

HYDRODAT - A COMPUTERISED STORAGE AND RETRIEVAL

SYSTEM FOR SEISMIC SHOT-HOLE DATA

BY

R. BIBBY, W. NEILSON, J. ORLECKY

INTERNAL REPORT,
GROUNDWATER DIVISION,
ALBERTA RESEARCH,

MAY, 1974

Alberta
RESEARCH COUNCIL



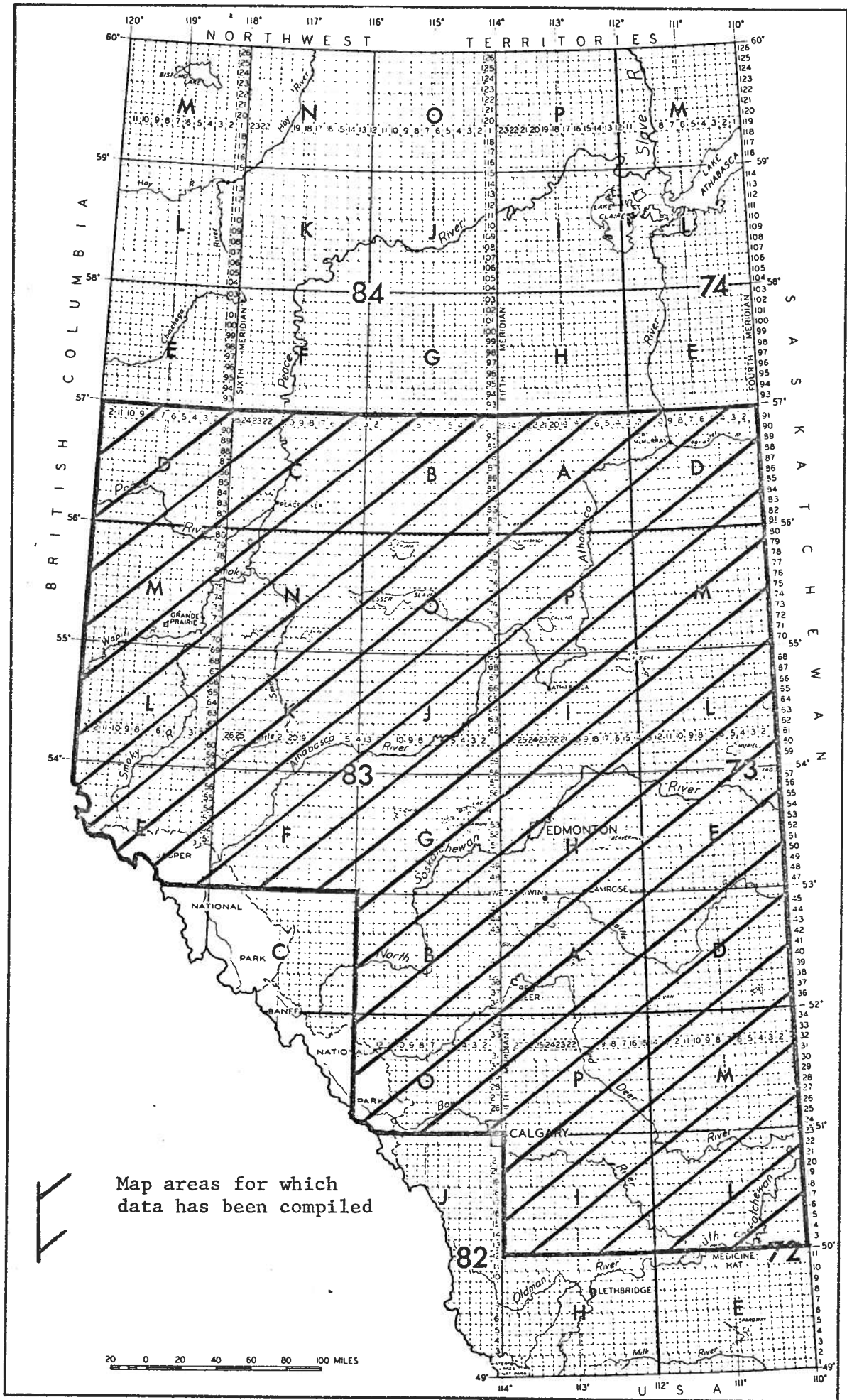
INTRODUCTION

Hydrodat is a computer oriented system for storing and retrieving seismic shot-hole data. The system was initiated in 1967 by V. A. Carlson¹, who designed it and supervised its operation until the end of 1973. Most of the work concerned with compiling the data was done by L. Withers and J. Orlecky. The computer programs were written by John Stasiuk at Computing Services, University of Alberta.

The objective of this report is to describe the system so that it may readily be used. The report consists of four sections: a description of how the basic data is compiled, a description of how new data is entered into the database, a description of how to make a retrieval from the database and a list of computer programs.

The Hydrodat system was established to facilitate the handling of large quantities of shot-hole data for use in the construction of bedrock topography maps of the Province. All the data entered in the system was obtained directly from oil companies. For any shot-hole, the location, surface elevation, drift thickness, lithology, "Blind" zones and "Flowing" zones are coded. Figure 1 shows the parts of the Province for which data has so far been compiled.

1 Research Officer, Alberta Environment, Formerly Research Officer, Alberta Research.



COMPILATION OF DATA FOR THE HYDRODAT SYSTEM

The original motive for compiling the seismic shot-hole data was to aid in the development of bedrock topography maps (NTS 1:125000). For this reason data was obtained for one map-sheet at a time. The following description of the procedure for compiling the data is therefore given with reference to a single map-sheet.

The locations of all shot-holes drilled in the Province are reported to Alberta Mines and Minerals. Oil companies conducting seismic surveys submit township plats on which are plotted shot-hole locations. These plats are kept on file. The first step in the compilation process is therefore to obtain from this file the names of oil companies which have done seismic work in the map area, the specific locations of this work and the year that it was done. This information is recorded on a township-basemap of the area (see Fig. 2). One basemap is used for each oil company and the year in which the company worked in a specific township is recorded.

A composite map (see Fig. 3) is then made on which is recorded, on the same township-basemap, the total number of companies which have made seismic surveys in each township. Colours are then used to distinguish areas of good, average and poor coverage.

A control sheet (see Fig. 4) is then made on which is listed by name all the companies which have operated in the map area. For each company, the total number of townships, in which it operated, is computed and the list is organised in decreasing order of magnitude.

Since not all the companies on the list will necessarily be contacted, those having done most work in the area are approached first, but

OIL COMPANY

Imperial Oil

															57
		61 62 63 64													56
								61							55
										61 63					54
				63											53
									61 62						52
					61 62										51
									61 62 63 64						50
															49
															48
															47
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

4

Figure 2 Township-Basemap of Mapsheet 83E

OIL COMPANY _____

															57
															56
															55
						11111 11111									54
										1111					53
															52
															51
										1111 111					50
															49
															48
															47
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Figure 3 Composite Map

CONTROL SHEET

Oil Company	No. Tps. Shot	Contacted	Received	Completed
Shell Oil	98	May 2	May 23	Yes
Imperial	44	June 5	June 25	Yes
Sun Oil	3	no	-----	---

Figure 4 Control Sheet

care is taken to obtain as much data as possible in the areas of poor coverage (as defined from the composite map (Fig. 3)).

When a company is contacted the exact townships for which data were obtained are recorded on a map (see Fig. 5). A separate map is used for each company. Not all the data submitted by a company is necessarily coded. For this reason, the exact townships which are coded are recorded on Fig. 2. The collection of maps similar to Fig. 5 is called the project book. The collection of maps similar to Figures 2, 3, 4 is called the coverage book.

Companies are contacted by letter requesting data. Copies of letters sent to companies in the past, together with replies, are kept on file.

The dates of the request letters and the reply letters are recorded on the control sheet.

Information is most often submitted by companies in the form of books of drillers logs and shot-hole location maps. After transcribing, the books are returned. Some companies require the data to be transcribed in their office. In either case, the date of completion is marked on the control sheet.

All shot-hole logs are transcribed onto the autoplot coding form (see Fig. 6). From this form the data is punched onto regular punch-cards which have 80 columns. Only the first 76 columns are used. Columns 1 to 27 must be filled in on every form. 16 items of information are coded on each form.

Item 1, Col.1:	Meridian
Item 2, Col. 2-4:	Township
Item 3, Col. 5-6:	Range
Item 4, Col. 7-9:	North - distance in miles north of S.E. corner of township

Item 5, Col. 10-12: West - distance in miles west of S.E. corner of township.

For items 4 and 5 the distance is always recorded to two decimal places. For example, 2.20 miles is coded as 220. These distances are read off the shot-hole location map supplied by the oil company.

Item 6, Col. 13-16: Surface Elevation - elevation of the shot-hole in feet above sea level. This is usually given on the shot-hole location map or in the survey book which accompanies the log books and location maps. Failing this, it must be read from a 1:50000 topographic map.

Item 7, Col. 17-19: Oil Company Number - each oil company has a unique code number which may be found listed in the company code book and on page 20 of this report.

Item 8, Col. 20-23: Driller - drillers names are no longer reported so that zero's are coded.

Item 9, Col. 24: "1" is coded if bedrock was encountered, "2" if it was not encountered. It is assumed that bedrock was encountered if any of the following lithologies were reported:

Shale	Shale + Ss stringers
Sandstone	Shale + Coal
Coal	Ss + Coal
Sh + Ss	

The following lithologies are all drift deposits:

Gravel	Sand
Gravel + Sand	Clay

Item 10, Col. 25-27: Footage - this is the thickness of the drift in feet if bedrock was encountered or it is the depth of the hole if bedrock was not encountered.

Item 11,12 Col. 28-62: Lithology - each lithology has a code which is given on the coding form. There is space for up to 5 different lithologies to be coded. The code number for a lithology is entered in columns 34, 41, 48, 55 and 62. The depth intervals at which these lithologies occur are coded in the columns opposite the lithology code.

Item 13,14 Col. 63-69: Blind - if a blind zone is reported by the driller then "1" is coded in column 63 if the depth interval is not given and "2" if the depth interval is given. In the latter case the depth interval is coded in columns 64-69.

Item 15,16 Col. 70-76: Flowing - same procedure as for blind.

The following points should be noted:

1. Any hole less than 50' is not to be coded unless surface sand or gravel is reported or bedrock is reached.
2. Bedrock = shale, sandstone, coal or any combination.
3. If a shot hole is drilled to 60 feet and no sand, gravel or bedrock is encountered the hole is of marginal usefulness and may be coded one shot hole per mile.
4. Cemented sand may be coded as sandstone if bedrock is reached in that area at about the same depth.
5. Hard brown clay is coded as shale.
Hard blue clay is coded as shale.

6. Sand and sandstone may be coded as either depending on the location of bedrock or sand depths.
7. All blind and flowing shot holes must be coded.

UPDATING THE HYDRODAT STORAGE SYSTEM

The database is stored on 3 magnetic tapes at the U. of A. At any one time, only one tape is completely up to date (Tape A), another tape contains the database prior to the last update (Tape B) and the third tape contains the database prior to the last two updates (Tape C). The database on each tape is strictly ordered, firstly by meridian, secondly by township, thirdly by range, fourthly by north and fifthly by west. It is of paramount importance that, when an update is made, the new cards are ordered in exactly the same manner.

Running an update results in the data being copied from tape A to tape C with new data being inserted in the appropriate place. Thus, tape C would become the most up-to-date tape, tape A becomes the database prior to the last update and tape B becomes the database prior to the last two updates. The next update would copy the database from tape C to tape B making the appropriate insertions. Updating in this cyclic manner minimises the chances of destroying the entire database. It does however necessitate keeping careful note of the current status of each tape and making sure that when running an update the correct tape numbers are used.

The Hydrodat Storage System has an updating program that has the capability of creating a new database, adding (inserting) new records into the existing database, deleting existing records, or listing all of the records in the existing database.

The program is in a file in the Computing Science Account CARA, and can be invoked using the following commands (for password and tape numbers see Bibby or Redberger).

\$SIGNON CARA 9TP=2 P=100 T=5M

PASSWORD -

\$MOUNT

**** 9TP *IN* VOL = T***** BLK=OFF

\$ENDFILE

\$RUN W025: TAKEØS PAR=E=MAPPING.UPDA+NEW: COBLIB, IO; SYSUT3=-
IN @ FB 92000,80) SYSUT4=*OUT* @ FB (200,80)

UPDATE (or CREATE or LIST)

CARDS TO BE INSERTED - CORRECTLY ORDERED

\$ENDFILE

\$SIGNOFF

NOTES

1. The UPDATE card will insert new records into the database.
The CREATE card will create a database from the new records.
The LIST card lists all the records in the database.
2. The records to be inserted must be in the form outlined in the BEDROCK TOPOGRAPHY DATA coding form. Furthermore, they must be ordered as described above. As well, the following checks are made:
 1. Any non-numeric data (except blanks).
 2. Township range 001 to 126.
 3. Range range 01 to 30.
 4. North distance range 000 to 599.
 5. West distance range 000 to 599.
 6. Surface elevation range 0450 to 9999.
 7. Column 24 is 1 or 2.
 8. LITH-codes (columns 34, 41, 48, 55, 62) must be in range 0 - 9.

