FOR DRAWING A POLYGON OF N POINTS
SUBROUTINE POLY(X,Y)
COMMON /POL/IPOL
DIMENSION X(10),Y(10)
CALL MOVEPOL(X(1),Y(1))

C IF(IPOL .EQ. 0)THEN

C CALL DRAPLOT(X(2),Y(2))
C CALL MOVEPOL(X(3),Y(3))
C CALL DRAPLOT(X(4),Y(4))

C RETURN
C ENDIF

DC 20 IO=8,N
N0 CALL DRAPLOT(X(I),Y(I))
CALL DRAPLOT(X(1),Y(1))
RETURN
END
SUBROUTINE FOR WRITING TEXT ON THE PLOTTER
SUBROUTINE TEXTPLOT(CHX,CHY,ICK)
COMMON/PLT/PLT,PFACT
IF(PILOT .EQ. 0)RETURN
CALL MOV PLOT(CHX,CHY)
WRITE(21,20)ICK
20 FORMAT(1X, 'S16,NI,GO,XI,YI,14,  _ ') RETURN
END

THIS SUBROUTINE WRITES TWO LABELS AT A SPECIFIED ANGLE
THIS IS CALLED FROM THE SIMULATION ROUTINE
SUBROUTINE TEXTPLOT(CHX,CHY,LABEL,LA,ANG1)
COMMON/PLT/PLT,PFACT
IF(PILOT .EQ. 0)RETURN
IY=INT(100.*TAN(ANG1))
CALL MOV PLOT(CHX,CHY)
WRITE(21,20)IY,LAB L A
20 FORMAT(1X, 'S16,NI,GO,XI,YI,16,  _ ') RETURN
END

SUBROUTINE FOR WRITING A LINE OF TEXT AT A SPECIFIED POSITION
SUBROUTINE TEXTPLOT(CHX,CHY,LABEL,LENGTH,ANG1,SISE)
CHARACTER *90, LABEL(1)
CHARACTER *2, SIZE(1), DSIZE
COMMON/PLT/PLT,PFACT
IF(PILOT .EQ. 0)RETURN
DSIZE='8'//SIZE(1)
IF(ABS(ANG1) .EQ. 90)ANG1=ANG1+1.
IY=INT(100.*TAN(ANG1))
CALL MOV PLOT(CHX,CHY)
WRITE(21,20)DSIZE,IY,LABEL
20 FORMAT(1X, 'S16,AB,NI,GO,XI,YI,16,  _ ')
RETURN
END

SUBROUTINE TEXTPLOT(CHX,CHY,DIR,LABEL,ANG1)
COMMON/PLT/PLT,PFACT
BYTE DIR(I)
IF(PILOT .EQ. 0)RETURN
IY=INT(100.*TAN(ANG1))
CALL MOV PLOT(CHX,CHY)
WRITE(21,20)IY,DIR,LAB L A
20 FORMAT(1X, 'S16,NI,GO,XI,YI,16,  _ ') RETURN
END
SUBROUTINE RMOVPLT(A,B)
 INTEGER X,Y
 C FACT IS USED FOR HI PLOTTING SCALING
 COMMON/ PLOTT/ I PLOT, PFACT
 C DATA PFACT/ &/
 IF(IPLOT .EQ. 0) RETURN
 X=IFIX((A/PFACT)+.5)
 Y=IFIX((B/PFACT)+.5)
 WRITE(1,610)X,Y
 610 FORMAT(1X,'U R ',17,/,17,/) RETURN
 END

SUBROUTINE RDRAPL T(A,B)
 INTEGER X,Y
 C PFACT IS USED FOR HI PLOTTING SCALING
 COMMON/ PLOTT/ I PLOT, PFACT
 C DATA PFACT/ &/
 IF(IPLOT .EQ. 0) RETURN
 X=IFIX((A/PFACT)+.5)
 Y=IFIX((B/PFACT)+.5)
 WRITE(1,610)X,Y
 610 FORMAT(1X,'D R ',17,/,17,/) RETURN
 END

