

ANL-79-71

Volume I

49. 135

ANL-79-71

Volume II

**AN INTERIM REPORT ON THE
DEVELOPMENT AND APPLICATION OF
ENVIRONMENTAL MAPPED DATA DIGITIZATION,
ENCODING, ANALYSIS, AND DISPLAY SOFTWARE
FOR THE ALICE SYSTEM**

Volume II

by

**L. W. Amiet, R. J. Lima, S. D. Schellbrock,
C. B. Shelton, and R. H. Weisman**

MASTER

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) among the U. S. Department of Energy, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona
Carnegie-Mellon University
Case Western Reserve University
The University of Chicago
University of Cincinnati
Illinois Institute of Technology
University of Illinois
Indiana University
The University of Iowa
Iowa State University

The University of Kansas
Kansas State University
Loyola University of Chicago
Marquette University
The University of Michigan
Michigan State University
University of Minnesota
University of Missouri
Northwestern University
University of Notre Dame

The Ohio State University
Ohio University
The Pennsylvania State University
Purdue University
Saint Louis University
Southern Illinois University
The University of Texas at Austin
Washington University
Wayne State University
The University of Wisconsin-Madison

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use or the results of such use of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. Mention of commercial products, their manufacturers, or their suppliers in this publication does not imply or constitute approval or disapproval of the product by Argonne National Laboratory or the United States Government.

Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5200 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy \$05
Microfilm copy \$05

Distribution Category:
Mathematics and Computers
(UC-32)

ANL-79-71
Volume II

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

AN INTERIM REPORT ON THE DEVELOPMENT AND APPLICATION
OF ENVIRONMENTAL MAPPED DATA DIGITIZATION, ENCODING,
ANALYSIS, AND DISPLAY SOFTWARE FOR THE ALICE SYSTEM

Volume II

by

L. W. Amiot, R. J. Lima,* S. D. Scholbrock,*
C. B. Shelton, and R. H. Wehman

Applied Mathematics Division

NOTICE
This report was prepared as an account of work
sponsored by the United States Government. Rights to
reproduce and to disseminate the information contained
herein are reserved by the United States Government
and are not to be distributed outside the Government
without the express written permission of the
Government.

June 1979

*Office of: Robert J. Lima, Box 8584, Chicago, Illinois 60680

TABLE OF CONTENTS

	<u>Page</u>
<u>VOLUME I</u>	
ABSTRACT	ix
Introduction	ix
1.0 THE ALICE SYSTEM	1
1.1 Overview	1
1.2 Hardware	1
1.3 Software	2
1.4 Input/Output	2
2.0 SOFTWARE OVERVIEW	5
2.1 Foundations, Modules and Tasks	5
2.2 MAP	5
2.3 CHAIN	5
2.4 FIX	6
2.5 DOUT	6
3.0 MAP: A PROGRAM TO DIGITIZE AND ENCODE MAPPED INFORMATION	7
3.1 Background	7
3.2 Overview of Programming Foundation	7
3.3 Operating Environment and Foundation Logic	7
3.3.1 Map preprocessing and initialization	7
3.3.2 Map digitization	8
3.3.3 Matrix registration	11
3.3.4 Polygon generation	12
3.3.5 Polygon editing	14
3.3.6 On-line attribute identification	15
3.3.7 Deferred attribute identification (ATT)	15
3.3.8 Polygon output	16
4.0 CHAIN: A PROGRAM TO GENERATE CHAIN-AND-NODE DATA	19
4.1 Overview of Programming Foundation	19
4.2 Operating Environment and Foundation Logic	19
4.2.1 Input mode	19
4.2.2 Chain generation	21
4.2.3 Node generation	23
4.2.4 Data cell generation	24
4.2.5 Output mode	24
5.0 FIX: A PROGRAM TO REASSEMBLE AND EDIT SCANNED MAP CELLS	26
5.1 Background	26
5.2 Overview of Programming Foundation	26
5.3 Operating Environment	28
5.4 Foundation Logic	28
5.5 Productivity and Efficiency	32

TABLE OF CONTENTS

	<u>Page</u>
6.0 DOUT: A PROGRAM TO FORMAT AND OUTPUT ALICE DATA	33
6.1 Background	33
6.2 Overview of Programming Foundation	33
6.3 Operating Environment	33
6.4 Foundation Logic	34
6.5 Productivity and Efficiency	34
7.0 APPLICATIONS OF THE SOFTWARE	35
7.1 An Example	35
7.2 Map Preprocessing	35
7.3 Cell-Matrix Superimposition	38
7.4 Data Processing	38
7.5 Analysis and Display	38
7.6 Final Output	41
8.0 FUTURE RESEARCH AND DEVELOPMENT	44
ACKNOWLEDGEMENTS	46
REFERENCES	48
<u>VOLUME II</u>	
Preface	ix
APPENDIX A - CODE LISTINGS	A-1
APPENDIX B - OPERATIONAL KEY/SUBROUTINE TABLES	B-1

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Hardware Organization of the ALICE System	3
2.	Artificial Matrix of Map Cells for Use in Scanning	9
3.	Example of Base Map with Artificial Matrix Superimposed for Scanning	10
4.	Example of Density Histogram from Scanned Data in MAP	11
5.	Polygons Defined by Horizontal Line Segments in MAP	12
6.	Display of Map Cell-Matrix Markers for Operator Registration in MAP	13
7.	Plot of Line Segments with a Common Tag Number in MAP	13
8.	Logic Testing for Desired Polygons vs. "Noise" in MAP	14
9.	Filtering of Polygon-Boundary "Noise" in MAP	15
10.	Deferred Attribute Identification (ATT) in MAP	17
11.	MAP vs. CHAIN Outputs	20
12.	Chain-and-Node Data Components	21
13.	Concept of Chain Generation in CHAIN	21
14.	Software Generated Display of Chains with Nodes Delineated with an "X " in CHAIN	24
15.	Data Cell "Windowing" in CHAIN	25
16.	FIX Data Flow	29
17.	FIX Foundation Operational Flow Chart	30
18.	Example of DOUT Output	34
19.	Source Map (Example): Soil Information - Base Map	36
20.	Source Map (Example): Soil Information after Preprocessing	37
21.	Composite Map (Example): Soil Information with Superimposed Cell Matrix	39
22.	Composite Map (Example): Sized to the 3 x 4 Scanning, Cell-Matrix Parameter	40

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
23.	ALICE Data (Example): Computerized Information from Source Map Interpreted for Conditions for Septic Tank Filter Fields Based on Soil Types	42
24.	ALICE Data (Example): Computerized Information from Source Map Quantified by ALICE	43

LIST OF TABLES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	MAP: Teletype Key Commands	B-1
2.	MAP: Subroutine Functions	B-5
3.	CHAIN: Subroutine Functions	B-8
4.	FIX: General Map Information	B-9
5.	FIX: Scanned Cell File Information	B-11
6.	FIX: Current Program Information	B-13
7.	FIX: Teletype Key Commands for Primary (Cell Editing) Mode . .	B-15
8.	FIX: Teletype Key Commands for Secondary (Chain Editing) Mode	B-20
9.	FIX: Sense Switch Options	B-22
10.	FIX: Subroutine Functions	B-25
11.	FIX: Data Organization	B-30

Preface

Volume I of, An Interim Report on the Development and Application of Environmental Mapped Data Digitization, Encoding, Analysis and Display Software for the ALICE System, provided an overall description of the software developed for the ALICE System and presented an example of its application. Whereas, the scope of the information presented in Volume I was directed both to the users and developers of digitization, encoding, analysis and display software, Volume II presents information which is directly related to the actual computer code and operational characteristics (keys and subroutines) of the software.

The authors expect that Volume II will be of more interest to developers of software than to users of the software. However, developers of software should be aware that the code developed for the ALICE System operates in an environment where much of the peripheral hardware to the PDP-10 is ANL/AMD built. For this reason, portions of the code may have to be modified for implementation on other computer system configurations.

Questions and comments relating to the code can be directed to:

Robert J. Lima
City and Regional Planner
c/o Applied Mathematics Division
Argonne National Laboratory
9700 South Cass Avenue
Room B-127
Argonne, Illinois 60439

****ALICE FOUNDATION SUMMARY****

Foundation Title: MAP

Date of Construct: 1978

Code Format: Fortran IV

Operating Computer: DEC PDP-10

Size Required: 55K

Number of Subroutines: 13

Foundation/Module/Task Summary:

- o The MAP foundation accepts mapped information recorded on hard copy. This foundation searches, digitizes and edits all polygons and then outputs ALICE polygon data to 9 track magnetic tape.