9. Column 63 is in 0-2 range.
10. Column 70 is in 0-2 range.

Any of the above errors will cause the record to be rejected.

3. Records can be deleted during an update run by having columns 1 - 12 match an existing record in the database, and coding DELT in columns 77 - 80. DELT is invalid if the run type is CREATE.
4. All input records, whether inserted, deleted, or rejected, are listed on the line printer with an appropriate message.
5. Note that the old database tape is called *IN*, and the new one called *OUT*: also, RING = IN must be specified on *OUT*.
6. Wells to be inserted that have a location identical to a well already in the database will be rejected.

RETRIEVING FROM THE HYDRODAT SYSTEM

The Hydrodat Retrieval System can be used to obtain printed output, plotted output or both of a selected subset of wells in the database. This is done by submitting batch runs for the university computer running programs stored in the account CARA (see Bibby or Redberger).

DESCRIPTION OF FILES USED FOR RETRIEVAL

There are four files in CARA which are used to retrieve information.

They are:

1. RET1 - runs the search
2. RET2 - defines the type of search
3. RET3 - sets up plot file
4. PLOT - does the plotting

RET1

This file has the following general form:

LINE

- 1 Signon
- 2 Create - A - Temporary Storage File
- 3 Create - B - Temporary Storage File
- 4 Mount tape - Tape containing database
- 5 Run WØ25. etc. - Searches the tape and retrieves the desired records.

It should never be necessary to change this file, except possibly for the parameters on the signon card.

RET2

This is the most important file from the point of view of the user since it contains the parameters specifying the retrieval required. It has the following general form:

RET2 (cont'd)

LINE

- 1 Title - Any character string. This title appears on the printout.
- 2 Type of Retrieval- Parameter specifying whether retrieval is to be printed, plotted etc.
- 3
- .
- .
- .
- Retrieval Criteria-Values defining retrieval required.
- .
- .
- .
- n
- n+1 20 - Signifies end of retrieval criteria.
- n+2 Plot Criterion - Defines parameter to be plotted.
- n+3 Release tape - Releases tape mounted by RET1.

RET2 is always run concatenated after RET1, therefore there is no need for signon cards. Also, RET3 may be concatenated after RET2, if plotted output is needed.

On line 2, Type of Retrieval, a short or long printed output may be specified. The short output will give a listing of all the information on every well retrieved. The long output will give the same listing as the short plus a listing of the number of wells in the whole database in every township in the Province. The following options are possible for line 2:

- PRINT - retrieval is printed only. Get long output.
- PLOT - retrieval is plotted only.
- BOTH - retrieval is printed and plotted. Get long listing.
- PRINT SHORT - retrieval is printed only. Get short listing.
- PRINT LONG - retrieval is printed only. Get long listing.

BOTH SHORT - retrieval is printed and plotted. Get short listing.

BOTH LONG - retrieval is printed and plotted. Get long listing.

The parameter used must start in column 1 and a single space must be left between words. If a plot is requested then RET3 must be concatenated after RET2.

The possible retrieval criteria, which appear on lines 3 to n, are summarised below.

<u>CODE NUMBER</u>	<u>ASSOCIATED MEANING</u>	<u>FIELD WIDTH</u>
01	Meridian	1
02	Township	3
03	Range	2
04	Distance North from S.E. corner of Tnship.	3
05	Distance West from S.E. corner of Tnship.	3
06	Surface Elevation	4
07	Oil Company	3
08	Driller	Not Used
09	Drift Thickness	1
10	Drift Thickness (footage)	3
11	Lithology (depth range)	3
12	Lithology (type)	1
13	Blind	1
14	Blind (depth interval)	3
15	Flowing	1
16	Flowing (depth interval)	3
17	Bedrock elevation	3

Each retrieval criterion employed by the user appears on a separate line in RET2. Each line has the following format:

COLUMN

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
x	x									x	x	x	x							x	x	x	x
Criterion Code Number											Lower Limit of Criterion											Upper Limit of Criterion	

The number of characters specifying the lower and upper limits of the criterion must be the same as the field width on the above table.

Retrieval criterion 09, Drift Thickness has two possible values of lower and upper limit. "1" means bedrock encountered and "2" means bedrock not encountered. Thus, for example, by coding "1" for both the lower and upper limit only wells penetrating to bedrock would be retrieved.

Retrieval criterion 12, Lithology Type, is used to retrieve wells having a specific lithology. Lithologic types used in the database are:

<u>CODE NUMBERS</u>	<u>LITHOLOGY</u>
1	- Gravel
2	- Sand + Gravel
3	- Sand
4	- Coal
5	- Sandstone
6	- Shale + Sandstone
7	- Shale + Sandstone Stringers
8	- Shale and Coal
9	- Sandstone and Coal

The code number of a lithologic type is entered as the upper and lower limit. Thus, for example, by setting the lower limit equal to 1 and the upper limit equal to 3, all wells having Gravel and/or Sand + Gravel and/or Sand would be retrieved.

Retrieval Criterion 07, oil company, is used to retrieve wells drilled by a particular oil company. The following is a list of the oil companies and the code numbers that are used as lower and upper limits.

<u>CODE NUMBER</u>	<u>OIL COMPANY</u>	<u>CODE NUMBER</u>	<u>OIL COMPANY</u>
001	Imperial Oil	044	Seafort Petroleum
002	Texaco	045	Siebens Oil & Gas
003	Gulf Oil	046	Skelly Oil
004	Amoco Canada	047	Western Decalta
005	Shell Canada	048	Sohio Petroleums
006	Union Oil	049	Pubco Canadian
007	BP Oil & Gas	050	Chieftain Development
008	Mobil Oil	051	Apache Corporation
009	Sun Oil	052	Jefferson Lake Petrochemicals
010	Tenneco Oil	053	Phillips Petroleum
011	Canadian Pacific	054	Calgary Power
012	Atlantic Richfield	055	Northern Explorers
013	Husky Oil	056	Barnsdall Oil Company
014	Hudson's Bay	057	Canadian Homestead
015	Home Oil		
016	Great Plains Development		
017	Scurry Rainbow		
018	Murphy		
019	Leonard Refineries		
020	Dome Petroleum		
021	Amerada Hess		
022	Central Del Rio Oils		
023	Canadian Superior		
024	Banff Oils		
025	Chevron Standard		
026	Altana Explorations		
027	Canada Cities and Service		
028	Water Well Logs		
029	Luscar Coal		
030	Water Resources		
031	Canadian Export Gas & Oil		
032	Canadian Industrial Gas & Oil		
033	Midwest Oil		
034	Teledyne Exploration		
035	Western Minerals		
036	Texas Gulf Sulphur		
037	Canadian Fina		
038	Pure Oil		
039	Fargo Oils		
040	Colorado Oil & Gas		
041	Monsanto Oils		
042	Pacific Petroleums		
043	Royalite Oil		

Retrieval Criterion 08, Driller, is always entered in the file as 000 and is not used for retrievals.

Retrieval Criterion 13, Blind, has 3 possible values of lower and upper limit. "0" means that no blind zone was reported by the driller. "1" means that a blind zone was encountered but the depth at which it occurred was not reported. "2" means that a blind zone was encountered and the depth at which it occurred was reported. Thus, for example, setting the lower limit equal to "1" and the upper limit equal to "2", all wells in which a blind zone was encountered would be retrieved.

Retrieval Criterion 15, flowing, has the same 3 possible values of lower and upper limit as 13 and they have exactly analogous meanings.

The following is a list of examples of the use of each retrieval criterion:

<u>COLUMN</u>										<u>WELLS RETRIEVED</u>						
1	2	.	.	.	10	11	12	13	.	.	.	20	21	22	23	
0	1				4							5				All wells west of 4th and 5th meridians
0	2				0	4	0					0	5	1		All wells in Townships 40 to 51 inclusive
0	3				0	6						2	8			All wells in ranges 6 to 28 inclusive
0	4,				0	0	0					3	0	0		All wells less than 3 miles north of S.E. corner of Township
0	5				3	0	0					6	0	0		All wells from 3 to 6 miles west of S.E. corner of Township
0	6				2	0	0	0				2	5	0	0	All wells having surface elevation from 2000' to 2500'
0	7				0	0	1					0	0	1		All wells of Oil company 1
0	8															Not used
0	9				2							2				All wells not penetrating bedrock
1	0				1	0	0					9	9	9		All wells having drift thickness from 100 to 999 ft.
1	1				0	5	0					1	0	0		All wells with lithology recorded for interval 50 to 100 ft.
1	2				4							4				All wells having recorded coal occurrence.
1	3				1							2				All wells having blind zone

<u>COLUMN</u>										<u>WELLS RETRIEVED</u>						
1	2	.	.	.	10	11	12	13	.	.	.	20	21	22	23	
1	4				0	0	0					1	0	0		All wells having base of blind zone in top 100 ft.
1	5				1							2				All wells having flowing zone
1	6				0	0	0					1	0	0		All wells having base, flowing zone in top 100 ft. of
1	7				1	9	0					2	5	0		All wells having bedrock elevation from 1900' to 2500'

The retrieval criteria can be used in any combination to obtain specific retrievals. The wells retrieved are those which satisfy all the specified criteria.

Line n+1 in RET2 is always coded as 20. It signifies the end of the retrieval criteria.

On line n+2, Plot Criterion, is coded a value determining which parameter is to be plotted. It is used only if a plot is requested on line 2, otherwise it is ignored. Only one plot can be made at a time. If this is used, RET3 must be concatenated after RET2.

The number coded as the Plot Criterion can have a value of 01 to 17. These numbers have exactly the same meaning as the retrieval criteria code numbers. Thus, for example, if 17 is coded as the plot criterion the bedrock elevations would be plotted. The following points should be noted:

1. If the plot criterion is set as 11 then;
 - (a) if 11 was set as a retrieval criterion but 12 was not, the value plotted will be the depth to the bottom of the deepest lithologic unit in the interval defined by the lower and upper limits coded with retrieval criterion 11.
 - (b) If both 11 and 12 were set as retrieval criteria, then the value plotted will be the depth to the bottom of the deepest lithologic

unit specified by retrieval criterion 12 which is in the interval defined by the lower and upper limits coded with retrieval criterion 11.

- (c) if both 11 and 12 were not set as retrieval criteria then the value plotted will be 000. Either 11 or 12 must be specified in the retrieval criteria in order to get an accurate value.
 - (d) if 11 was not set as a retrieval criterion but 12 was, then the value plotted will be the depth to the bottom of the deepest lithologic unit specified by retrieval criterion 12.
2. In general, the value plotted will be exactly the same as that coded on the original coding form and will have the same field width (for criteria 14 and 16 where depth interval is coded, the bottom of the interval is plotted).

RET2 has to be edited using the MTS line editor prior to making a retrieval.

RET3

This file has the general form:

LINE

- 1 Mount tape - Contains values to be plotted
- 2 Sort - Sort the plot file for optimal plot time
- 3 Signoff -

RET3 is only used concatenated after RET2, therefore, there is no need for signon cards. It should never be necessary to change RET3.

PLOT

File PLOT has the general form:

LINE

- 1 Signon
- 2 Password
- 3 Create - D - file for digital plot commands
- 4 Mount tape - contains values to plot
- 5 Run WØ25: etc - fills - D with digital plot commands
- 6 Run *CALCOMPQ - queues plot
- 7 Signoff

It should never be necessary to change PLOT, except possibly for the parameters on the signon card.

PLOT is always run by itself, after a retrieval has been made by RET1, RET2 and RET3.

Retrieved data is sorted in townships and each township is plotted separately at a scale of 1 inch to 1 mile.

DESCRIPTION OF HOW TO RUN THE RETRIEVAL FILES

The following examples demonstrate how to make a retrieval for both printed and plotted outputs.

EXAMPLE 1

To retrieve all wells in Tp. 40 to 54 inclusive, Rge. 21 to 26 inclusive, west of the 4th mer. obtaining only a long printed output.