DUA0:(SR1PLOTSUP.FOR? ; type? [H];
C TITLE BLOCK DEVELOPED FOR AEP
C PROGRAMMED BY SRIDHAR CHALUVADI JAN 1986
C
SUBROUTINE(SX,SY)
IMPLICIT INTEGER (-A-Z)
REAL SX1,SY1,SWIN
COMMON /WINDOW/TWIN(4),SWIN(4)
DATA TWIN,SWIN/*0,0,4*0,0/
CHARACTER *30,TEXT(1)
BYTE BYS(80)
C DATA ANGI/1,/
CALL PLOTANS
PRINT *,"ENTER THE X AND Y VALUES"
READ *,SX,SY
C LENGTH AND HEIGHT ARE SET
C LENGTH=1000
C HEIGHT=500
L=1000
H1=500
SX1=SX
SY1=SY
C DRAWING THE BLOCK
CALL TMOVE(SX,SY)
CALL MOV PLOT(SX1,SY1)
CALL TDRAW(SX,(SY+H1))
CALL DRAPLOT((SX1),(SY+H1))
CALL TDRAW((SX1+L),(SY+H1))
CALL DRAPLOT((SX1+L),(SY+H1))
CALL TDRAW((SX1+L),(SY))
CALL DRAPLOT((SX1+L),SY1)
CALL TDRAW(SX,SY)
CALL DRAPLOT(SX1,SY1)
C TOP LINE
H2=500
CALL TMOVE(SX,(SY+H2))
CALL MOV PLOT(SX1,(SY1+H2))
CALL TDRAW((SX1+L),(SY+H2))
CALL DRAPLOT((SX1+L),(SY1+H2))
C BOTTOM LINE
H3=100
CALL TMOVE(SX,(SY-H3))
CALL MOV PLOT(SX1,(SY1+H3))
CALL TDRAW((SX1+L),(SY+H3))
CALL DRAPLOT((SX1+L),(SY1+H3))
C VERTICAL LINE
CALL TMOVE((SX1+L/2),SY)
CALL MOV PLOT((SX1+L/2),SY1)
CALL TDRAW((SX1+L/2),(SY+H3))
CALL DRAPLOT((SX1+L/2),(SY1+H3))
C WRITING THE TEXT
C TOP TWO LINES SIZE 3, WI=30, HT=20, SPA=10
WI=30
SPA=10
IAN=1
C CALL GROT(IANG1)
C CALL GSIZE(I)
C TEXT(1)= 'AMERICAN ELECTRIC POWER'
TEXT(1)= 'ISLAND CREEK'
CALL OUTCHAR(TEXT(1),L, WI, SPA, SX, SY, 400)
C TEXT(1)= 'FUEL SUPPLY DEPARTMENT'
TEXT(1)= 'COAL SUPPLY DEPARTMENT'
CALL OUTCHAR(TEXT(1),L, WI, SPA, SX, SY, 300)
CALL GSIZE(I)
WI=15
SPA=10
PRINT *, 'ENTER THE FIRST LINE OF TEXT'
READ(*,20) TEXT(1)
20 FORMAT(A80)
CALL OUTCHAR(TEXT(1),L, WI, SPA, SX, SY, 240)
PRINT *, 'ENTER THE SECOND LINE OF TEXT'
READ(*,20) TEXT(1)
CALL OUTCHAR(TEXT(1),L, WI, SPA, SX, SY, 190)
PRINT *, 'ENTER THE THIRD LINE OF TEXT'
READ(*,20) TEXT(1)
CALL OUTCHAR(TEXT(1),L/2, WI, SPA, SX, SY, 130)
PRINT *, 'ENTER THE DATE'
READ(*,20) TEXT(1)
CALL OUTCHAR(TEXT(1),L/2, WI, SPA, SX, SY, 40)
PRINT *, 'ENTER THE SCALE INFO'
READ(*,20) TEXT(1)
CALL OUTCHAR(TEXT(1),L/2, WI, SPA, (SX+L/2), SY+40)
STOP
END

SUBROUTINE OUTCHAR(TEXT,L, WI, SPA, SX, SY, H)
IMPLICIT INTEGER(A-Z)
REAL SX, SY
CHARACTER *60 TEXT(1)
CHARACTER*2, SIZE(1)
BYTE BYS(80)
SX= SX
SY= SY
CALL COUNTCHAR(KOUNT,TEXT(1))
CALL LSTART(L, WI, SPA, KOUNT, LSTART)
CALL TMOVE((SX+START),(SY+H))
ENCODE (KOUNT,10,BYS(80))
FORMAT(A(KOUNT))
CALL 3TEXT1(KOUNT, BYS)
PRINT *, 'ENTER THE SIZE OF THE PLOT LETTERS'
READ(*,15) SIZE(1)
15 FORMAT(A2)
   IANGLE=0
   CALL TXTPLT((SX1+START),(SY1+H),TEXT(1),KOUNT,IANGLE,SIZE(1))
   RETURN
END

SUBROUTINE LSTART(L,W1,SPA,KOUNT,START)
 IMPLICIT INTEGER(A-Z)
 CHARACTER *80 TEXT(1)
 BYTE 3YS(80)
 START=(L-(KOUNT*(W1+SPA)))/2
 RETURN
END