Type of Input Data Required by Foundation:

Map information on hard copy

Type of Output Data Required by Foundation:

ALICE polygons on 9 track magnetic tape

References:

Section 3.0 of this report, Volume I; Table 1 & 2 of this report, Volume II


```

76 CONTINUE
77 CONTINUE
100 GO TO 99
C
101 GO TO 44      A IS OCTAL 101
C
102 GO TO 99      B IS OCTAL 102
C
103 CONTINUE      C IS OCTAL 103
      GO TO 99      CHANGE W02 OR V07 TO THE OTHER MODE
C
104 CONTINUE      D IS OCTAL 104
      GO TO 99      CHANGE DISCRIMINATOR LEVEL
      CALL L01DID(0.510-DEML)
      ASSIGN 1040 TO CONW04
      GO TO 199
C
1043 CALL R01DIB(TDIN)
      TDIN=IRICH(TDIN)/512
      IF(TDIN.LT.64)GO TO 1041
      IBIN=63
      CALL L01DID(0.510-63)
1041 CONL=TDIN
      GO TO 1991
C
105 IF (HVFLAG.EQ.0) NOL=XH-NOL*YFI
      IF (HVFLAG.EQ.1) NOL=XH-NOL*XLI
      CALL L01DIB(XI.0)
      ASSIGN 1050 TO CONW04
      GO TO 199
C
1050 CALL R01DIB(TDIN)
      XR=ILEFT(TDIN)
      IF (XR.LE.X) GO TO 1051
      IF (HVFLAG.EQ.0) NOL=(XH-X)/XFI
      IF (HVFLAG.EQ.1) NOL=(XH-X)/XLI
      GO TO 1991
1051 IF (HVFLAG.EQ.0) NOL=2
      IF (HVFLAG.EQ.1) NOL=2
      GO TO 1991
C
106 GO TO 99      F IS OCTAL 106
C
107 CONTINUE      G IS OCTAL 107
      CALL IDENK
      NOL=(XCH(XQP,YQP)*330)/200
      NOL=(YCH(XQP,YQP)*330)/200
      QPAP(1,5)=NOL
      QPAP(1,6)=NOL
      GO TO 99
C
110 CALL RCHAR (IN)
      IF (IN.EQ.0) GO TO 110
      CALL SHANI
      IF (IN.EQ.48) CALL STATION(0)
      IF (IN.EQ.49) CALL STATION(1)
      IF (IN.EQ.50) CALL STATION(2)
      IF (IN.EQ.51) CALL STATION(3)
      GO TO 999

```

```

C          I IS OCTAL 111
C  RE-INITIALIZE COUNTING PARAMETERS
111  CALL WTD 107,200
     IF (IN.LE.1).OR.(IN.GT.4) GO TO 99
     GO 107,201,10
112  WTD(I)=COUNT(I)
     GO TO 99
C          J IS OCTAL 112
C  SAVE CURRENT PARAMETERS I FOR I NEXT
112  CALL WTD 107,200
     IF (IN.LE.1).OR.(IN.GT.4) GO TO 99
     GO 107,201,10
113  WTD(I)=COUNT(I)
     GO TO 99
C          K IS OCTAL 113
113  CONTINUE
     GO TO 99
C          L IS OCTAL 114
114  WTD 104
     CALL PUNCH(1)
     EXCEPTS,1040,FRAMES
     CALL INITIATE.PHONES
     READY=LATCH
     LINE RELEASE 194
     GO TO 99
115  WTD 105
     CALL PUNCH(1)
     EXCEPTS,1040,FRAMES
     CALL INITIATE.PHONES
     READY=LATCH
     LINE RELEASE 194
     GO TO 99
C          M IS OCTAL 115
115  WTD 106
     CALL PUNCH(1)
     EXCEPTS,1040,FRAMES
     CALL OFFICE (P,TRACES)
     WTD 104 MTRM
     CALL PUNCH(1)
     GO TO 99
C          N IS OCTAL 116
C  CHANGE (NORTH) BOUNDARY
116  IF (IN.FLAG.EQ.0) NH=Y+NDL+YLI
     IF (IN.FLAG.EQ.1) NH=Y+NOP+YPI
     CALL I.PHOTO(1)
     GO TO 106 TO CONTINUE
     GO TO 99
C          CODE WHEN NH CONNECTED TO TS
1160 CALL READBITBIN)
     NH=IRIGHTBIN)
     IF (NH.LE.Y) GO TO 1161
     IF (IN.FLAG.EQ.0) NDL=(YH-Y)/YLI
     IF (IN.FLAG.EQ.1) NOP=(YH-Y)/YPI
     GO TO 1991
1161 IF (IN.FLAG.EQ.0) NDL=2
     IF (IN.FLAG.EQ.1) NOP=2
     GO TO 1991
C          O IS OCTAL 117
117  GO TO 99
C          P IS OCTAL 120
120 CONTINUE
1200 CALL COM(RN)
     IF (N.EQ.0) GO TO 1202
     COUNTING
     IF (N.GE.49).AND.(N.LE.51) GO TO 300

```

```

IF (IN.EQ.28).OR.(IN.EQ.29)) GO TO 1288
IF (IN.EQ.72) GO TO 1272
IF (IN.EQ.73) GO TO 1273
IF (IN.EQ.24) GO TO 400
GO TO 599

```

```

C                                     FN
1272 CONTINUE
CALL PLPTND (XP,YP)
IF (XP.LT.5).OR.(YP.LT.5)) GO TO 599
302 IF (XP.LG.14.0) GO TO 502
CALL G510P
YD=1
YD=1
CALL PLRQNE
1273 CALL PLRQNE
CALL PLRQNE
GO TO 502

```

```

C                                     QUANTER SECTION
C                                     POSITIONING
C                                     OFD
C                                     4 * * *
C                                     3 * * *
C                                     2 * * *
C                                     1 * * *
C                                     1 2 3

```

```

300 XPOS=N-49
310 CALL PCHP (AL)
IF (AL.GT.0) GO TO 310
C                                     P11-3111-4)
IF (AL.LT.49).OR.(AL.GT.52)) GO TO 310
YPOS=AL-49
CALL MOVEPL (XPOS,YPOS)
GO TO 402
311 CONTINUE
C=XPOS-20+YPOS-11+200
Y=YPOS-11+400
CALL PLRQEXP (XP)
CALL PLRQNE
YD=2000
YD=YD+20
GO TO 599

```

```

C                                     FT
C                                     MOVE PL TRANSPORT WITH TRACKBALL

```

```

400 CONTINUE
FLU=1
RETURN
401 CONTINUE
CALL RREAD (XP,YP)
YD=YP
YD=YP
YD=YP/2
CALL PLTB (XP,YD)
FLU=2
GO TO 902
410 CONTINUE
CALL GRID (XP,YD)
YD=YD+2
IF ((XP.LT.5).OR.(YP.LT.5)) GO TO 450
IF ((XP.GT.300).OR.(YP.GT.1793)) GO TO 450
IF ((ABS(XP-XP).GT.300)) GO TO 450

```



```

1230 CALL READTB(TBIN)
      YL=IRIGHT(TBIN)
      IF (YL.GE.YH) GO TO 1231
      Y=YL
      IF (HVFLAG.EQ.0) NOL=(YH-Y)/YLI
      IF (HVFLAG.EQ.1) NOP=(YH-Y)/YPI
      GO TO 1231
1231 IF (HVFLAG.EQ.0) NOL=2
      IF (HVFLAG.EQ.0) Y=YH-2*YLI
      IF (HVFLAG.EQ.1) NOP=2
      IF (HVFLAG.EQ.1) Y=YH-2*YPI
      GO TO 1231
C           T IS OCTAL 124
124  CONTINUE
      GO TO 99
C
C           U IS OCTAL 125
C TYING CHANGES IN SCANNING PARAMETERS
125  DO 1250 I=1,11
      TYPE 666,I
      TYPE 667,APARS(I)
      CALL RDEC (QF,INPUT)
      IF (QF.EQ.0) GO TO 1250
      APARS(I)=INPUT
1250 CONTINUE
      GO TO 99
C           V IS OCTAL 126
126  CONTINUE
      TYPE 1713
1713 FORMAT ( > POSITION MAGTAPE TO LAST ENTRY > )
      TYPE 1714
1714 FORMAT ( > IDEN= >4)
      CALL RDEC(QF,IDEN)
      FHS=3
      TYPE 1700,IDEN
      TYPE 1701
1700 FORMAT('13)
1701 FORMAT( > IF OK TYPE A IF NOT TYPE Z >)
1716 CALL RCMR(D)
      IF(D.EQ.65) GO TO 1717
      IF(D.EQ.90) GO TO 99
      GO TO 1716
1717 CALL HTRP(PERSV,550)
      IF(PERSV(1).NE.IDEN)GO TO 1717
      IF(PERSV(2).NE.FHS)GO TO 1717
      GO TO 99
C           W IS OCTAL 127
C CHANGE WEST(LHS) BOUNDARY
127  CALL LOADTB(X,0)
      IF (HVFLAG.EQ.0) XH=X*NOP*XPI
      IF (HVFLAG.EQ.1) XH=X*NOL*XLI
      ASSIGN 1270 TO CONISH
      GO TO 159
C           CODE WHEN >W< CONNECTED TO TB
1270 CALL READTB(TBIN)
      XL=ILEFT(TBIN)
      IF (XL.GE.XH) GO TO 1271
      X=XL
      IF (HVFLAG.EQ.0) NOP=(XH-X)/XPI
      IF (HVFLAG.EQ.1) NOL=(XH-X)/XLI