The necessary file set-up is as follows. RET2 has to be set-up using the MTS line editor. The parameters on the signon card may need to be changed from those given below. For information on the password and the tape volume serial number see Bibby or Redberger.

```
RET1 { $SIG CARA 9TP=1 T=5M P=300 PRIO =L RETURN =EDUC
      $CREATE - A TYPE=SEQ SIZE=20T
      $CREATE - B TYPE=SEQ SIZE=80T
      $MOUNT **** 9TP *T* VOL=***** BLK=OFF
      $RUN W025:TAKE0S PAR=E=RETOBJ+NEW: COBLIB, I0; SYSUT4=*T*@FB (2000,80) -
      SYSUT8=-A @FB (2000,20) SYST3 =-B @FB (6600,132)

RET2 { TITLE
      PRINT
      01      4      4
      02      040    054
      03      21     26
      20
      $RELEASE *T*
```

Having set-up these files submit a batch run as:

```
$RUN *BATCH SCARDS=RET1+RET2
```

The computer will prompt for the password.

Note that the "RETURN=EDUC" parameter on the signon card will have your output delivered to the 3rd floor of the New Education Building.

EXAMPLE 2

To retrieve all flowing wells and obtain a short printed output and plot.

The necessary file set-up is as follows. RET2 has to be set-up using the MTS line editor. The parameters on the signon card may need to be changed from those given below. For information on the password and tape volume serial numbers see Bibby or Redberger.

```
RET1 { $SIG CARA 9TP=1 T=5M P=300 PRIO=L RETURN=EDUC
      $CREATE-A TYPE=SEQ SIZE=20T
      $CREATE-B TYPE=SEQ SIZE=80T
      $MOUNT **** 9TP *T* VOL=***** BLK=OFF
      $RUN WØ25: TAKEØS PAR=E=RETOBJ + NEW: COBLIB, IØ; SYSUT4=*T* @ FB(2000,80)-
      SYSUT8=-A @ FB (2000,20) SYST3 =-B @ FB (6600,132)

RET2 { TITLE
      BØTH
      15          1          2
      20
      15
      $RELEASE *T*

RET3 { $MØUNT **** 9TP *P* RING=IN VØL=***** BLK=ØFF
      $RUN *SØRT PAR=SORT=CH; A; 6; CH; A; 13; 3 INPUT=-A; FB; 20;2000-
      ØUTPUT = *P*; FB: 20; 800 MNR=100000
      $SIGNØFF
```

Having set up these files submit a batch run as:

```
$RUN *BATCH SCARDS = RET1 + RET2 + RET3
```

The computer will prompt for the password

Note that the "RETURN=EDUC" parameter on the signon card will have your output delivered to the 3rd floor of the New Education Building.

After checking the output to make sure the run was executed correctly, set-up the following file to get the plot. The parameters on the signon card may need to be changed from those given below. For information on the password and tape volume serial number see Bibby or Redberger.

PLOT {

```
$SIG CARA T=4M PRIØ=L 9TP=1 RETURN=EDUC
PASSWORD
$CREATE -D TYPE=SEQ SIZE=20T
$MOUNT **** 9TP *P* VØL=***** BLK=ØFF
$RUN WØ25: TAKEØS 9=-D PAR=E=MAP.PLOT +* PLOTLIB; IO; FTØ1FØØ1=-
*P* @ FB (800,20)
$RUN *CALCØMPQ SCARDS =-D PAR=DELIVERY = "EDUCATION"
$SIGNØFF
```

This file is executed by submitting a batch run as:

```
$RUN *BATCH SCARDS = PLOT
```

The output will be delivered to the New Education Building.

EXAMPLE 3

To retrieve bedrock elevations on all wells in Tp. 1 to 10 inclusive, Rge. 1 to 16 inclusive, west of the 4th meridian and obtain a plotted output.

The necessary file setup is exactly the same as in Example 2 with the following modifications in RET2:

- (a) line 2 should be PLOT instead of BOTH
- (b) line 3 should be replaced by the following four lines:

01	4	4
02	001	010
03	01	16
17	000	999

(c) line 5 should be replaced by 17

It should be noted that by setting the lower and upper limits on retrieval criterion 17, Bedrock Elevation, at 000 and 999 respectively, bedrock elevations at all depths are retrieved.

LISTING OF COMPUTER PROGRAMS

\$C UPDATE

\$COMMENT UPDATE PROGRAM FOR RETRIEVAL PROGRAM

IDENTIFICATION DIVISION.

PROGRAM-ID. 'MAPPING'.

AUTHOR. NUCLEUS.

INSTALLATION. U OF A COMPUTING CENTER.

DATE-WRITTEN. JULY 1967.

DATE-COMPILED. JULY 1967.

SECURITY. CONFIDENTIAL.

REMARKS. MAINTENANCE PROGRAM FOR LITHOLOGY MAPPING.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IEM-360.

OBJECT-COMPUTER. IBM-360.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT PRINTER ASSIGN TO 'SYSOUT' UNIT-RECORD.

SELECT TRANS ASSIGN TO 'SYSIN' UNIT-RECORD.

SELECT OLD-MASTER ASSIGN TO 'SYSUT3' UTILITY.

SELECT NEW-MASTER ASSIGN TO 'SYSUT4' UTILITY.

DATA DIVISION.

FILE SECTION.

FD PRINTER RECORD CONTAINS 133 CHARACTERS

LABEL RECORDS ARE OMITTED

RECORDING MODE IS F

DATA RECORD IS PRINT-REC.

01 PRINT-REC.

07 FILLER PICTURE X.

07 PRINT-LINE.

10 CARD-PRINT PICTURE X(80).

10 MESSAGE-AREA PICTURE X(52).

FD TRANS RECORD CONTAINS 80 CHARACTERS

LABEL RECORDS ARE OMITTED

RECORDING MODE IS F

DATA RECORD IS INPCT.

01 INPCT.

07 CARD-PORCION PICTURE X(80).

FD OLD-MASTER BLOCK CONTAINS 25 RECORDS

RECORD CONTAINS 80 CHARACTERS

LABEL RECORDS ARE STANDARD

RECORDING MODE IS F

DATA RECORD IS OLD-MAST.

01 OLD-MAST.

07 FILLER PICTURE X(80).

FD NEW-MASTER BLOCK CONTAINS 25 RECORDS

RECORD CONTAINS 80 CHARACTERS

LABEL RECORDS ARE STANDARD

RECORDING MODE IS F

DATA RECORD IS NEW-MASTER.

01 NEW-MASTER.

07 FILLER PICTURE X(80).

WORKING-STORAGE SECTION.

77 NONE PICTURE X VALUE IS SPACE.

77 SKIP PICTURE X VALUE IS 'I'.

77 CREATE-FLAG PICTURE IS X VALUE IS ZERO.

77 LINE-CTR PICTURE 9999 VALUE IS ZERO.

77 GCFLAG PICTURE 9999 VALUE IS ZERO.

77 LCW-TEST PICTURE X(12) VALUE IS LOW-VALUES.

77 INDEX-GO-FLAG PICTURE X VALUE IS ZERO.

```

77 SUB PICTURE 99 VALUE IS ZERC.
77 HANK PICTURE X.
77 F1 PICTURE 9999 VALUE IS ZERC.
01 NF.
   03 N8 PICTURE X(8).
01 X1 REDEFINES NR.
   02 N7 PICTURE IS X(7).
   02 FILLER PICTURE IS 9.
01 X2 REDEFINES X1.
   02 FILLER PICTURE 9.
   02 N6 PICTURE X(6).
   02 FILLER PICTURE IS 9.
01 X3 REDEFINES X2.
   02 FILLER PICTURE 9(2).
   02 N5 PICTURE X(5).
   02 FILLER PICTURE IS 9.
01 X4 REDEFINES X3.
   02 FILLER PICTURE 9(3).
   02 N4 PICTURE X(4).
   02 FILLER PICTURE IS 9.
01 X5 REDEFINES X4.
   02 FILLER PICTURE 9(4).
   02 N3 PICTURE X(3).
   02 FILLER PICTURE IS 9.
01 X6 REDEFINES X5.
   02 FILLER PICTURE 9(5).
   02 N2 PICTURE X(2).
   02 FILLER PICTURE IS 9.
01 X7 REDEFINES X6.
   02 FILLER PICTURE 9(6).
   02 N1 PICTURE X(1).
   02 FILLER PICTURE IS 9.
01 NSTAR REDEFINES X7.
   03 ASTIC PICTURE X(7).
   03 FILLER PICTURE X.
01 TRANS-CARD.
   07 INDEX-NO-TRANS.
   11 C11 PICTURE IS          X.
   11 CLM12.
   22 CL12.
   33 C12 PICTURE IS          X.
   33 C13 PICTURE IS          X.
   33 C14 PICTURE IS          X.
   22 CL15.
   33 C15 PICTURE IS          X.
   33 C16 PICTURE IS          X.
   22 CL17.
   33 C17 PICTURE IS          X.
   33 C18 PICTURE IS          X.
   33 C19 PICTURE IS          X.
   22 CL110.
   33 C110 PICTURE IS         X.
   33 C111 PICTURE IS         X.
   33 C112 PICTURE IS         X.
07 REST.
   11 C113 PICTURE IS         X(4).
   11 C117 PICTURE IS         X(3).
   11 C120 PICTURE IS         X(4).
   11 C124 PICTURE IS         X.
   11 C125 PICTURE IS         X(3).

```



```

11 CL128.
22 C128 PICTURE IS      X(3).
22 C131 PICTURE IS      X(3).
22 C134 PICTURE IS      X.
22 C135 PICTURE IS      X(3).
22 C138 PICTURE IS      X(3).
22 C141 PICTURE IS      X.
22 C142 PICTURE IS      X(3).
22 C145 PICTURE IS      X(3).
22 C148 PICTURE IS      X.
22 C149 PICTURE IS      X(3).
22 C152 PICTURE IS      X(3).
22 C155 PICTURE IS      X.
22 C156 PICTURE IS      X(3).
22 C159 PICTURE IS      X(3).
22 C162 PICTURE IS      X.
11 LITHOLOGY REDEFINES CL128.
22 LITHCCDE OCCURS 5 TIMES.
33 LITH-FRCM PICTURE IS X(3).
33 LITH-TO   PICTURE IS X(3).
33 LITH-CCDE PICTURE IS  X.
11 C163 PICTURE IS      X.
11 C164 PICTURE IS      X(3).
11 C167 PICTURE IS      X(3).
11 C170 PICTURE IS      X.
11 C171 PICTURE IS      X(3).
11 C174 PICTURE IS      X(3).
11 DELT PICTURE IS X(4).
01 OLD-MAST-WORK.
11 CLD-MAST-INDEX PICTURE IS X(12).
11 FILLER PICTURE IS X(68).
01 RUN-TYPE.
11 FILLER PICTURE IS X(32) VALUE IS
   '      TYPE OF RUN REQUESTED-----'.
11 CONTROL-CARD PICTURE IS X(6).
01 HEAD-LINE.
11 L-1 PICTURE X(39) VALUE IS SPACE.
11 L-2 PICTURE X(55) VALUE IS
   'EEDRCK TCFCOGRAPHY MAPPING RESEARCH CCUNCIL OF ALBERTA'.
01 SEC-LINE.
11 FILLER PICTURE X(5) VALUE IS SPACES.
11 FILLER PICTURE X(53) VALUE IS
   'COMPLETE LIST OF TRANSACTION CARDS PROCESSED THIS RUN'.
PROCEDURE DIVISION.
HEAD SECTION.
A1.
   CFEN INPUT TRANS OUTPUT PRINTER.
   READ TRANS INTO CONTROL-CARD AT END GO TO BAD-RUN.
   IF CONTROL-CARD = 'UPDATE' GO TO UPDATE.
   IF CONTROL-CARD = 'LIST  ' GO TO LIST.
   IF CONTROL-CARD = 'CREATE' GO TO CREATE.
   MOVE RUN-TYPE TO CARD-PRINT.
   MOVE '   INCORRECT RUN TYPE REQUESTED' TO MESSAGE-AREA.
   WRITE PRINT-REC AFTER SKIP.
   GO TO BAD-RUN.
CREATE.
   CFEN OUTPUT NEW-MAST.
   MOVE HIGH-VALUES TO OLD-MAST-INDEX.
   MOVE 1 TO CREATE-FLAG.
   GO TO HEADER.