```

```

GO TO 1991
1271 IF (HVFLAG.EQ.0) NOP=2
    IF (HVFLAG.EQ.0) X=XH-2*XPI
    IF (HVFLAG.EQ.1) NOL=2
    IF (HVFLAG.EQ.1) X=XH-2*XLI
GO TO 1991
C          X IS OCTAL 130
C          CHANGE X INCREMENT
130 CALL LOADTB(64*(XLI+XPI),0)
    ASSIGN 1300 TO CONNSN
GO TO 129
C          CODE WHEN >X> CONNECTED TO TB
1300 CALL READTB(TBIN)
    TX=ILEFT(TBIN)/64
    IF (TX.LT.0) GO TO 1991
    IF (HVFLAG.EQ.0) XD=NOP*XPI
    IF (HVFLAG.EQ.1) XD=NOL*XLI
    IF (XD/TX.LT.0) GO TO 1991
    IF (HVFLAG.EQ.0) XPI=TX
    IF (HVFLAG.EQ.0) NOP=XD/XPI
    IF (HVFLAG.EQ.1) XLI=TX
    IF (HVFLAG.EQ.1) NOL=XD/XLI
GO TO 1991
C          Y IS OCTAL 131
C          CHANGE Y INCREMENT
131 CALL LOADTB(64*(YPI+YLI))
    ASSIGN 1310 TO CONNSN
GO TO 129
C          CODE WHEN >Y> CONNECTED TO TB
1310 CALL READTB(TBIN)
    TY=IRIGHT(TBIN)/64
    IF (TY.LT.0) GO TO 1991
    IF (HVFLAG.EQ.0) YD=NOL*YLI
    IF (HVFLAG.EQ.1) YD=NOP*YPI
    IF (YD/TY.LT.0) GO TO 1991
    IF (HVFLAG.EQ.0) YLI=TY
    IF (HVFLAG.EQ.0) YD=YD/YLI
    IF (HVFLAG.EQ.1) YPI=TY
    IF (HVFLAG.EQ.1) NOP=YD/YPI
GO TO 1991
C          Z IS OCTAL 132
132 CONTINUE
133 CONTINUE
134 CONTINUE
135 CONTINUE
136 CONTINUE
137 CONTINUE
140 GO TO 99
C
C          RETURN POINT FOR NU AND SIMPLE CHAR PROGRAMS
C
998 CHARIN=0
999 SCPLAC=2
    IFLAG=0
99 CHARIN=0
C
C          RETURN POINT AFTER A PARAMETER CONNECTED TO TB
C
199 RETURN
C          COMMON RETURN TO DISPLAY PROGRAM

```

```

C          AFTER CURRENT CONDITION OF TB EXAMINED
1991 RETURN
656 FORMAT(I3,5)
589 FORMAT(8I10)
667 FORMAT(10I,17,1X,5)
END
SUBROUTINE MOVERL (XPOS, YPOS)
IMPLICIT INTEGER (A-Y)
COMMON /MVER/XPINCR, YPINCR, XZERO, YZERO, XP, YP
COMMON /MCM/ QSMX(4,4), XCM(4,4), YCM(4,4)
XPO=XPOS+1
YPO=YPOS+1
IX=10
IY=20
XC1=XCM(1,YPO)-WY
XC2=XCM(2,YPO)-WY
YC1=YCM(XPO,1)-WX
YC2=YCM(XPO,2)-WX
YC3=YCM(XPO,3)-WX
IF(XPOS.EQ.0)XP=XZERO
IF(XPOS.EQ.1)XP=XZERO+XC1
IF(XPOS.EQ.2)XP=XZERO+XC1+XC2
IF(YPOS.EQ.0)YP=YZERO
IF(YPOS.EQ.1)YP=YZERO+YC1+2
IF(YPOS.EQ.2)YP=YZERO+YC1+2 +YC2+2
IF(YPOS.EQ.3)YP=YZERO+YC1+2 +YC2+2+YC3+2
CALL RLREAD (PXP,PYP)
IF ((XP.LT.PXP).OR.(YP.LT.PYP)) GO TO 20
CALL REMOVE (XP,YP)
CALL RLWAIT
RETURN
20 XP=XP-10
YP=YP-10
CALL REMOVE (XP,YP)
CALL RLWAIT
XP=XP+10
YP=YP+10
CALL REMOVE (XP,YP)
CALL RLWAIT
RETURN
450 CONTINUE
CALL BELL
CALL BELL
TYPE 460
460 FORMAT (> LIMIT>)
RETURN
END
SUBROUTINE BOUNDS
IMPLICIT INTEGER(A-Y)
COMMON /REFER/
1 LTD,LABEL,SEQNO,PROGNO,IDNO,DAY,MONTH,YEAR,
2 X,Y,XPI,YPI,NOP,XLI,YLI,NOL,DENL,BCI,HVFLAG,
3 CONNSN,CHARIN,BOP,ACTFN,REQFN,IFLAG
C
C      PLOT BOUNDARIES ON SLAVE SCOPE
C      OF AREA GIVEN BY QUANTITIES IN /REFER/
C
TX=X
TY=Y
TXPI=XPI

```

```

      TYPI=YPI
      TNOP=NOP
      CALL SKANL(X)
      X=X+NOL+XLI
      Y=Y+NOL+YLI
      CALL SKANL(X)
      X=TX
      Y=TY
      XPI=XLI
      YPI=YLI
      NOP=NOL
      CALL SKANL(X)
      X=X+TNOP+TXPI
      Y=Y+TNOP+TYPI
      CALL SKANL(X)
      X=TX
      Y=TY
      XPI=TXPI
      YPI=TYPI
      NOP=TNOP
      END
      SUBROUTINE DENSITY (DH,KEY)
      IMPLICIT INTEGER(A-Y)
      COMMON /REFER/
      1      LIQ,LABEL,SEMO,PROGNO,IONO,DAY,MONTH,YEAR,
      2      X,Y,XPI,YPI,NOP,XLI,YLI,NOL,DENL,CON,NVFLAG,
      3      CORRIN,CHARIN,POP,ACTEN,REGFM,IFLAG
      COMMON /BUFFS/
      1      NDUFF(256),HTBUFF(256),
      2      JDLACK(64),JREB(64),FBLACK(64),KREB(64),
      3      CURPY,ENDL,LINESP,IP,JI,KI
      COMMON/RTV/GPARD(4,11)
      COMMON /SCFLAG/ SCFLAG
      DIMENSION DH(6/63),TDEN(6)
      EQUIVALENCE (TDEN,KREB)
C
C      KEY=0 SAYS PLOT HISTERGRAM
C      KEY=1 SAYS DO NOT PLOT HISTERGRAM C
C
C      ZERO OUT THE HISTERGRAM AREA
C
      DO 640 I=0,63
640      DH(I)=0
      TWC=(NOP+NOL)/6
      KC=0
C
C      START OF THE DENSITY SCAN
C
      CALL SKAND(NDUFF,256,X)
641      IF (SCFLAG.EQ.0) GO TO 641
      DO 643 I=1,256
C      UPDEN IS A FAST MACRO PROGRAM TO UNPACK A DENSITY WORD
C
      CALL UPDEN(NDUFF(I),TDEN)
      DO 642 J=1,6
642      DH(TDEN(J))=DH(TDEN(J))+1
      KC=KC+1
      IF (KC.GE.TWC) GO TO 644
643      CONTINUE
      IF (SCFLAG.NE.1) GO TO 644