```

```

UPDATE.
  CFEN INPUT OLD-MASTER OUTPUT NEW-MAST.
  REAC CLD-MASTER INTO OLD-MAST-WCRK AT END GO TO BAD-RUN.
  GC TO HEADER.
LIST.
  CLOSE TRANS. OPEN INPUT CLD-MASTER.
  MCVE 56 TO LINE-CTR.
A2.
  READ CLD-MASTER INTO TRANS-CARD AT END GO TO A3.
  IF LINE-CTR IS > 55 PERFORM HEADER.
  ACD 1 TC LINE-CTR.
  WRITE PRINT-REC FROM TRANS-CARD AFTER NCNE.
  IF INDEX-NO-TRANS IS NCT = HIGH-VALUES GO TO A2.
A3.
  CLOSE OLD-MASTER WITH LCCK
  PRINTER. STOP RUN.
HEADER SECTION.
B1.
  MCVE SPACES TO PRINT-LINE.
  WRITE PRINT-REC AFTER SKIP.
  WRITE PRINT-REC FROM HEAD-LINE AFTER NONE.
  MCVE SPACES TO PRINT-LINE.
  WRITE PRINT-REC AFTER NCNE.
  WRITE PRINT-REC FROM RUN-TYPE AFTER NONE.
  WRITE PRINT-REC FROM SEC-LINE AFTER NONE.
  MCVE SPACES TO PRINT-LINE.
  WRITE PRINT-REC AFTER NCNE.
  MCVE 6 TO LINE-CTR.
B2.  EXIT.
BODY SECTION.
C1.
  REAC TRANS INTO TRANS-CARD AT END GO TO GOOD-END.
  ACD 1 TO LINE-CTR.
  IF LINE-CTR IS > 55 PERFORM HEADER.
  PERFORM EDIT-CHECK.
  MCVE TRANS-CARD TO PRINT-LINE.
  IF GCFLAG NOT = ZERO MCVE
  ' ***** REJECTED DUE TO ERRORS ***** '
  TC MESSAGE-AREA
  WRITE PRINT-REC AFTER NONE GO TO C1.
C2.
  IF DELT = 'DELT' GO TO CELETE-ROUTINE.
C4.  IF OLD-MAST-INDEX IS > INDEX-NO-TRANS
  WRITE NEW-MASTER FROM TRANS-CARD
  MCVE ' INSERTED' TO MESSAGE-AREA
  WRITE PRINT-REC AFTER NCNE
  GC TO C1.
J9.
  IF CLD-MAST-INDEX = INDEX-NC-TRANS MOVE
  ' ***** ATTEMPT TO INSERT DUP WELL ***** '
  TC MESSAGE-AREA WRITE PRINT-REC
  AFTER NONE GO TO C1.
C3.
  WRITE NEW-MASTER FROM OLD-MAST-WCRK.
  READ CLD-MASTER INTO OLD-MAST-WCRK AT END GO TO BAD-RUN.
  GC TO C4.
GOOD-END.
  IF CLD-MAST-INDEX IS NOT = HIGH-VALUES GO TO C6.
C5.
  WRITE NEW-MASTER FROM CLD-MAST-WCRK.

```

CLOSE TRANS PRINTER NEW-MAST WITH LOCK.
IF CREATE-FLAG = ZERO CLOSE CLD-MASTER WITH LOCK.
STOP RUN.

C6.

WRITE NEW-MASTER FROM CLD-MAST-WCRK.
READ CLD-MASTER INTO OLD-MAST-WCRK AT END GO TO EAD-RUN.
GC TO GOOD-END.

DELETE-ROUTINE.

IF CREATE-FLAG = 1 MOVE
' ***** ILLEGAL DELETE CARD ***** '
TC MESSAGE-AREA
WRITE PRINT-REC AFTER NCNE GC TO C1.

C8.

IF CLD-MAST-INDEX IS < INDEX-NC-TRANS GC TO C7.
IF CLD-MAST-INDEX IS > INDEX-NC-TRANS MCVE
' ***** ATTEMPT TO DELETE NCNEXISTANT FILE **** '
TC MESSAGE-AREA
WRITE PRINT-REC AFTER NCNE GC TO C1.
READ CLD-MASTER INTO OLD-MAST-WORK AT END GO TO BAD-RUN.
MCVE ' FILE WITH THIS NUMBER DELETED '
TC MESSAGE-AREA
WRITE PRINT-REC AFTER NCNE GC TO C1.

C7.

WRITE NEW-MASTER FROM CLD-MAST-WCRK.
READ CLD-MASTER INTO OLD-MAST-WORK AT END GO TO BAD-RUN.
GC TO C8.

BAD-RUN.

MCVE ' JOB ABORTED DUE TO UPDATE FAILURE '
TC PRINT-LINE.
WRITE PRINT-REC AFTER SKIP.
CLOSE TRANS PRINTER NEW-MAST.
IF CREATE-FLAG = ZERO CLOSE CLD-MASTER WITH LOCK.
STOP RUN.

EDIT-CHECK SECTION.

E1.

MCVE ZERO TO GOFLAG INDEX-GC-FLAG.

E2.

MCVE ZEROS TO N8.
MCVE C11 TO N1. IF N8 NOT NUMERIC MCVE '*' TO C11
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
IF C11 < '4' OR > '6' MOVE '*' TO C11
MCVE 1 TO GOFLAG INDEX-GO-FLAG.
MCVE ZEROS TO N8.
MCVE CL12 TO N3. IF N8 NOT NUMERIC MOVE '*' TO C12
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
IF CL12 < '001' OR > '126' MCVE '***' TO CL12
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
MCVE ZEROS TO N8.
MCVE CL15 TO N2. IF N8 NOT NUMERIC MOVE '*' TO C15
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
IF CL15 < '01' OR > '30' MOVE '***' TO CL15
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
MCVE ZEROS TO N8.
MCVE CL17 TO N3. IF N8 NOT NUMERIC MOVE '*' TO C17
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
IF CL17 < '000' OR > '599' MCVE '***' TO CL17
MCVE 1 TO GOFLAG INDEX-GC-FLAG.
MCVE ZEROS TO N8.
MCVE CL110 TO N3. IF N8 NOT NUMERIC MOVE '*' TO C110
MCVE 1 TO GOFLAG INDEX-GC-FLAG.

IF CL110 < '000' OR > '599' MOVE '****' TC CL110
MCVE 1 TO GOFLAG INDEX-GC-FLAG.

E3.

IF C113 = SPACES MOVE ZERCS TO C113 ELSE
MCVE ZEROS TO N8 MOVE C113 TC N4 PERFORM ACHK
MCVE N4 TO C113.

IF C113 < '0450' OR > '9999' MOVE '*****' TO C113
MCVE 1 TO GOFLAG INDEX-GO-FLAG.

IF C117 = SPACE MOVE ZERCS TC C117 ELSE
MCVE ZEROS TO N8 MOVE C117 TC N3 PERFORM ACHK
MCVE N3 TO C117.

IF C120 = SPACES MOVE ZERCS TC C120 ELSE
MCVE ZEROS TO N8 MCVE C120 TC N4 PERFORM ACHK
MCVE N4 TO C120.

IF C124 IS < '1' OR > '2' MOVE '*' TC C124 MOVE 1 TO GOFLAG.

IF C125 = SPACES MOVE ZERCS TO C125 ELSE
MCVE ZEROS TO N8 MCVE C125 TC N3 PERFORM ACHK
MCVE N3 TO C125.

IF C125 = ZERO MOVE ALL '*' TO C125 MOVE 1 TO GOFLAG.
MCVE ZERO TO F1.

PERFORM E4 VARYING SUB FROM 1 BY 1 UNTIL SUB IS > 5.
GC TO E5.

E4.

IF LITH-FROM (SUB) = SPACE MCVE ZEROS TC LITH-FROM (SUB)
MOVE 1 TO F1

ELSE
MCVE ZERCS TO N8 MOVE LITH-FROM (SUB) TO N3 PERFORM ACHK
MCVE N3 TC LITH-FROM (SUB)

IF F1 NOT = ZERO MOVE '****' TC LITH-FROM (SUB) MOVE 1 TO
GOFLAG.

IF LITH-TO (SUB) = SPACE MCVE ZERCS TC LITH-TO (SUB)
MOVE 1 TO F1

ELSE
MCVE ZERCS TO N8 MOVE LITH-TO (SUB) TO N3 PERFORM ACHK
MCVE N3 TO LITH-TO (SUB)

IF F1 NOT = ZERO MOVE '****' TO LITH-TO (SUB) MOVE 1 TO
GOFLAG.

IF LITH-CODE (SUB) = SPACES MOVE ZERO TO LITH-CODE (SUB)
MCVE 1 TO F1

ELSE
MCVE ZERCS TO N8 MOVE LITH-CODE (SUB) TO N1 PERFORM ACHK
MCVE N1 TO LITH-CODE (SUB)

IF F1 NOT = ZERO MOVE '*' TC LITH-CODE (SUB) MOVE 1 TO
GOFLAG.

IF LITH-CODE (SUB) < '0' OR > '9' MOVE '*' TO
LITH-CODE (SUB) MOVE 1 TO GOFLAG.

MCVE 'U' TO FANK.

E5.

IF C163 = SPACES MOVE ZERO TC C163.

IF C163 < '0' OR > '2' MCVE '*' TO C163 MOVE 1 TO GOFLAG.

IF C164 = SPACES MOVE ZERCS TO C164 ELSE

MCVE ZEROS TO N8 MCVE C164 TC N3 PERFORM ACHK
MCVE N3 TO C164.

IF C167 = SPACES MOVE ZEROS TO C167 ELSE

MCVE ZEROS TO N8 MCVE C167 TC N3 PERFORM ACHK
MCVE N3 TO C167.

IF C170 = SPACE MOVE ZERO TO C170.

IF C170 < '0' OR > '2' MCVE '*' TC C170 MOVE 1 TO GOFLAG.

IF C171 = SPACES MOVE ZEROS TO C171 ELSE

MCVE ZEROS TO N8 MCVE C171 TC N3 PERFORM ACHK

MCVE N3 TO C171.
IF C174 = SPACES MOVE ZERCS TO C174 ELSE
MCVE ZEROS TO N8 MOVE C174 TC N3 PERFORM ACHK
MCVE N3 TO C174.

E6.

IF INDEX-GO-FLAG NOT = ZERO GO TC E7.
IF INDEX-NO-TRANS < LOW-TEST MCVE 1 TO GOFLAG ELSE
MCVE INDEX-NO-TRANS TO LOW-TEST.

E7. EXIT.

ACHK SECTION.

H1.

IF N8 IS NOT NUMERIC MOVE ALL '*' TC ASTIC
MCVE 1 TO GOFLAG.

H2. EXIT.

\$C RETRIEVAL

IDENTIFICATION DIVISION.
PROGRAM-ID. 'MAPPING'.
AUTHOR. NUCLEUS.
INSTALLATION. U OF A COMPUTING CENTER.

DATE-WRITTEN. JULY 1967.
DATE-COMPILED. JULY 1967.
SECURITY. CONFIDENTIAL.

REMARKS. MAINTENANCE PROGRAM FOR LITHOLOGY MAPPING.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM-360.
OBJECT-COMPUTER. IBM-360.
INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT PRINTER ASSIGN TO 'UT-S-SYSOUT' UNIT-RECORD.
SELECT TRANS ASSIGN TO 'UT-S-SYSIN' UNIT-RECORD.
SELECT NEW-MAST ASSIGN TO 'UT-S-SYSUT4' UTILITY.
SELECT REPFIL ASSIGN TO 'UT-S-SYSUT3' UTILITY.
SELECT FLOT-TAPE ASSIGN TO 'UT-S-SYSUT8' UTILITY.

DATA DIVISION.

FILE SECTION.

FD PRINTER RECORD CONTAINS 133 CHARACTERS

LABEL RECORDS ARE OMITTED
RECORDING MODE IS F
DATA RECORD IS PRINT-REC.

01 PRINT-REC.

07 FILLER PICTURE X.
07 PRINT-LINE.

10 CARD-PRINT PICTURE X(80).

10 MESSAGE-AREA PICTURE X(52).

FD TRANS RECORD CONTAINS 80 CHARACTERS

LABEL RECORDS ARE OMITTED
RECORDING MODE IS F
DATA RECORD IS INPCT.

01 INPCT.

07 CARD-PORTION PICTURE X(80).

FD NEW-MAST BLOCK CONTAINS 25 RECORDS

RECORD CONTAINS 80 CHARACTERS
LABEL RECORDS ARE STANDARD
RECORDING MODE IS F
DATA RECORD IS NEW-MASTER.

01 NEW-MASTER.

07 FILLER PICTURE X(80).

FD FLOT-TAPE BLOCK CONTAINS 40 RECORDS

RECORD CONTAINS 20 CHARACTERS
LABEL RECORDS ARE STANDARD
RECORDING MODE IS F
DATA RECORD IS PLITER.

01 PLITER.

07 FILLER PICTURE IS X(20).

FD REPFIL RECORD CONTAINS 132 CHARACTERS

BLOCK CONTAINS 50 RECORDS
LABEL RECORDS ARE STANDARD
RECORDING MODE IS F
DATA RECORD IS REPREC.

01 REPREC.

05 FILLER PICTURE X(132).

WORKING-STORAGE SECTION.

77 CCM-TIP PICTURE IS S9999.
77 CTR5 PICTURE IS S999 COMPUTATIONAL.
77 CTR6 PICTURE IS S999 CCMPUTATIONAL.
77 CTR1 PICTURE IS S999 CCMPUTATIONAL VALUE IS 2.
77 FIRST-FLAG PICTURE IS S99.
77 LYNE PICTURE IS S99 COMPUTATIONAL.
77 CTR2 PICTURE IS S999 CCMPUTATIONAL.
77 RUN-T PICTURE IS S99 CCMPUTATIONAL.
77 PRINT-T PICTURE IS S99 CCMPUTATIONAL.
77 NCNE PICTURE X VALUE IS SPACE.
77 SKIP PICTURE X VALUE IS '1'.
77 BLANK2 PICTURE XXX VALUE IS SPACES.
77 LINE-CTR PICTUPE 9999 VALUE IS ZERC.
77 GCFLAG PICTURE 9999 VALUE IS ZERO.
77 SLB PICTURE 99 VALUE IS ZERC.
77 HANK PICTURE X.
77 N-SUB PICTURE S999 COMPUTATIONAL.
77 W-SUB PICTURE S999 CCMPUTATIONAL.
77 PLIT PICTURE IS 99.
77 P-SURF PICTURE IS 9999.
77 FIVE PICTURE IS 9999 VALUE IS 5.
77 ED5M PICTURE ----9.
77 ED4P PICTURE ZZZ9.
77 ED3P PICTURE ZZ9.
77 R-CTR9 PICTURE S9(8) CCMPUTATIONAL VALUE ZERO.
77 R-CTR8 PICTURE S9(8) CCMPUTATIONAL VALUE ZERO.
77 CTR8 PICTURE S999 COMPUTATIONAL.
77 CTR9 PICTURE S999 CCMPUTATIONAL.
77 TMHCLD PICTURE 9999.
01 FGH-HLD.
03 FG-HLD OCCURS 5 TIMES PICTURE X.
01 REP-WORK.
05 FILLER PICTURE X(20) VALUE SPACES.
05 FILLER PICTURE X(28) VALUE IS
'NUMBER OF WELLS IN MERIDIAN'.
05 R-MERD PICTURE X.
05 FILLER PICTURE X(10) VALUE IS
'TCWNSHIP'.
05 R-TP PICTURE XXX.
05 FILLER PICTURE X(7) VALUE IS
'RANGE'.
05 R-RG PICTURE XX.
05 FILLER PICTURE X(4) VALUE IS
'IS'.
05 R-COUNT PICTURE Z(7)9.
05 FILLER PICTURE X(21) VALUE IS
'NUMBER RETRIEVED IS'.
05 RT-CGUNT PICTURE Z(7)9.
05 FILLER PICTURE X(20) VALUE IS SPACES.
01 HD-FICT.
03 HD-PIC OCCURS 4 TIMES PICTURE X.
01 LCW-HLD.
03 LW-HLD OCCURS 4 TIMES PICTURE X.
01 REL-HLD.
03 C134F PICTURE X.
03 C141F PICTUPE X.
03 C148H PICTURE X.
03 C155H PICTURE X.
03 C162H PICTURE X.
01 REAL-HLD REDEFINES REL-HLD.

03 RL-HLD OCCURS 5 TIMES PICTURE X.
 01 DATA-BAD.
 03 FILLER PICTURE IS X VALUE IS SPACE.
 03 W-ERR PICTURE X(80).
 03 FILLER PICTURE IS X(19) VALUE IS
 ' DATA ERROR *****'.

01 FLIT-CD.
 05 PLIT-INDEX PICTURE IS X(12).
 05 SQ-MILE PICTURE IS XX.
 05 HF-MILE PICTURE IS X.
 05 PLT-ENTRY PICTURE IS XXXX.
 05 FILLER PICTURE IS X VALUE IS SPACE.

01 TRANS-CARD.
 07 INDEX-NO-TRANS.
 09 REP-TIL.
 11 C-11 PICTURE IS X.
 11 CLM12.
 22 CL12.
 33 C-12 PICTURE IS X.
 33 C13 PICTURE IS X.
 33 C14 PICTURE IS X.
 22 CL15.
 33 C15 PICTURE IS X.
 33 C16 PICTURE IS X.
 09 REP-REST.
 22 CL17.
 33 C17 PICTURE IS X.
 33 N-MILE REDEFINES C17 PICTURE IS 9.
 24 N-HALF.
 33 C18 PICTURE IS X.
 33 C19 PICTURE IS X.
 22 CL110.
 33 C110 PICTURE IS X.
 33 W-MILE REDEFINES C110 PICTURE IS 9.
 24 W-HALF.
 33 C111 PICTURE IS X.
 33 C112 PICTURE IS X.
 07 REST.
 11 C113 PICTURE IS X(4).
 11 C-113 REDEFINES C113 PICTURE IS 9999.
 11 C117 PICTURE IS X(3).
 11 C120 PICTURE IS X(4).
 11 C124 PICTURE IS X.
 11 C125 PICTURE IS X(3).
 11 C-125 REDEFINES C125 PICTURE IS 999.
 11 CL128.
 22 C128 PICTURE IS X(3).
 22 C131 PICTURE IS X(3).
 22 C134 PICTURE IS X.
 22 C135 PICTURE IS X(3).
 22 C138 PICTURE IS X(3).
 22 C141 PICTURE IS X.
 22 C142 PICTURE IS X(3).
 22 C145 PICTURE IS X(3).
 22 C148 PICTURE IS X.
 22 C149 PICTURE IS X(3).
 22 C152 PICTURE IS X(3).
 22 C155 PICTURE IS X.
 22 C156 PICTURE IS X(3).
 22 C159 PICTURE IS X(3).

22 C162 PICTURE IS X.
 11 LITHOLOGY REDEFINES CL128.
 22 LITHCCDE OCCURS 5 TIMES.
 33 LITH-FRCM PICTURE IS X(3).
 33 LITH-TO PICTURE IS X(3).
 33 LITH-CODE PICTURE IS X.

11 LITH-NUM REDEFINES LITHCLOGY.
 22 LT-NUM OCCURS 5 TIMES.
 33 LT-NUM-FM PICTURE 999.
 33 LT-NUM-TO PICTURE 999.
 33 LT-NUM-C PICTURE X.

11 C163 PICTURE IS X.
 11 C164 PICTURE IS X(3).
 11 C-164 REDEFINES C164 PICTURE IS 999.
 11 C167 PICTURE IS X(3).
 11 C-167 REDEFINES C167 PICTURE IS 999.
 11 C170 PICTURE IS X.
 11 C171 PICTURE IS X(3).
 11 C-171 REDEFINES C171 PICTURE IS 999.
 11 C174 PICTURE IS X(3).
 11 C-174 REDEFINES C174 PICTURE IS 999.
 11 DELT PICTURE IS X(4).

01 IN-HLD PICTURE X(6) VALUE ZERCS.
 01 IN-HDL REDEFINES IN-HLD.
 03 IN-M PICTURE X.
 03 IN-TP PICTURE XXX.
 03 IN-RG PICTURE XX.

01 RUN-TYPE.
 11 FILLER PICTURE IS X(32) VALUE IS
 ' TYPE OF RUN REQUESTED-----'.
 11 CONTROL-CARD PICTURE IS X(11).

01 BC-ELV.
 03 RGCK-ELV PICTURE XXXX.

01 NUM-BD-ELV REDEFINES BC-ELV.
 03 NM-ROCK PICTURE 9999.

01 P-EDIT.
 03 FILLER PICTURE IS X VALUE IS '.'.

03 P-EDT PICTURE IS XXX.

01 M-EDIT.
 03 FILLER PICTURE IS X VALUE IS '-'.

03 M-EDT PICTURE IS XXX.

01 PTAELE.
 03 FILLER PICTURE XX VALUE IS '01'.
 03 FILLER PICTURE XX VALUE IS '02'.
 03 FILLER PICTURE XX VALUE IS '03'.
 03 FILLER PICTURE XX VALUE IS '04'.
 03 FILLER PICTURE XX VALUE IS '05'.
 03 FILLER PICTURE XX VALUE IS '06'.
 03 FILLER PICTURE XX VALUE IS '12'.
 03 FILLER PICTURE XX VALUE IS '11'.
 03 FILLER PICTURE XX VALUE IS '10'.
 03 FILLER PICTURE XX VALUE IS '09'.
 03 FILLER PICTURE XX VALUE IS '08'.
 03 FILLER PICTURE XX VALUE IS '07'.
 03 FILLER PICTURE XX VALUE IS '13'.
 03 FILLER PICTURE XX VALUE IS '14'.
 03 FILLER PICTURE XX VALUE IS '15'.
 03 FILLER PICTURE XX VALUE IS '16'.
 03 FILLER PICTURE XX VALUE IS '17'.
 03 FILLER PICTURE XX VALUE IS '18'.

```

03 FILLER PICTURE XX VALUE IS '24'.
03 FILLER PICTURE XX VALUE IS '23'.
03 FILLER PICTURE XX VALUE IS '22'.
03 FILLER PICTURE XX VALUE IS '21'.
03 FILLER PICTURE XX VALUE IS '20'.
03 FILLER PICTURE XX VALUE IS '19'.
03 FILLER PICTURE XX VALUE IS '25'.
03 FILLER PICTURE XX VALUE IS '26'.
03 FILLER PICTURE XX VALUE IS '27'.
03 FILLER PICTURE XX VALUE IS '28'.
03 FILLER PICTURE XX VALUE IS '29'.
03 FILLER PICTURE XX VALUE IS '30'.
03 FILLER PICTURE XX VALUE IS '36'.
03 FILLER PICTURE XX VALUE IS '35'.
03 FILLER PICTURE XX VALUE IS '34'.
03 FILLER PICTURE XX VALUE IS '33'.
03 FILLER PICTURE XX VALUE IS '32'.
03 FILLER PICTURE XX VALUE IS '31'.
01 ATABLE REDEFINES PTABLE.
03 STABLE OCCURS 6 TIMES.
05 TTABLE OCCURS 6 TIMES PICTURE IS XX.
01 HEAD-LINE.
    03 FILLER PICTURE IS X(7) VALUE IS SPACES.
03 FILLER PICTURE IS X(58) VALUE IS
' R E S E A R C H   C O U N C I L   O F   A L B E R T A   - '.
03 FILLER PICTURE IS X(58) VALUE IS
' G R O U N D   W A T E R   D I V I S I O N   -   M A P P I N ' .
03 FILLER PICTURE IS XX VALUE IS ' G ' .
01 HEAD2.
03 FILLER PICTURE IS X(26) VALUE IS SPACES.
03 CRD-RC PICTURE IS X(80).
03 FILLER PICTURE IS X(26) VALUE IS SPACES.
01 HEADERS3.
    03 FILLER PICTURE IS X(33) VALUE IS
' L C C A T I O N   O F   W E L L   I S   W E S T   O F   T H E ' .
03 MERID PICTURE IS X.
03 FILLER PICTURE IS X(12) VALUE IS ' T H   M E R I D I A N . ' .
01 CARD-TYPE.
03 CRIT PICTURE IS XX.
03 FILLER PICTURE X(7).
03 LCRIT PICTURE IS XXXX.
03 FILLER PICTURE IS X(6).
03 HCRIT PICTURE IS XXXX.
03 FILLER PICTURE IS X(57).
01 CHEKER.
03 FILLER PICTURE IS X VALUE IS ZERC.
03 NUMB PICTURE IS XX.
03 NUMM REDEFINES NUMB PICTURE IS 99.
03 FILLER PICTURE IS X VALUE IS ZERO.
01 CEKER REDEFINES CHEKER PICTURE IS XXXX.
01 INTERPETER.
03 PETER OCCURS 100 TIMES.
    05 TPE PICTURE 99.
    05 LOW-AL PICTURE IS XXXX.
    05 HIGH-AL PICTURE IS XXXX.
01 CARD-TYPE.
03 FILLER PICTURE X(11).
01 ERROR1.
03 FILLER PICTURE X(30) VALUE IS
' N C   I N P U T   C A R D S   R U N   T E R M I N A T E D ' .