```



```

      CALL SPARK (NBUFF,256)
      GO TO 651
644 IF (NEY.NE.0) RETURN
C
C   KEY OF ZERO SAYS TO PLOT HISTOGRAM
C
      CALL QEMPTY
      CALL QSIZE (2)
      NY=850
      NYM=800
      NYX=0
      DO 645 I=0,63
6451 IF (DN(I).GT.MAX) MAX=DN(I)
645  NY=100
      DO 646 I=0,63
          IF (DN(I).EQ.0) GO TO 646
          EXCDE (5,7107,0V) DN(I)
7107 FOTDN (13)
          CALL QDPHMS(MY)
          CALL QDOPDN1
          CALL QDOPD10C(MY)
          CALL QDOPD10V
          CALL QDOP1(MY)
          CALL QVECTOR (FOTDN+(000)/MAX,MY)
646  NY=NY-16
70  CALL QSTART
637 CALL PCHP (CHAPIN)
      IF (CHMIN.EQ.0) GO TO 637
      CALL QSTOP
      IF (CHMIN.EQ.71) GO TO 650
      IF (CHMIN.EQ.87) GO TO 651
      IF (CHMIN.EQ.70) GO TO 49
      RETURN
49  CONTINUE
      CALL QDEC(QQ,MI)
      GO TO 441,MI
      CALL QFILM
50  CONTINUE
      CALL QMOVE
      CALL QDVE
      CALL QSTART
      GO TO 637
650 CONTINUE
      WRITE(3,648)
      DO 647 I=0,63
647  IF (DN(I).NE.0) WRITE (3,649) I,DN(I)
      GO TO 70
651 CONTINUE
      TYPE 955
956 FORMAT( ' THRESHOLD DENSITY>>')
      TYPE 957
957 FORMAT( ' DENH =-4)
      CALL FREQ(Q2,DENL)
      PDENH 1,91-DENL
      FQDN=0
      FQDN1=0
      FQDN2=0
      FQDN3=0
      GO 652 I=1,63

```

```

PSUM=PSUM+DWH1)
IF(1.LE.DWH1) GO TO 452
PDENT=PDEN+DWH1)
IF(1.LE.DWH1) GO TO 452
PDENC=PDEN+DWH1)
IF(1.LE.DWH1) GO TO 452
PDEND=PDEN+DWH1)
452 CONTINUE
F1=PSUM+DWH1+100*PSUM
F2=PD+DWH1+100*PD
F3=PD+DWH1+100*PD
F4=PD+DWH1+100*PD
IF(1.LE.F1) PDENT=PDENT+DWH1
500 CONTINUE(1)
510 CONTINUE(1)
520 CONTINUE(1)
530 CONTINUE(1)
540 CONTINUE(1)
550 CONTINUE(1)
560 CONTINUE(1)
570 CONTINUE(1)
580 CONTINUE(1)
590 CONTINUE(1)
600 CONTINUE(1)
610 CONTINUE(1)
620 CONTINUE(1)
630 CONTINUE(1)
640 CONTINUE(1)
650 CONTINUE(1)
660 CONTINUE(1)
670 CONTINUE(1)
680 CONTINUE(1)
690 CONTINUE(1)
700 CONTINUE(1)
710 CONTINUE(1)
720 CONTINUE(1)
730 CONTINUE(1)
740 CONTINUE(1)
750 CONTINUE(1)
760 CONTINUE(1)
770 CONTINUE(1)
780 CONTINUE(1)
790 CONTINUE(1)
800 CONTINUE(1)
810 CONTINUE(1)
820 CONTINUE(1)
830 CONTINUE(1)
840 CONTINUE(1)
850 CONTINUE(1)
860 CONTINUE(1)
870 CONTINUE(1)
880 CONTINUE(1)
890 CONTINUE(1)
900 CONTINUE(1)
910 CONTINUE(1)
920 CONTINUE(1)
930 CONTINUE(1)
940 CONTINUE(1)
950 CONTINUE(1)
960 CONTINUE(1)
970 CONTINUE(1)
980 CONTINUE(1)
990 CONTINUE(1)

```

```

CALL WRTCHQW,VDT)
TIME 190
190 FORMAT= PAGE NO X=4)
CALL WRTCHQW,Q220)
TIME 191
290 FORMAT= THREE PAGE NO X=4)
CALL WRTCHQW,Q220)
DO 1000 A=1,4
DO 1100 J=1,4
WTD=Q220+J-1
XWD=1000+J-1
DO 1200 J,K=1,4)WTD,WDD)
STOP CONTINUE
1000 CONTINUE
1100 CONTINUE
TIME 1900
2100 FORMAT= J=1)
CALL WRTCHQW,Q220)
TIME 1901
2700 FORMAT= J=1)
CALL WRTCHQW,Q220)
STOP CONTINUE
DO 1000 I=1,4
DO 1100 J=1,4
TIME J,I)
1000 CONTINUE
1100 CONTINUE
2000 CONTINUE
2100 CONTINUE
2200 CONTINUE
2300 CONTINUE
2400 CONTINUE
2500 CONTINUE
2600 CONTINUE
2700 CONTINUE
2800 CONTINUE
2900 CONTINUE
3000 CONTINUE
3100 CONTINUE
3200 CONTINUE
3300 CONTINUE
3400 CONTINUE
3500 CONTINUE
3600 CONTINUE
3700 CONTINUE
3800 CONTINUE
3900 CONTINUE
4000 CONTINUE
4100 CONTINUE
4200 CONTINUE
4300 CONTINUE
4400 CONTINUE
4500 CONTINUE
4600 CONTINUE
4700 CONTINUE
4800 CONTINUE
4900 CONTINUE
5000 CONTINUE
5100 CONTINUE
5200 CONTINUE
5300 CONTINUE
5400 CONTINUE
5500 CONTINUE
5600 CONTINUE
5700 CONTINUE
5800 CONTINUE
5900 CONTINUE
6000 CONTINUE
6100 CONTINUE
6200 CONTINUE
6300 CONTINUE
6400 CONTINUE
6500 CONTINUE
6600 CONTINUE
6700 CONTINUE
6800 CONTINUE
6900 CONTINUE
7000 CONTINUE
7100 CONTINUE
7200 CONTINUE
7300 CONTINUE
7400 CONTINUE
7500 CONTINUE
7600 CONTINUE
7700 CONTINUE
7800 CONTINUE
7900 CONTINUE
8000 CONTINUE
8100 CONTINUE
8200 CONTINUE
8300 CONTINUE
8400 CONTINUE
8500 CONTINUE
8600 CONTINUE
8700 CONTINUE
8800 CONTINUE
8900 CONTINUE
9000 CONTINUE
9100 CONTINUE
9200 CONTINUE
9300 CONTINUE
9400 CONTINUE
9500 CONTINUE
9600 CONTINUE
9700 CONTINUE
9800 CONTINUE
9900 CONTINUE
10000 CONTINUE

```

C

```

CALL GOITCH(30,130)
DOW=1

```

```

CALL SSNTCH(33,133)
IF(I33.EQ.2) CALL MT7OPEN
IF(I30.EQ.1) GO TO 5
6 TYPE 923
923 FORMAT(> POSITION MAGTAPE TYPE IDENT>)
TYPE 933
933 FORMAT(> MAPX=>4)
CALL FDEC(QQ,MAPX)
TYPE 934
934 FORMAT(> MAPY=>4)
CALL FDEC(QQ,MAPY)
IDENT=JCOMD(MAPX,MAPY)
TYPE 935,MAPX,MAPY
TYPE 924
924 FORMAT(> IS IDENT CORRECT Y OR N>)
7 CALL RCHAR(D)
IF(D.EQ.0) GO TO 7
IF(D.EQ.73) GO TO 6
8 CALL MT9R(FERSM,550)
IF(PERSM(1).NE.IDENT) GO TO 8
9 CALL MT9R(FERSM,550)
IF(PERSM(2).NE.3) GO TO 9
TYPE 925
925 FORMAT(> MAGTAPE POSITIONED>)
RETURN
5 CONTINUE
30 CONTINUE
CALL SSNTCH(28,128)
IF(I28.EQ.1) GO TO 40
CALL MT7OPEN
31 TYPE 931
931 FORMAT(> IDENT START=>)
CALL FDEC(QQ,IDES)
TYPE 932,IDES
32 CALL RCHAR(D)
IF(D.EQ.0) GO TO 32
IF(D.EQ.70) GO TO 31
33 TYPE 932
932 FORMAT(> IDENT END=>)
CALL FDEC(QQ,IDEN)
TYPE 933,IDEN
34 CALL RCHAR(D)
IF(D.EQ.0) GO TO 34
IF(D.EQ.78) GO TO 33
35 CALL MT7R(FERSM,550)
IF(PERSM(1).NE.IDES) GO TO 35
CALL MT9H(FERSM,550)
36 CALL MT7R(FERSM,550)
CALL MT9H(FERSM,550)
IF(PERSM(1).NE.IDEN) GO TO 36
37 CALL MT7R(FERSM,550)
CALL MT9H(FERSM,550)
IF(PERSM(2).NE.3) GO TO 37
CALL MT7CLOSE
RETURN
40 CONTINUE
ERG=0
1 CONTINUE
CALL SSNTCH(34,134)
CALL SSNTCH(35,135)