```

```

01  EFRRCR2.
03  FILLER PICTURE IS X(54) VALUE IS
    ' NC CRITERIA CARDS CR CRITERIA END CARD RUN TERMINATED'.
01  EFRRCR3.
03  FILLER PICTURE IS X(38) VALUE IS
    ' INVALID PLOT ASKED FOR RUN TERMINATED'.
01  EFRRCR4.
03  FILLER PICTURE X(42) VALUE IS
    ' INVALID CRITERIA ASKED FOR RUN TERMINATED'.
01  EFRRCR5.
03  FILLER PICTURE X(42) VALUE IS
    ' RUN TYPE CODE IS NOT VALID RUN TERMINATED'.
01  EFRRCR6.
03  FILLER PICTURE X(18) VALUE IS
    ' INTERPRETER ERROR'.
01  LITH-KODE.
03  FILLER PICTURE IS X(40) VALUE IS
    ' LITHCODE ALL TRUE -- FIRST ONE DELETED.'.
01  EFRRCR7.
03  FILLER PICTURE IS X(37) VALUE IS
    ' NC PLOT CARD SUPPLIED RUN TERMINATED'.
01  FLG1.
03  FG1 PICTURE 9.
03  FG2 PICTURE 9.
03  FG3 PICTURE 9.
03  FG4 PICTURE 9.
03  FG5 PICTURE 9.
01  FLG1-CC REDEFINES FLG1.
03  FLH1 OCCURS 5 TIMES PICTURE 9.
01  FLG2.
03  FL1 PICTURE 9.
03  FL2 PICTURE 9.
03  FL3 PICTURE 9.
03  FL4 PICTURE 9.
03  FL5 PICTURE 9.
01  FLG2-CC REDEFINES FLG2.
03  FLH OCCURS 5 TIMES PICTURE 9.
01  CLT-LINE.
    02  F-PART.
03  FILLER PICTURE X.
03  LN1 PICTURE XXX.
03  FILLER PICTURE X.
03  LN2 PICTURE XX.
03  FILLER PICTURE X.
03  LN3 PICTURE XXX.
03  FILLER PICTURE X.
03  LN4 PICTURE XXX.
03  FILLER PICTURE XX.
03  LN28 PICTURE XXXXX.
03  LN28-1 PICTURE X.
03  FILLER PICTURE X.
03  LN7 PICTURE XXXX.
03  FILLER PICTURE X.
03  LN8 PICTURE XXX.
03  LNA PICTURE X.
03  FILLER PICTURE XX.
    02  LITH-PART.
03  LN9 PICTURE XXX.
03  FILLER PICTURE X.
03  LN10 PICTURE XXX.

```

03 FILLER PICTURE X.
03 LN11 PICTURE XXX.
03 FILLER PICTURE XX.
03 LN12 PICTURE XXX.
03 FILLER PICTURE X.
03 LN13 PICTURE XXX.
03 FILLER PICTURE X.
03 LN14 PICTURE XXX.
03 FILLER PICTURE XX.
03 LN15 PICTURE XXX.
03 FILLER PICTURE X.
03 LN16 PICTURE XXX.
03 FILLER PICTURE X.
03 LN17 PICTURE XXX.
03 FILLER PICTURE XX.
03 LN18 PICTURE XXX.
03 FILLER PICTURE X.
03 LN19 PICTURE XXX.
03 FILLER PICTURE X.
03 LN20 PICTURE XXX.
03 FILLER PICTURE XX.
03 LN21 PICTURE XXX.
03 FILLER PICTURE X.
03 LN22 PICTURE XXX.
03 FILLER PICTURE X.
03 LN23 PICTURE XXX.
03 FILLER PICTURE XX.

02 LIT-PART REDEFINES LITH-PART.

03 LT-ARA OCCURS 5 TIMES.

05 LT-CD PICTURE XXX.

05 FILLER PICTURE X.

05 LT-FM PICTURE XXX.

05 ENCT PICTURE X.

05 LT-TG PICTURE XXX.

05 FILLER PICTURE XX.

02 L-PART.

03 LN24 PICTURE XXX.
03 FILLER PICTURE X.
03 LN25 PICTURE XXX.
03 FILLER PICTURE XX.
02 E-PART.
03 LN26 PICTURE XXX.
03 FILLER PICTURE X.
03 LN27 PICTURE XXX.
03 FILLER PICTURE XX.
03 LN5 PICTURE XXXX.
03 FILLER PICTURE X.
03 LN6 PICTURE XXX.

01 TSTR.

02 CMPT PICTURE XXX.

02 CMNT REDEFINES CMPT PICTURE 999.

01 TCT-HEAD.

03 FILLER PICTURE X(56) VALUE IS

' TP. RG NCR WST BR ELV DRIL DRFT LITH FRM TO LITH FRM '.

03 FILLER PICTURE X(58) VALUE IS

' TC LITH FRM TC LITH FRM TC LITH FRM TO BLIND FLOWI'.

03 FILLER PICTURE X(2) VALUE IS

'NG'.

03 FILLER PICTURE IS X(10) VALUE IS

' SURF CO.'.

PROCEDURE DIVISION.
MAIN SECTION.

MCVE 11111 TO FLG1 FLG2.
OPEN INPUT TRANS OUTPUT PRINTER.
OPEN OUTPUT REPFIL.
MCVE '4' TO MERID.

MCVE SPACES TO OUT-LINE.

ST1.

MCVE 1 TO FIRST-FLAG.
MCVE 19 TO TIPE (1).
READ TRANS INTO CRD-TC AT END GO TO ERR1.
PERFORM WRT-HEAD.
MCVE 2 TO PRINT-T.
READ TRANS INTO CARD-TIPE AT END GO TO ERR2.
IF CARD-TIPE = 'PRINT' MCVE 1 TO RUN-T
MCVE 2 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'PRINT LONG' MCVE 1 TO RUN-T
MCVE 2 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'PRINT SHCPT' MCVE 1 TO RUN-T
MCVE 1 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'PLOT' MCVE 2 TO RUN-T
MCVE 1 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'BOTH LONG' MCVE 3 TO RUN-T
MCVE 2 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'BOTH SHORT' MCVE 3 TO RUN-T
MCVE 1 TO PRINT-T GO TO ST1-1.
IF CARD-TIPE = 'BOTH' MCVE 3 TO RUN-T
MCVE 2 TO PRINT-T ELSE GO TO ERR5.

ST1-1.

MCVE CARD-TIPE TO CONTROL-CARD.
WRITE PRINT-REC FROM RUN-TYPE AFTER POSITIONING NONE.

ST2.

READ TRANS INTO CARD-TYPE AT END GO TO ERR2.
IF CRIT = '20' GO TO ST3.
MCVE CRIT TO NUMB.
IF CEKER IS NOT NUMERIC GO TO ERR4.
MCVE NUMM TO TIPE (CTR1).
MCVE LCRIT TO LOW-AL (CTR1).
MCVE HCRIT TO HIGH-AL (CTR1).
MCVE CARD-TYPE TO PRINT-LINE.
WRITE PRINT-REC AFTER POSITIONING NONE.
ADD 1 TO CTR1.
GO TO ST2.

WRT-HEAD.

MCVE ZERO TO LYNE.
WRITE PRINT-REC FROM HEAD-LINE AFTER POSITIONING SKIP.
WRITE PRINT-REC FROM HEAD2 AFTER POSITIONING 2.
WRITE PRINT-REC FROM HEADER3 AFTER POSITIONING 2.
WRITE PRINT-REC FROM
TCT-HEAD AFTER POSITIONING 1.

ST3.

MCVE 20 TO TIPE (CTR1).
ADD 1 TO CTR1.
MCVE 21 TO TIPE (CTR1).
MCVE 57 TO LYNE.
OPEN INPUT NEW-MAST.
IF RUN-T = 2 OR RUN-T = 3 PERFORM READ-PLOT.
GO TO ST5.

READ-FLCT.

READ TRANS INTO CARD-TYPE AT END GO TO ERR7.

```

MCVE CRIT TO NUMB.
IF CEKER IS NOT NUMERIC GO TO ERR4.
MCVE NUMM TO PLIT.
MCVE 'PLOT' TO LCRIT.
MCVE CARD-TYPE TO PRINT-LINE.
WRITE PRINT-REC AFTER POSITICNING NONE.
CFEN OUTPUT PLOT-TAPE.
ST5.
MCVE 1 TO CTR2.
ST6.
MCVE TIPE (CTR2) TO CCM-TIP.
GC TO
XC01 XC02 XC03 XC04 XC05 XC06 XC07 XC08 XC09 XC10
XC11 XC12 XC13 XC14 XC15 XC16 XC17 XC18 XC19 XC20
XC21 XC22 XC23 XC24 XC25 XC26
DEPENDING CN CGM-TIP.
GC TO ERR4.
TRUE.
ADD 1 TO CTR2.
GC TO ST6.
FALS.
GC TO ST5.
ERR1.
WRITE PRINT-REC FROM ERRCR1 AFTER POSITIONING NONE.
GC TO STOPPER.
ERR2.
WRITE PRINT-REC FROM ERRCR2 AFTER POSITIONING NONE.
GC TO STOPPER.
ERR3.
WRITE PRINT-REC FROM ERRCR3 AFTER POSITIONING NONE.
GC TO STOPPER.
ERR4.
WRITE PRINT-REC FROM ERRCR4 AFTER POSITIONING NONE.
GC TO STOPPER.
ERR5.
WRITE PRINT-REC FROM ERRCR5 AFTER POSITIONING NONE.
GC TO STOPPER.
ERR6.
WRITE PRINT-REC FROM ERRCR6 AFTER POSITIONING NONE.
GC TO END-STOP.
ERR7.
WRITE PRINT-REC FROM ERRCR7 AFTER POSITIONING NONE.
GC TO END-STOP.
ERR8.
MCVE NEW-MASTER TO CARD-PRINT.
MCVE ' PLOTTED VALUE EITHER ZERO OR BLANK ' TO MESSAGE-AREA.
WRITE PRINT-REC AFTER POSITIONING NONE.
ADD 1 TO LYNE.
GC TO TRUE.
STOPPER.
CLOSE PRINTER TRANS.
STOP RUN.
ERR-DATA.
IF LYNE IS GREATER THAN 53 PERFORM WRT-HEAD.
MCVE TRANS-CARD TO W-ERR.
WRITE PRINT-REC FROM DATA-BAD AFTER POSITIONING NONE.
ADD 1 TO LYNE.
GC TO ST5.
LIST-DEC.
ADD 1 TO R-CTR8.

```



```
IF RUN-T = 1
OR RUN-T = 3
GC TO LIST-PRT.
LIST-FT.
IF RUN-T = 2
OR RUN-T = 3
GC TO LIST-PLT.
GC TO TRUE.
LIST-FRT.
```

```
IF C-11 IS NOT = MERID
MCVE C-11 TO MERID
PERFORM WRT-HEAD.
IF LYNE IS GREATER THAN 53
PERFORM WRT-HEAD.
ADD 1 TO LYNE.
MCVE SPACES TO OUT-LINE.
MCVE CL12 TO LN1.
MCVE CL15 TO LN2.
MCVE CL17 TO LN3.
MCVE CL110 TO LN4.
CCMFUTE NM-ROCK = C-113 - C-125.
IF NM-ROCK = ZERO NEXT SENTENCE ELSE
MCVE NM-ROCK TO ED5M
MCVE ED5M TO LN28.
IF C120 = ZERO NEXT SENTENCE ELSE
MCVE C120 TO LN7.
IF C125 = ZERO NEXT SENTENCE ELSE
MCVE C-125 TO ED3P
MCVE ED3P TO LN8.
IF C124 = '2' MCVE '+' TO LNA MCVE '-' TO LN28-1.
PERFORM LITH-MOVER VARYING CTR5 FROM 1 BY 1 UNTIL
CTR5 IS GREATER THAN 5.
```

LIST2.

```
IF C163 = '0' GO TO LIST3.
IF C163 = '1' MOVE 'BLIND' TO L-PART
GC TO LIST3.
IF C164 = ZERO GC TO ERR-DATA.
MCVE C-164 TO ED3P
MCVE ED3P TO LN24.
IF C167 = ZERO GO TO ERR-DATA.
MCVE C-167 TO ED3P
MCVE ED3P TO LN25.
```

LIST3.

```
IF C170 = '0' GO TO LIST4.
IF C170 = '1'
MCVE 'FLOWING' TO E-PART
GC TO LIST4.
IF C171 = ZERO GO TO ERR-DATA.
MCVE C171 TO LN26.
IF C174 = ZERO GO TO ERR-DATA.
MCVE C174 TO LN27.
```

LIST4.

```
IF C113 = ZERO NEXT SENTENCE ELSE
MCVE C-113 TO ED4P
MCVE ED4P TO LN5.
IF C117 = ZERO NEXT SENTENCE ELSE
MCVE C117 TO LN6.
WRITE PRINT-REC FROM OUT-LINE AFTER POSITIONING NONE.
GC TO LIST-PT.
```

XC01.

IF C-11 IS GREATER THAN LOW-AL (CTR2)
OF C-11 IS EQUAL TO LOW-AL (CTR2)
IF C-11 IS LESS THAN HIGH-AL (CTR2)
OF C-11 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC02.

IF PRINT-T = 2 GO TO XC02-1.
IF CL12 IS GREATER THAN HIGH-AL (CTR2)
GC TO END-STOP1.

XC02-1.

IF CL12 IS GREATER THAN LCW-AL (CTR2)
OF CL12 IS EQUAL TO LCW-AL (CTR2)
IF CL12 IS LESS THAN HIGH-AL (CTR2)
OF CL12 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC03.

IF CL15 IS GREATER THAN LCW-AL (CTR2)
OF CL15 IS EQUAL TO LCW-AL (CTR2)
IF CL15 IS LESS THAN HIGH-AL (CTR2)
OF CL15 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC04.

IF CL17 IS GREATER THAN LCW-AL (CTR2)
OF CL17 IS EQUAL TO LCW-AL (CTR2)
IF CL17 IS LESS THAN HIGH-AL (CTR2)
OF CL17 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC05.

IF CL110 IS GREATER THAN LCW-AL (CTR2)
OF CL110 IS EQUAL TO LCW-AL (CTR2)
IF CL110 IS LESS THAN HIGH-AL (CTR2)
OF CL110 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC06.

IF C113 IS GREATER THAN LCW-AL (CTR2)
OF C113 IS EQUAL TO LCW-AL (CTR2)
IF C113 IS LESS THAN HIGH-AL (CTR2)
OF C113 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC07.

IF C117 IS GREATER THAN LCW-AL (CTR2)
OF C117 IS EQUAL TO LCW-AL (CTR2)
IF C117 IS LESS THAN HIGH-AL (CTR2)
OF C117 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC08.

IF C120 IS GREATER THAN LCW-AL (CTR2)
OF C120 IS EQUAL TO LCW-AL (CTR2)
IF C120 IS LESS THAN HIGH-AL (CTR2)
OF C120 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC09.

IF C124 IS GREATER THAN LCW-AL (CTR2)
OF C124 IS EQUAL TO LCW-AL (CTR2)
IF C124 IS LESS THAN HIGH-AL (CTR2)
OF C124 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.

XC10.

IF C125 IS GREATER THAN LCW-AL (CTR2)
OF C125 IS EQUAL TO LCW-AL (CTR2)

IF C125 IS LESS THAN HIGH-AL (CTR2)
OF C125 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE, GO TO FALS.

XC11.

MCVE ZEROS TO FLG1.
IF C131 IS GREATER THAN LOW-AL (CTR2)
OF C131 IS EQUAL TO LOW-AL (CTR2)
IF C128 IS LESS THAN HIGH-AL (CTR2)
OF C128 IS EQUAL TO HIGH-AL (CTR2)

MCVE 1 TO FG1.
IF C138 IS LESS THAN C131 GC TO XC11-1.
IF C138 IS GREATER THAN LOW-AL (CTR2)
OF C138 IS EQUAL TO LOW-AL (CTR2)
IF C135 IS LESS THAN HIGH-AL (CTR2)
OF C135 IS EQUAL TO HIGH-AL (CTR2)

MCVE 1 TO FG2.
IF C145 IS LESS THAN C131 GC TO XC11-1.
IF C145 IS GREATER THAN LOW-AL (CTR2)
OF C145 IS EQUAL TO LOW-AL (CTR2)
IF C142 IS LESS THAN HIGH-AL (CTR2)
OF C142 IS EQUAL TO HIGH-AL (CTR2)

MCVE 1 TO FG3.
IF C152 IS LESS THAN C131 GC TO XC11-1.
IF C152 IS GREATER THAN LOW-AL (CTR2)
OF C152 IS EQUAL TO LOW-AL (CTR2)
IF C149 IS LESS THAN HIGH-AL (CTR2)
OF C149 IS EQUAL TO HIGH-AL (CTR2)

MCVE 1 TO FG4.
IF C159 IS LESS THAN C131 GC TO XC11-1.
IF C159 IS GREATER THAN LOW-AL (CTR2)
OF C159 IS EQUAL TO LOW-AL (CTR2)
IF C156 IS LESS THAN HIGH-AL (CTR2)
OF C156 IS EQUAL TO HIGH-AL (CTR2)

MCVE 1 TO FG5.

XC11-1.

IF FLG1 IS EQUAL TO ZERC GC TO FALS. GO TO TRUE.

XC12.

MCVE ZERCS TO FLG2.
MCVE LOW-AL (CTR2) TO LOW-HLD.
MCVE HIGH-AL (CTR2) TO HGH-HLD.
MCVE C134 TO C134H.
MCVE C141 TO C141H.
MCVE C148 TO C148H.
MCVE C155 TO C155H.
MCVE C162 TO C162H.

PERFORM XC12-MV VARYING CTR8 FROM 1 BY 1 UNTIL CTR8 IS
GREATER THAN 5 AFTER CTR9 FROM 1 BY 1
UNTIL CTR9 IS GREATER THAN 4.

IF FLG2 IS EQUAL TO ZERC GC TO FALS. GO TO TRUE.

XC12-MV.

IF RL-HLD (CTR8) = LW-HLD (CTR9) OR RL-HLD (CTR8) >
LW-HLD (CTR9) IF RL-HLD (CTR8) = HG-HLD (CTR9) OR
RL-HLD (CTR8) < HG-HLD (CTR9)
MCVE 1 TO FLH (CTR8).

XC12-CK.

WRITE PRINT-REC FROM ERROR6 AFTER POSITIONING NONE.
GC TO END-STOP1.

XC13.

IF C163 IS GREATER THAN LOW-AL (CTR2)
OF C163 IS EQUAL TO LOW-AL (CTR2)

```

IF C163 IS LESS THAN HIGH-AL (CTR2)
OR C163 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.
XC14.
IF C167 IS GREATER THAN LCW-AL (CTR2)
OR C167 IS EQUAL TO LCW-AL (CTR2)
IF C164 IS LESS THAN HIGH-AL (CTR2)
OR C164 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.
XC15.
IF C170 IS GREATER THAN LCW-AL (CTR2)
OR C170 IS EQUAL TO LCW-AL (CTR2)
IF C170 IS LESS THAN HIGH-AL (CTR2)
OR C170 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.
XC16.
IF C174 IS GREATER THAN LOW-AL (CTR2)
OR C174 IS EQUAL TO LOW-AL (CTR2)
IF C171 IS LESS THAN HIGH-AL (CTR2)
OR C171 IS EQUAL TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.
XC17.
CCMPUTE NM-ROCK = C-113 - C-125.
IF ROCK-ELV IS GREATER THAN LOW-AL (CTR2)
OR ROCK-ELV IS = TO LOW-AL (CTR2)
IF ROCK-ELV IS LESS THAN HIGH-AL (CTR2)
OR ROCK-ELV IS = TO HIGH-AL (CTR2)
GC TO TRUE. GO TO FALS.
XC18.
GC TO ERR6.
XC19.
READ NEW-MAST INTO TRANS-CARD AT END GO TO END-STOP1.
IF REP-TIL NOT = IN-HLD GC TO RP-WRITE.
ADD 1 TO R-CTR9.
XC19-1.
IF INDEX-NO-TRANS = HIGH-VALUES GC TO ENDER.
GC TO TRUE.
XC20.
IF FL1 = 1 IF FG1 = 1 GC TO LIST-DEC.
IF FL2 = 1 IF FG2 = 1 GC TO LIST-DEC.
IF FL3 = 1 IF FG3 = 1 GC TO LIST-DEC.
IF FL4 = 1 IF FG4 = 1 GC TO LIST-DEC.
IF FL5 = 1 IF FG5 = 1 GC TO LIST-DEC.
GC TO FALS.
XC21.
GC TO ST5.
XC22.
GC TO ERR6.
XC23.
GC TO ERR6.
XC24.
GC TO ERR6.
XC25.
GC TO ERR6.
XC26.
GC TO ERR6.
RP-WRITE.
IF PRINT-T = 1 GO TO XC19-1.
IF IN-HLD = ZERCS MOVE REP-TIL TO IN-HLD
ADD 1 TO R-CTR9 GO TO XC19-1.

```

```

MCVE IN-M TO R-MERC.
MCVE IN-TP TO R-TP.
MCVE IN-RG TO R-RG.
MCVE F-CTR9 TO R-CCUNT.
MCVE F-CTR8 TO RT-CCUNT.
WRITE REPREC FROM REP-WORK.
MCVE ZEROS TO R-CTR9 R-CTR8.
MCVE REP-TIL TO IN-HLD.
ADD 1 TO R-CTR9.
GC TO XC19-1.
END-STOP.
MCVE BLANKZ TO PRINT-LINE.
WRITE PRINT-REC AFTER POSITIONING SKIP.
CLCSE REPFIL.
IF PRINT-T = 1 GO TO REF-CN-2.
OPEN INPUT REPFIL.
REP-AGN.
READ REPFIL AT END GO TO REF-ON.
MCVE REPREC TO PRINT-LINE.
WRITE PRINT-REC AFTER POSITIONING NONE.
GC TO REP-AGN.
REP-ON.
CLCSE REPFIL.
REP-ON-2.
CLCSE PRINTER TRANS NEW-MAST.
STCF RUN.
END-STOP1.
IF RUN-T = 2 OR RUN-T = 3 CLCSE PLOT-TAPE.
GC TO END-STOP.
ENDER.
IF PRINT-T = 1 GO TO END-STCF1.
ENDER-1.
READ NEW-MAST AT END GO TO END-STOP1.
GC TO ENDER-1.
LIST-PLT.
MCVE INDEX-NC-TRANS TO PLIT-INDEX.
MCVE N-MILE TO N-SUB.
ADD 1 TO N-SUB.
MCVE W-MILE TO W-SUB.
ADD 1 TO W-SUB.
MCVE TTABLE (N-SUB, W-SUB) TO SQ-MILE.
IF N-HALF IS > '49' MOVE '2' TO HF-MILE ELSE
    MOVE '1' TO HF-MILE.
GC TO
P01 P02 P03 P04 P05 P06 P07 P08 P09 P10
P11 P12 P13 P14 P15 P16 P17 P18 P19 P20
P21
DEPENDING ON PLIT.
GC TO ERR3.
P01.
MCVE C-11 TO PLT-ENTRY.
GC TO TRUE1.
P02.
MCVE CL12 TO PLT-ENTRY.
GC TO TRUE1.
P03.
MCVE CL15 TO PLT-ENTRY.
GC TO TRUE1.
P04.
MCVE CL17 TO PLT-ENTRY.

```

```

GC TO TRUE1.
P05.
MCVE CL110 TO PLT-ENTRY.
GC TO TRUE1.
P06.
CCMPUTE TMHOLD = C-113 + FIVE.
MCVE TMHOLD TO P-EDT.
MCVE P-EDIT TO PLT-ENTRY.
GC TO TRUE1.
P07.
MCVE C117 TO PLT-ENTRY.
GC TO TRUE1.
P08.
MCVE C120 TO PLT-ENTRY.
GC TO TRUE1.
P09.
MCVE C124 TO PLT-ENTRY.
GC TO TRUE1.
P10.
CCMPUTE P-SURF = C-125 + FIVE.
IF C124 = 1 MCVE P-SURF TO P-EDT MCVE P-EDIT TO PLT-ENTRY
ELSE MCVE P-SURF TO M-ECT MCVE M-EDIT TO PLT-ENTRY.
GC TO TRUE1.
P11.
IF FL1 = 1
IF FG1 = 1 MOVE C131 TO P-ECT MCVE P-EDIT TO PLT-ENTRY.
IF FL2 = 1
IF FG2 = 1 MOVE C138 TO P-ECT MCVE P-EDIT TO PLT-ENTRY.
IF FL3 = 1
IF FG3 = 1 MOVE C145 TO P-ECT MCVE P-EDIT TO PLT-ENTRY.
IF FL4 = 1
IF FG4 = 1 MOVE C152 TO P-ECT MCVE P-EDIT TO PLT-ENTRY.
IF FL5 = 1
IF FG5 = 1 MOVE C159 TO P-ECT MCVE P-EDIT TO PLT-ENTRY.
GC TO TRUE1.
P12.
MCVE SPACES TO HD-PICT.
IF FLG1 = 11111
IF FLG2 = 11111 WRITE PRINT-REC FROM LITH-KODE
AFTER POSITIONING NONE
ADD 1 TO LYNE
MCVE 0 TO FL1.
MCVE 1 TO CTR9.
PERFORM P12-MV VARYING CTR8 FROM 1 BY 1 UNTIL CTR8
IS GREATER THAN 5.
MCVE FD-PICT TO PLT-ENTRY GO TO TRUE1.
P12-MV.
IF FLH1 (CTR8) = 1
IF FLH (CTR8) = 1 MOVE RL-HLD (CTR8) TO HD-PIC (CTR9)
ADD 1 TO CTR9.
P12-MVI.
GC TO XC12-CK.
P13.
MCVE C163 TO PLT-ENTRY.
GC TO TRUE1.
P14.
MCVE C167 TO PLT-ENTRY.
GC TO TRUE1.
P15.
MCVE C170 TO PLT-ENTRY.