```



```

CALL PGMWAT
IF(23.EQ.1)RETURN
TYPE 929
929 FUDYAT(0110)
GO YOU WANT TO PROCESS NEXT PAGE Y OR N)
931 CALL PGMWAT
IF(10.01.01) GO TO 151
IF(10.00.01) RETURN
CALL M0100
CALL M0101
CALL M0102
RETURN
933 CONTINUE
CALL STWTRN(15,115)
IF(10.EQ.0) GO TO 801
TYPE 933
933 FUDYAT(1 > POSITION QUARTER SECTION > )
TYPE 933
937 CONTINUE
OPR=0
MOL=50
FDR=1
MOP=0
CALL SKANI
OPR=0
CALL STWTRN(1)
9771 CALL SKAN(COFF,256,X)
GO 7774 J=1,2000
9774 CONTINUE
9775 CALL STWTRN(1)
CALL PGMWAT
IF(10.EQ.0) AND(100.EQ.1) 100=0
IF(10.EQ.0) AND(100.EQ.0) 100=1
IF(10.EQ.0) 100=0
IF(10.EQ.0) GO TO 140
IF(10.EQ.0) 100=1
IF(10.EQ.0) 100=0
IF(100.EQ.0) GO TO 7772
IF(100.EQ.1) GO TO 7773
IF(SFLAG.EQ.0) GO TO 7771
CALL SKAN(COFF,256)
GO TO 7772
9775 CALL SKANI
OPR=1,1=X
OPR=1,2=Y
801 CONTINUE
MOP=1000000*(YOP)+3301/200
MOL=1000000*(YOP)+3301/200
OPR=1,0=MOL
OPR=1,5=MOP
RETURN
909 FUDYAT(0110)
END
ROUTINE ISOLATE(FLO)
INTEGER I(1)-Y
COMMON /COMMON/ YOP, YDT, YDT, FNO, LNO
COMMON /COMMON/ SO, HQ, SO, HQ, C, INCP, F, AVDEN
COMMON /COMMON/ CTAG(9000), FCK(9000), SECT
COMMON /COMMON/ I0E
COMMON /COMMON/ MIPR, MIPR, MIPR, MIPR
COMMON /REFER/

```

```

1      LID, LABEL, SEQNO, PROGNO, IDNO, DAY, MONTH, YEAR,
2      X, Y, XPI, YPI, NOP, XLI, YLI, NOL, DENL, BOH, HVFLAG,
3      CONNSH, CHARIN, BOP, ACTFN, REQFN, IFLAG
COMMON /BUFFS/
1      NBUFF(256), MTPUFF(256),
2      JBLACK(64), JPRED(64), KBLACK(64), KRED(64),
3      CURRY, ENDL, LINECP, IP, JI, KI
COMMON/IS/ 10, I1, I2, I3, I4
COMMON/JT/JTAG(64), KTAG(64), NBINA(300), LINK(300),
1      NBINYL(300), NBINXH(300), NBIRYL(300), NBIRXH(300), NQ
COMMON /SCFLAG/ SCFLAG
COMMON/NA/NA(6)
DATA MINA /9999/

C
C
CALL QSTOP
TOSP= 9300
JBLACK(1)=-1
KBLACK(1)=-1
BFC=300
DO 7 I=1, BFC
LINK(I)=0
7  NBINA(I)=0
APRILT=NOL+NOP
ADSVS=0
MEREJ=0
LINECP=YLI
CALL QEMPTY
YSTR=Y
SCF=NOP*XPI
CURRY=Y
ENDL=Y+NOL*YLI
NIE=1
CFCOT=1

C
C
C      START OF ACTUAL SCAN OF AREA OF INTEREST
C
NOL=1
20 CONTINUE
CALL SNTCH(29, I29)
IF (I29.EQ.1) CALL SKAND(NBUFF, 256, X)
IF (I29.EQ.2) CALL SKAND(NBUFF, 256, X)

C
C
C      LOOP OVER WHICH COMPLETELY NEW AREAS ARE INITIATED
C
505 CALL NEXTL(Q)
IF (Q.EQ.0) GO TO 505
IF (Q.EQ.999) GO TO 1111
OPEN
402 CONTINUE
42  CONTINUE
LS=JPRED(JI)-JBLACK(JI)
NBINA(NIE)=LS
JTAG(JI)=NIE
NBINXH(NIE)=JPRED(JI)
NBIRYL(NIE)=JBLACK(JI)
NBIRXH(NIE)=CURRY
NIE=NIE+1
NA(1)=(CURRY-YDO)/CINCR
JKL=JBLACK(JI)

```

```

JRE=JRED(JI)
MA(2)=(JPL-XED)/CINCR
MA(3)=(JRE-XDD)/CINCR
GTAG(SEGCT)=JTAG(JI)
CIG=1
CALL SSNTCH(2,12)
IF(12.EQ.2)
PRINT 309,GTAG(SEGCT),MA(1),MA(2),MA(3),SEGCT,ONE
989 FORTAG(018)
CALL TRSEFPAC(MA,PKI(SEGCT))
TOP1=SEGCT+1
PRO=0
IF(SEGCT.GT.TOP1) CALL NBINDISPLAY(PRO)
JI=JI+1
IF (JBLACK(JI).NE.-1) GO TO 400
C
C     NORMAL START OF GRABED LOOP
C
403 CONTINUE
CALL NEXTLEQ
1313 IF (Q.NE.1) GO TO 424
C
C     TESTING OF LEFT SIDES
C
410 IF (KBLACK(KI).EQ.-1) GO TO 430
IF (JBLACK(JI).EQ.-1) GO TO 420
IF (JRED(JI).LT.KBLACK(KI)) GO TO 430
IF (JBLACK(JI).GT.(RED(KI)) GO TO 420
C     BLACK
NRKTAG(KI)
JTAG(JI)=NR
IF (JBLACK(JI).LT.NBINX(NN)) NBINX(NN)=JBLACK(JI)
C
C     TESTING OF RIGHT SIDES
C
413 IF (KBLACK(KI+1).EQ.-1) GO TO 414
IF (JRED(JI).GT.KBLACK(KI+1)) GO TO 450
414 IF (JBLACK(JI+1).EQ.-1) GO TO 415
IF (JBLACK(JI+1).LT.KRED(KI)) GO TO 460
C     RED
415 CONTINUE
NRKTAG(KI)
JTAG(JI)=NR
LS=JRED(JI)-JBLACK(JI)
NBINA(NN)=NBINA(NN)+LS
IF (JRED(JI).GT.NBINX(NN)) NBINX(NN)=JRED(JI)
MA(1)=(CURTY-YDD)/CINCR
JPL=JBLACK(JI)
JRE=JRED(JI)
MA(2)=(JPL-XED)/CINCR
MA(3)=(JRE-XDD)/CINCR
GTAG(SEGCT)=JTAG(JI)
CIG=2
IF(12.EQ.2)
PRINT 309,GTAG(SEGCT),MA(1),MA(2),MA(3),SEGCT,ONE
CALL TRSEFPAC(MA,PKI(SEGCT))
SEGCT=SEGCT+1
PRO=0
IF(SEGCT.GT.TOP1) CALL NBINDISPLAY(PRO)
JI=JI+1