```

```

GC TO TRUE1.
P16.
MCVE C174 TO PLT-ENTRY.
P17.
CCMFUTE P-SURF = C-113 - C-125 + FIVE.
IF C124 = 1 MOVE P-SURF TO P-EDT
MCVE P-EDIT TO PLT-ENTRY ELSE
MCVE P-SURF TO M-EDT
MCVE M-EDIT TO PLT-ENTRY.
GC TO TRUE1.

P18.
P19.
P20.
P21.
GC TO ERR3.

TRUE1.
IF FLT-ENTRY = ZERO OR
PLT-ENTRY = SPACES GO TO ERR8.
WRITE FLITER FROM PLIT-CD.
GC TO TRUE.

LITH-MOVER SECTION.
LTH1.
IF LITH-FROM (CTR5) = ZERC AND
LITH-TO (CTR5) = ZERC GO TO LTH-EX.
IF LITH-TO (CTR5) IS LESS THAN LITH-FROM (CTR5)
GC TO ERR-DATA.
IF CTR5 = 1 GO TO LTH2.
CCMFUTE CTR6 = CTR5 - 1.
IF LITH-FROM (CTR5) IS LESS THAN LITH-TO (CTR6)
GC TO ERR-DATA.

LTH2.
MCVE LITH-FROM (CTR5) TO LT-FM (CTR5).
MCVE LITH-TO (CTR5) TO LT-TC (CTR5).
CCMFUTE CMNT = C-125 - 5.
IF C124 = '1'
IF LITH-CODE (CTR5) IS GREATER THAN '0'
IF LITH-CODE (CTR5) IS LESS THAN '4'
IF CMPT IS LESS THAN LITH-TC (CTR5)
MCVE '*' TO ENCT (CTR5).
PERFORM LTH-CD-MOVE.

LTH-EX. EXIT.
LTH-CD-MOVE SECTION.
LH1.
IF LITH-CODE (CTR5) = '1' MCVE 'GVL' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '2' MCVE 'S&G' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '3' MCVE 'SND' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '4' MCVE 'COL' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '5' MCVE 'SDS' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '6' MCVE 'SHS' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '7' MCVE 'SSS' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '8' MCVE 'S&C' TO LT-CD (CTR5)
GC TO LH-EX.
IF LITH-CODE (CTR5) = '9' MCVE 'SSC' TO LT-CD (CTR5)

```

GC TO LH-EX.
LH-EX. EXIT.

\$C RPLCT

IDENTIFICATION DIVISION.
PROGRAM-ID. 'MAPPING'.
AUTHOR. NUCLEUS.
INSTALLATION. U OF A COMPUTING CENTER.

DATE-WRITTEN. JULY 1967.
DATE-COMPILED. JULY 1967.
SECURITY. CONFIDENTIAL.
REMARKS. PLOTTING PROGRAM.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IEM-360.
OBJECT-COMPUTER. IEM-360.
INPUT-OUTPUT SECTION.

FILE-CONTROL.
SELECT PRINTER ASSIGN TO 'SYSOUT' UNIT-RECORD.
SELECT INPUT ASSIGN TO 'FT01F001' UTILITY.

DATA DIVISION.

FILE SECTION.

FD PRINTER RECORD CONTAINS 133 CHARACTERS

LABEL RECORDS ARE OMITTED
RECORDING MODE IS F
DATA RECORD IS PRINT-REC.

01 PRINT-REC.
05 FILLER PICTURE X.
05 F-NAME PICTURE X(132).

FD INPUT RECORD CONTAINS 20 CHARACTERS
BLOCK CONTAINS 40 RECORDS
LABEL RECORDS ARE STANDARD

RECORDING MODE IS F
DATA RECORD IS INREC.

01 INREC.
05 FILLER PICTURE X(20).

WORKING-STORAGE SECTION.

77 DUMM PICTURE S9999 COMPUTATIONAL VALUE ZERO.
77 INDEX-SAVE PICTURE X(6).
77 D4 PICTURE S99999 COMPUTATIONAL.
77 D5 PICTURE S99999 COMPUTATIONAL.
77 D6 PICTURE S99999 COMPUTATIONAL.
77 D7 PICTURE S99999 COMPUTATIONAL.
77 D8 PICTURE S99999 COMPUTATIONAL.
77 D9 PICTURE S99999 COMPUTATIONAL.
77 D10 PICTURE X(4).
77 CTR-1 PICTURE S9 COMPUTATIONAL.
77 NUM-PLOTS PICTURE S99 COMPUTATIONAL VALUE ZERO.
77 PLOT-TIMES PICTURE S99 COMPUTATIONAL VALUE ZERO.
01 FLOT-IN.

05 INDEX.
07 C01 PICTURE X.
07 C02 PICTURE XXX.
07 C05 PICTURE XX.
05 NORTH.
07 C07 PICTURE 9.
07 C08 PICTURE 99.
05 WEST.
07 C10 PICTURE 9.
07 C11 PICTURE 99.
05 SQ-MILE PICTURE 99.
05 SIDE PICTURE 9.

```

05 F-DATA.
07 F-1 PICTURE X.
07 F-2 PICTURE X.
07 P-3 PICTURE X.
07 P-4 PICTURE X.
05 FILLER PICTURE X.

01 ERR1.
03 FILLER PICTURE X(48) VALUE IS
' ERROR-1 DATA PASSED WAS BLANK --- NOT PLOTTED'.
03 PLCT-IMAGE PICTURE X(20).

01 D-ALL.
05 FILLER PICTURE X(3) VALUE 'MER'.
05 D1 PICTURE X.
05 FILLER PICTURE X(3) VALUE ' TP'.
05 D2 PICTURE X(3).
05 FILLER PICTURE X(3) VALUE ' RG'.
05 D3 PICTURE XX.
05 FILLER PICTURE X VALUE SPACE.
05 P-Y PICTURE ZZ9.
05 FILLER PICTURE X(15) VALUE IS
' POINTS PLOTTED'.

PROCEDURE DIVISION.
START SECTION.
BEGIN.
    MOVE ZERO TO NUM-PLOTS.
    OPEN INPUT INPUT OUTPUT PRINTER.
    READ INPUT INTO PLOT-IN AT END GO TO FINISH.
    ENTER LINKAGE.
    CALL 'INIT' USING DUMM.
    ENTER COBOL.
    GO TO HEAD-WRITE.

LOOP.
    READ INPUT INTO PLOT-IN AT END GO TO FINISH.
    IF INDEX NOT = INDEX-SAVE GO TO SEE-4.

A5.
    MOVE C07 TO D4.
    MOVE C08 TO D5.
    MOVE C10 TO D6.
    MOVE C11 TO D7.
    MOVE SQ-MILE TO D8.
    MOVE SIDE TO D9.
    IF P-DATA = SPACES GO TO ERRCR-1.
    IF F-1 = '-' GO TO CCNT.
    IF D9 = 2 GO TO A2.
    IF F-1 = '.' MOVE P-2 TO P-1
    MOVE P-3 TO P-2 MOVE P-4 TO P-3
    MOVE SPACE TO P-4. GO TO CCNT.

A2.
    IF F-1 = '.' MOVE SPACE TO P-1 GO TO CCNT.

A3.
    IF P-4 NOT = SPACE GO TO CCNT.
    MOVE P-3 TO P-4 MOVE P-2 TO P-3
    MOVE P-1 TO P-2 MOVE SPACE TO P-1.
    GO TO A3.

CCNT.
    MOVE P-DATA TO D10.

AL.
    ENTER LINKAGE.
    CALL 'FPLOT' USING D4 D5 D6 D7 D8 D9 D10.
    ENTER COBOL.

```



```

      ADD 1 TO PLOT-TIMES.
      GO TO LOOP.
SEE-4.
      MOVE PLOT-TIMES TO P-T.
      MOVE ZERO TO PLOT-TIMES.
      MOVE D-ALL TO P-NAME.
      WRITE PRINT-REC AFTER 1.
      ENTER LINKAGE.
      CALL 'WLEFT' USING DUMM.
      ENTER COBOL.
      ADD 1 TO NUM-PLOTS.
      IF NUM-PLOTS = 4 GO TO RESET-PEN.
      ENTER LINKAGE.
      CALL 'MLEFT' USING DUMM.
      ENTER COBOL.
HEAD-WRITE.
      MOVE C01 TO D1.
      MOVE C02 TO D2.
      MOVE C05 TO D3.
      ENTER LINKAGE.
      CALL 'WRIGHT' USING D-ALL.
      ENTER COBOL.
      MOVE INDEX TO INDEX-SAVE.
      GO TO A5.
RESET-PEN.
      MOVE ZERO TO NUM-PLOTS.
      ENTER LINKAGE.
      CALL 'MRIGHT' USING DUMM.
      ENTER COBOL.
      GO TO HEAD-WRITE.
FINISH.
      ENTER LINKAGE.
      CALL 'FINISH' USING NUM-PLOTS.
      ENTER COBOL.
      MOVE ' PLCT COMPLETED O.K.' TO PRINT-REC.
      WRITE PRINT-REC AFTER 0.
      CLOSE PRINTER INPUT.
      STOP RUN.
ERRCR-1.
      MOVE INREC TO PLCT-IMAGE.
      WRITE PRINT-REC FROM ERR1 AFTER 0.
      GO TO LOOP.

```

/*

```

//XY EXEC FCRTHCL,PARM,FORT='DECK'
//FORT.SYSPLNCH DD SYSOUT=B
//FORT.SYSIN DD *
      SUBROUTINE INIT(I)
      DIMENSION A(1024)
      CALL FLCTS (A(1),4096)
      CALL FLCT (1.0,0.0,23)
      RETURN
      END
      SUBROUTINE WLEFT (I)
      CALL FLCT(6.0,5.5,3)
      CALL FLCT (6.0,6.0,2)
      CALL FLCT (5.5,6.0,2)
      RETURN
      END
      SUBROUTINE MLEFT (I)
      CALL FLCT (0.0,6.0,-3)

```

```
RETURN  
END  
SUBROUTINE MRIGHT (I)  
CALL FLCT(7.0,-18.0,-3)  
RETURN  
END
```

```
SUBROUTINE FINICH (NUM)  
INTEGER*2 NUM  
I = NUM  
F = I * 6 * (-1)  
CALL WLEFT (I)  
CALL FLCT (7.0,F,-3)  
CALL FLCT (0.0,0.0,999)  
RETURN  
END
```

```
SUBROUTINE WRIGHT (HCL)  
DIMENSION HCL (4)  
CALL SYMBCL(-0.35,3.0,0.20,HCL,270.0,16)  
CALL FLCT (0.5,0.0,3)  
CALL FLCT(0.0,0.0,2)  
CALL FLCT (0.0,0.5,2)  
RETURN  
END
```

```
SUBROUTINE FPLOTT(I4,I5,I6,I7,I8,L9,I10)  
DATA N4,N5,N6,N7,N8,N9,N10,N1,N2/0,0,0,0,0,0,0,0,0/
```

```
21 L6 = 6  
I9 = L9  
22 L1 = 1  
24 IF (N8 .NE. I8) GO TO 6  
25 IF (N9 .NE. I9) GO TO 6  
26 IF (N4 .NE. I4) GO TO 6  
27 IF (N6 .NE. I6) GO TO 6  
28 IF (I5 .GE. N5) GO TO 9  
29 GO TO 10  
8 IF (N1 .GE. L6) GO TO 6  
30 IF (I7 .GE. N7) GO TO 11  
31 GO TO 12  
20 IF (N2 .GE. L6) GO TO 6  
32 IF (I9 .EQ. L1) GO TO 7  
33 I9 = 1  
GO TO 6  
9 N1 = I5 - N5  
GO TO 8  
10 N1 = N5 - I5  
GO TO 8  
11 N2 = I7 - N7  
GO TO 20  
12 N2 = N7 - I7  
GO TO 20  
7 I9 = 2  
6 N8 = I8  
N9 = I9  
N4 = I4  
N6 = I6  
N5 = I5  
N7 = I7  
THETA = 315.00  
FI4 = I4  
FI6 = I6  
FI5 = I5
```

FI5 = FI5 / 100.00

FI7 = I7

FI7 = FI7 / 100.00

X = FI4 + FI5

Y = FI6 + FI7

3 X1 = X + .02

Y1 = Y + .02

CALL FLCT (X,Y,3)

CALL FLCT (X,Y1,2)

CALL FLCT (X1,Y1,2)

CALL FLCT (X1,Y,2)

CALL PLCT (X,Y,2)

IF (I9 .EQ. 1) GO TO 4

X = X - .15

Y = Y + .15

GO TO 2

4 X = X + .02

Y = Y - .02

2 CONTINUE

CALL SYMBOL(X,Y,0.06,I10,THETA,4)

N1 = N2

RETURN

END