```



```

      KI=KI+1
C
417 IF ((JBLACK(JI).EQ.-1).AND.(KBLACK(KI).EQ.-1)) GO TO 403
      GO TO 410
C
      CLOSE
420 CONTINUE
      ASSIGN 417 TO CRET
      GO TO 3003
C
C
430 OPEN
      CONTINUE
      LS=JRED(JI)-JBLACK(JI)
      NBINA(NNE)=LS
      JTAG(JI)=NNE
      NBINXH(NNE)=JRED(JI)
      NBINXL(NNE)=JBLACK(JI)
      NBINYL(NNE)=CURRY
      MA(1)=(CURV-YDD)/CINCR
      JPL=JBLACK(JI)
      JRE=JRED(JI)
      MA(2)=(JKL-XDD)/CINCR
      MA(3)=(JRE-XDD)/CINCR
      GTAG(SEGCT)=JTAG(JI)
      CUE=3
      IF(I2.EQ.2)
1PRINT 909,GTAG(SEGCT),MA(1),MA(2),MA(3),SEGCT,ONE
      CALLTDRBPACINA,PCR(SEGCT)
      SEGCT=SEGCT+1
      PDS=0
      IF(SEGCT.GT.TGSP) CALL NBINDISPLAY(PRO)
      JI=JI+1
      NNE=NNE+1
      IF(NNE.GT.DFC) GO TO 4003
C
      GO TO 417
C
450 FUSION
      CONTINUE
      MTT=JTAG(JI)
      HT=KTAG(KI+1)
      IF (HT.EQ.MTT) GO TO 80
      IF (HT.LT.MTT) GO TO 535
      GO TO 536
535 HT=MTT
      MTT=KTAG(KI+1)
536 NBINA(MTT)=NBINA(MTT)+NBINA(HT)
      NBINXL(MTT)=MIN0(NBINXL(MTT),NBINXL(HT))
      NBINXH(MTT)=MAX0(NBINXH(MTT),NBINXH(HT))
      JTAG(JI)=MTT
      NBINA(MT)=0
      NED=JI-1
      DO 603 I=1,NED
603 IF (JTAG(I).EQ.HT) JTAG(I)=MTT
      DO 604 I=1,SEGCT
      IF(GTAG(I).EQ.HT)GTAG(I)=MTT
604 CONTINUE
      DO 605 I=KI,64
605 IF (KTAG(I).EQ.HT) KTAG(I)=MTT
80 KI=KI+1
C
      GO TO 461

```

```

C      SPLIT
460  CONTINUE
      NI=JTAG(JI)
      NBINA(NI)=NBINA(NI)+JRED(JI)-JBLACK(JI)
      JTAG(JI+1)=JTAG(JI)
      MA(1)=(CURRY-YBO)/CINCR
      JKL=JBLACK(JI)
      JRE=JRED(JI)
      MA(2)=(JKL-XBO)/CINCR
      MA(3)=(JRE-XBO)/CINCR
      GTAG(SEGCT)=JTAG(JI)
      ONE=4
      IF(I2.EQ.0)
        PRINT 509,GTAG(SEGCT),MA(1),MA(2),MA(3),SEGCT,ONE
        CALLTHREEDAC(MA,POK(SEGCT))
      SEGCT=SEGCT+1
      I2=0
      IF(SEGCT.GT.TOSP) CALL NBINDISPLAY(FRO)
      JI=JI+1
C
461  IF ((JBLACK(JI).EQ.-1).OR.(KBLACK(KI).EQ.-1)) GO TO 417
      GO TO 413
C
424  CLOSE
      CONTINUE
      ASSIGN 425 TO CRET
      GO TO 3003
425  CONTINUE
C
      IF (KBLACK(KI).NE.-1) GO TO 424
      IF (Q.EQ.0) GO TO 505
      GO TO 1111
C
C      CLOSE OPERATIONS
C
3003 CONTINUE
      CN=KTAG(KI)
      IF (JI.EQ.1) GO TO 3017
      DO 3007 I=1,JI
3007  IF (CJLGO(I).EQ.BN) GO TO 3057
3017  DO 3027 I=KI+1,KI
      IF (JBLACK(I).EQ.-1) GO TO 3037
3027  IF (KTAG(I).EQ.BN) GO TO 3057
3037  CONTINUE
      IF (NBINA(BN).LT.MINA) AREREJ=AREREJ+NBINA(BN)
      IF (NBINA(BN).GE.MINA) ARESVE=ARESVE+NBINA(BN)
      IF (NBINA(BN).LT.MINA) NBINA(BN)=0
3057  NBINYH(BN)=CURRY-LINESP
      KI=KI+1
      GO TO CRET
C
C      GARBAGE COLLECTION (ON NBINA BEING 0)
C
4003 CALL SSNCH (3,13)
      TYPE 923
      923 FORMAT(> GARBAGE COLLECTION>)
      NQ=0
      DO 200 I=1,DFC
      IF(NBINA(I).EQ.0) GO TO 200
      NQ=NQ+1
      NBINA(NQ)=NBINA(I)

```

```

NBINXL(NQ)=NBINXL(I)
NBINXH(NQ)=NBINXH(I)
NBINYL(NQ)=NBINYL(I)
NBINYH(NQ)=NBINYH(I)
LINK(I)=NQ
200 CONTINUE
NNE=NQ+1
IF(NQ.EQ.BFC)
1PAUSE > ALL BINS FILLED, NO GARBAGE COLLECTION POSSIBLE>
DO 201 I=NNE,BFC
201 NBIN(I)=0
DO 202 I=1,64
IF(JTAG(I).NE.0)JTAG(I)=LINK(JTAG(I))
IF(KTAG(I).NE.0)KTAG(I)=LINK(KTAG(I))
202 CONTINUE
IF(I3.EQ.1) GO TO 4170
WRITE (3,4056) NNE
DO 4071 I=1,BFC
4071 PRINT 4056,I,LINK(I),NBIN(I)
4056 FORMAT (1H1,I7)
4170 CONTINUE
DO 4172 I=1,SECT-1
STAG(I)=LINK(STAG(I))
4172 CONTINUE
GO TO 417
C
C COMPLETION OF ALL VECTOR GENERATION
C
1111 CONTINUE
Y=YSTART
4056 FORMAT(10I2)
END
SUBROUTINE NEXTL (FLAG)
IMPLICIT INTEGER(A-Y)
COMMON /REFER/
1 LID,LABEL,SEQNO,PROGNO,IDNO,DAY,MONTH,YEAR,
2 X,Y,XPI,YPI,NOP,XLI,YLI,NOL,ONL,SON,HVFLAG,
3 COMMON,CHARIN,CCP,ACTFN,REQFN,IFLAG
COMMON /DUFFS/
1 NBUFF(256),HTBUFF(256),
2 JBLACK(64),JRED(64),KBLACK(64),KRED(64),
3 CURRY,ENDL,LINESP,IP,JI,KI
COMMON/JT/JTAG(64),KTAG(64),NBIN(300), LINK(300),
1 NBINXL(300),NBINXH(300),NBINYL(300),NBINYH(300),NQ
COMMON /DIAMEN/ SQ,HSQ,DD,HDD,CINCR,F,AVDEN
COMMON /SCFLAG/ SCFLAG
DIMENSION DVAL(500),DAVE(500)
23 IF (SCFLAG.EQ.0) GO TO 23
IF(SCFLAG.EQ.1) PAUSE > BUFFER OVERFLOW >
CALL SHTCH(29,I29)
IF(I29.EQ.2) GO TO 50
CALL SKANCC(Q)
CALL MOVE(NBUFF,HTBUFF,Q)
53 CONTINUE
Y=Y+YLI
IF(I29.EQ.1) CALL SKANC(NBUFF,256,X)
IF(I29.EQ.2) CALL SKAND(NBUFF,256,X)
C
C PUTS CO-ORDS OF NEXT LINE (IF ANY)IN JBLACK,JRED
C FLAG=0,NO COORDS,FLAG=1,SOME COORDS,FLAG=999,END OF DATA

```

```

C           -1 TERMINATES DATA
C   MOVE CURRENT LINE TO PREVIOUS LINE
C
C   I=1
117 KBLACK(I)=JBLACK(I)
    JRED(I)=JRED(I)
    KTAS(I)=JTAS(I)
118 IF (JBLACK(I).EQ.-1) GO TO 119
    I=I+1
    GO TO 117
119 CURRY=CURRY+LINESP
    JI=1
    KI=1
    I=1
C
C   LOOP FOR GETTING COORDS FOR NEXT LINE
C
299 IF (CURRY.EQ.ENDL) GO TO 550
    IF(Q.EQ.0) GO TO 28
    GO 27 J=1,Q,2
    JBLACK(I)=ILEFT(MTBUFF(J))
    JRED(I)=ILEFT(MTBUFF(J+1))
    I=I+1
    IF(I.EQ.64) PAUSE > I IS EQUAL TO 64 >
27 CONTINUE
28 CONTINUE
    JBLACK(I)=-1
    JRED(I)=-1
    FLAG=1
    IF(Q.EQ.0) FLAG=0
    RETURN
50 CONTINUE
    CALL SUPDEN(NBUFF,DVAL,NOP)
    CALL QEMPTY
    MM=5
    SUM=0
    DO 54 I=1,50
54 SUM=SUM+DVAL(I)
    DO 55 I=1,25
55 DAVE(I)=SUM/50+MM
    DO 56 I=26,NOP-25
    SUM =SUM +DVAL(I+25)-DVAL(I-25)
56 DAVE(I)=SUM/50+MM
    DO 57 I=NOP-25,NOP
57 DAVE(I)=SUM/50+MM
    CALL QNOP(1,DAVE(1)*6)
    DO 60 I=2,NOP
60 CALL QVECTOR(I*28/10,DAVE(I)*6)
    DO 51 I=1,NOP
    CALL QNOPI(I*28/10,1)
    CALL QVECTOR(I*28/10,DVAL(I)*6)
51 CONTINUE
    Q=0
    K=0
    MARK=0
    IF(DVAL(1).LE.DAVE(1))K=K+1
    IF(DVAL(1).LE.DAVE(1))MTBUFF(K)=ICOMB(X*XPI,CURRY)
    IF(DVAL(1).LE.DAVE(1))MARK=1
    DO 61 I=2,NOP
    IF(MARK.EQ.1) GO TO 661

```

```

      IF(DVAL(I).GT.DAVE(I)) GO TO 61
      K=K+1
      NTBUFF(K)=ICOMB(X+I*XPI,CURRY)
      MARK=1
      GO TO 61
661 IF(DVAL(I).LE.DAVE(I)) GO TO 61
      K=K+1
      NTBUFF(K)=ICOMB(X+I*XPI,CURRY)
      MARK=0
61  CONTINUE
      IF(MARK.EQ.0) GO TO 662
      K=K+1
      NTBUFF(K)=ICOMB(X+NCP*XPI,CURRY)
662 CONTINUE
      Q=K
      DO 70 I=1,0-1,2
      XD=ILEFT(NTBUFF(I))
      XN=ILEFT(NTBUFF(I+1))
      XB=(XB-X)/XPI
      XR=(XR-X)/XPI
      CALL QNDP(XB*28/10,(DAVE(XB)+ 5)*6)
70  CALL QVECTOR(XR*28/10,(DAVE(XB)+5)*6)
      CALL QSNTCH(9,I9)
      IF(I9.EQ.2) GO TO 62
      CALL QSTART
52  CALL QCHAR(D)
      IF(D.EQ.0) GO TO 52
      CALL QSTOP
62  CONTINUE
989 FORMAT(8I10)
      GO TO 53
550 FLAG=999
      END
      SUBROUTINE NBINDISPLAY(PRO)
      IMPLICIT INTEGER(A-Y)
      COMMON/SGT/SEG,PGC
      COMMON/EDC/EDC(4)
      COMMON/STST/STST(99),SAST(99),SASH(99)
      COMMON/RP/REP
      COMMON/PERSH/PERSH(550)
      COMMON/DIANEN/ SQ,HSQ,DD,HDD,CINCR,F,AVDEN
      COMMON/HO/GTAG(9000),PCK(9000),SEGCT
      COMMON/ND/NBLACK(2500),NRED(2500),NCURRY(2500),NTAG(2500)
      1,NTAG(2500)
      COMMON/ENO/XBO,YBO,XDT,YDT,FNO,LNO
      COMMON/XXX/XHH,XLL,YHH,YLL
      COMMON/CHNE/ NNE
      COMMON/REFER/
      1  LID,LABEL,SEQNO,PROGNO,IDNO,DAY,MONTH,YEAR,
      2  X,Y,XPI,YPI,NOP,XLI,YLI,NOL,DENL,EOW,HVFLAG,
      3  CONSIG,CHARIN,BOP,ACTFN,REQFN,IFLAG
      COMMON/BUFFS/
      1  HBUFF(256),MTBUFF(256),
      2  JBLACK(64),JPED(64),KBLACK(64),KRED(64),
      3  CURRY,ENOL,LINESP,IP,JI,KI
      COMMON/IS/ I0,I1,I2,I3,I4
      COMMON/BPARS/MINLEN,MINLEN,MINA
      COMMON/QTV/ QPARS(4,11)
      COMMON/SCFLAG/ SCFLAG
      DIMENSION APARS(11)

```

```

EQUIVALENCE (AFARS(1),X)
COMMON/JT/JTAG(64),KTAG(64),NBINA(300), LIMG(300),
1 NBINXL(300),NBINXH(300),NBINYL(300),NBINYH(300),NQ
COMMON/MA/MA(4)
COMMON/TART/TART
COMMON /AP/AR
DATA AR/0/

```

C

```

CALL BSNTCH(I2,I32)
SRD=900
FK=0
FGC=0
DO 1 I=1,SEGCT-1
1 IF(FGC.LT.GTAG(I))FGC=GTAG(I)
DO 4 I=1,4
4 DDC(I)=0
DO 5 I=1,99
SIST(I)=0
SAST(I)=0
5 SASN(I)=0
TART=0
SEGCT=SEGCT-1
SKT=PNQ
IF(LND.GT.PHQ)SKT=LND
LNL=1
LNH=LND/2
FTL=1
PTH=PNQ/2
CALL QLTB(50,50)
DO 69 J=1,4
CALL QEMPHY
DO 70 I=1,SEGCT
CALL UPTHREE(FCK(I),MA)
IF((MA(1).LT.LNL).OR.(MA(1).GT.LNH)) GO TO 70
IF((MA(2).GT.PTH).OR.(MA(3).LT.PTL)) GO TO 70
IF((MA(2).LE.PTH).AND.(MA(3).GT.PTH))MA(3)=PTH
IF((MA(2).LT.PTL).AND.(MA(3).GE.PTL))MA(2)=FTL
CALL QNOP(MA(2)*SKB/SKT+50,MA(1)*SKB/SKT+50)
CALL QVECTOR(MA(3)*SKB/SKT+50,MA(1)*SKB/SKT+50)
CALL QREADI(IPAR)
IF(IPAR.GT.4090) GO TO 71
70 CONTINUE
71 CONTINUE
CALL QREADI(IPAR)
66 CONTINUE
CALL QSTOP
CALL QRTB(CX,CY)
CALL QSETI(IPAR)
CALL QNOP(CX-32,CY-32)
CALL QVECTOR(CX+32,CY+32)
CALL QNOP(CX-32,CY+32)
CALL QVECTOR(CX+32,CY-32)
CALL QONCE
CALL QWAIT
CALL FCHAR(D)
IF(D.EQ.0) GO TO 66
IF(D.NE.70) GO TO 64
CALL RDEC(QQ,QM)
DO 644 Q=1,QM
CALL QFILM

```

```

644 CONTINUE
CALL QTDVE
CALL QTDVE
GO TO 65
64 CONTINUE
IF(J.EQ.4) GO TO 65
IF(J.EQ.3) GO TO 68
IF(J.EQ.2) GO TO 67
LNR=LND/2
LDR=LND
CALL QUTEX(50,800)
GO TO 65
67 CONTINUE
PIL=PDM-2
PDM=PDM
CALL QUTEX(800,800)
GO TO 65
60 CONTINUE
LNL=1
LNR=LNR/2
CALL QUTEX(800,50)
65 CONTINUE
DDB=CK-50+SKT/SKB
DTC=CK-50+SKT/SIB
DUE=J-10000/DML/DAY
69 CONTINUE
I=GTAG(1)
CALL UPTHREE(FCK(1),MA)
UP=MA(1)
REP=0
9 CONTINUE
REP=0
CALL QSTOP
CALL QDNTY
CALL QUTEX(50,50)
CALL QVECTOR(10,LND*SKB/SKT+50)
CALL QVECTOR(10,SIB/SKT+50,LND*SKB/SKT+50)
CALL QVECTOR(10,SIB/SKT+50,50)
CALL QVECTOR(50,50)
HARE=0
L=0
JWR=0
XLL=1000
YWR=0
YLL=1000
DO TO I=1,SECT
IF(GTAG(I).NE.K) GO TO 10
GTAG(I)=GTAG(I)
L=L+1
CALL UPTHREE(FCK(I),MA)
IF(MA(1).EQ.0)MA(1)=1
IF(MA(2).EQ.0)MA(2)=1
ITLACK(L)=MA(2)
ITDYL(L)=MA(1)
ITDZYL(L)=MA(1)
IF(MA(3).GT.0)JWR=MA(3)
IF(MA(2).LT.0)YLL=MA(2)
YLL=HCLTRY(I)
HARE=HARE+MA(3)+MA(2)
CALL QDPT(MA(2)*SKB/SKT+50,MA(1)*SKB/SKT+50)

```



```

SEG=L
PK=PK+1
IF(PK.GT.99) TYPE 940
940 FORMAT(> POLYGON OVRFLOW>)
CALL EXAMINE(HK,PK)
IF(REP.EQ.2)GO TO 45
22 CONTINUE
IF(D.EQ.33)TART=TART+MARE
IF(REP.EQ.0) GO TO 95
TYPE 909,REP
K=COD(REP)
REP=REP-1
GO TO 9
95 CONTINUE
DO 12 I=1,SEGCT
IF(IGTAG(I).LE.0) GO TO 12
%GTAG(I)
GO TO 9
12 CONTINUE
989 FORMAT(8I10)
C
C   END OF A PICTURE
C
IF(D.EQ.47)RETURN
TOT=0
CALL CLOSOUT(TOT,PK)
CALL QSTOP
RETURN
200 CONTINUE
45 CONTINUE
CALL QEMPTY
CALL UPTRREE(PCK(1),MA)
M=CM(1)
DO 100 I=1,SEGCT
CALL UPTRREE(PCK(I),MA)
IF(MA(1).NE.MAC) GO TO 99
CALL QCOPY(MA(2)+SKB/SKT+50,MA(1)+SKB/SKT+50)
CALL QVECTOR(MA(3)+SKB/SKT+50,MA(1)+SKB/SKT+50)
GO TO 100
99 CONTINUE
IF(MA(1).GT.MAC)MAC=MAC+3
100 CONTINUE
CALL QSTART
101 CONTINUE
CALL RCHAR(E)
IF(E.EQ.0) GO TO 101
CALL QSTOP
IF(E.NE.70) GO TO 47
CALL QDEC(QQ,QM)
DO 646 Q=1,QM
CALL QFILM
646 CONTINUE
CALL QSTART
CALL SNTCH(20,I20)
IF(I20.EQ.2)GO TO 101
CALL QMOVE
CALL QMOVE
CALL QSTART
GO TO 101
47 CONTINUE

```



```

      0=0
      CALL ROTARY
      GO TO 50
C GENERATES CONTOUR OF SELECTED AREA
C
      70 CONTINUE
      CALL SECTION(0,130)
      CALL SECTION(2,10)
      DO 67 M=1,55
      JBLACK(K)=0
      JRED(K)=0
      JGRAY(K)=0
      67 JBLACK(K)=0
      DO 71 N=1,SEG
      JRED(K)=0
      PRINT 900,(JBLACK(K),JRED(K),JGRAY(K),K
      JGRAY(K)=0
      71 JBLACK(K)=0
      K=1
      68 JGRAY(K)=-1
      IF(JGRAY(K).NE.JGRAY(K+1)) GO TO 69
      K=K+1
      GO TO 63
      69 CONTINUE
      M=1
      L=2
      GO TO 72
      403 CONTINUE
      M=L
      L=L+1
      72 IF(JGRAY(M).NE.JGRAY(L)) GO TO 73
      L=L+1
      GO TO 72
      73 CONTINUE
C
C      NORMAL START OF GRAND LOOP
C
      KI=1
      DO 74 H=K,SEG
      JBLACK(KI)=JBLACK(H)
      JRED(KI)=JRED(H)
      JI=KI+1
      IF(JGRAY(H).NE.JGRAY(H+1)) GO TO 75
      74 CONTINUE
      75 JBLACK(KI)=-1
      KI=1
      IF(L.GT.SEG ) GO TO 424
      JI=1
      DO 76 H=L,SEG
      JBLACK(JI)=JBLACK(H)
      JRED(JI)=JRED(H)
      JI=JI+1
      IF(JGRAY(H).NE.JGRAY(H+1)) GO TO 77
      76 CONTINUE
      77 JBLACK(JI)=-1
      JI=1
C      TESTING OF LEFT SIDES
C
      410 IF (JBLACK(KI).EQ.-1) GO TO 430
      IF (JBLACK(JI).EQ.-1) GO TO 420

```

```

      IF (JRED(JI).LT.KBLACK(KI)) GO TO 430
      IF (JBLACK(JI).GT.KRED(KI)) GO TO 420
C     BLACK
      NTAG(K+KI-1)=JI+L-1
C
C     TESTING OF RIGHT SIDES
C
413  IF (KBLACK(KI+1).EQ.-1) GO TO 414
      IF (JRED(JI).GT.KBLACK(KI+1)) GO TO 450
414  IF (JBLACK(JI+1).EQ.-1) GO TO 415
      IF (JBLACK(JI+1).LT.KRED(KI)) GO TO 460
C     RED
415  CONTINUE
      NTAG(L+JI-1)=KI+K-1
      JI=JI+1
      KI=KI+1
C
417  IF ((JBLACK(JI).EQ.-1).AND.(KBLACK(KI).EQ.-1)) GO TO 403
      GO TO 410
C     CLOSE
420  CONTINUE
      NTAG(K+KI-1)=-1
      KI=KI+1
      GO TO 417
C
C     OPEN
430  CONTINUE
      NTAG(L+JI-1)=-1
      JI=JI+1
C
      GO TO 417
C     FUSION
450  CONTINUE
      NTAG(K+KI)=-2
      KI=KI+1
      GO TO 461
C
C     SPLIT
460  CONTINUE
      NTAG(L+JI-1)=-2
      JI=JI+1
C
461  IF ((JBLACK(JI).EQ.-1).OR.(KBLACK(KI).EQ.-1)) GO TO 417
      GO TO 413
C     CLOSE
C     CLOSE OPERATIONS
C
424  CONTINUE
      NTAG(K+KI-1)=-1
      KI=KI+1
      IF (KBLACK(KI).NE.-1) GO TO 424
      CALL SSNTCH(10,I10)
      IF(I10.EQ.1) GO TO 2424
      DO 2423 I=1,SEG
2423 PRINT 939,NBLACK(I),NRED(I),NCURRY(I),NTAG(I),MTAG(I),I
2424 CONTINUE
C
C     PACK CONTURE IN MINIMUM FORMAT
C
      CALL QEMPTY

```

```

K=1
ST=1
MELT=0
APAT=0
CHARCNT=0
G=5
PERSVIG)=1
G=G+1
PERSVIG)=NCURRY(K)
G=G+1
PERSVIG)=NBLACK(K)
90 CONTINUE
IF(MTAG(K).LT.0) GO TO 92
I=INTAG(K)
91 CONTINUE
IF(MELT.EQ.1) GO TO 104
G=G+1
IF(G.GT.2200)TYPE 9490
9490 FORMAT (5X, > FER OVERFLOW HIT KEY J >)
IF(G.GT.2200) GO TO 4999
PERSVIG)=NBLACK(K)
910 CONTINUE
GO TO 90
104 CONTINUE
LLL=NBLACK(K)
E=E+1
IF(E.GT.99)TYPE 931
IF((E.GT.99).AND.(I30.EQ.1))CHARCNT=10
IF((E.GT.99).AND.(I30.EQ.1))GO TO 2000
IF(LLL.LT.0) GO TO 1040
APAT=APAT+NBLACK(K)-NRED(K-1)
CHASTE(E)=NBLACK(K)
IF(NBLACK(K).GT.HIYX)HIYX=NBLACK(K)
IF(NRED(K-1).LT.LOWX)LOWX=NRED(K-1)
IF(NCURRY(K).GT.YHIY)YHIY=NCURRY(K)
NBLACK(K)=NBLACK(K)
NRED(K-1)=NRED(K-1)
GO TO 910
1040 CONTINUE
CHASTE(E)=NBLACK(K)
GO TO 910
92 CONTINUE
IF(MTAG(K).EQ.-2) GO TO 94
93 CONTINUE
IF(MELT.EQ.1) GO TO 105
G=G+1
IF(G.GT.2200)TYPE 9490
IF(G.GT.2200) GO TO 4999
PERSVIG)=-NRED(K)
930 CONTINUE
IF(K.EQ.ST) GO TO 96
IF(MTAG(K).LT.0) GO TO 95
K=MTAG(K)
GO TO 93
105 CONTINUE
E=E+1
IF(E.GT.99)TYPE 931
931 FORMAT (> CHAR CNTURE OVERFLOW CHECK FOR ISLAND >)
IF((E.GT.99).AND.(I30.EQ.1))CHARCNT=10
IF((E.GT.99).AND.(I30.EQ.1))GO TO 2000

```

```

LLL=NBLACK(K+1)
IF(LLL.LT.0) GO TO 1050
ARAT=ARAT+NBLACK(K+1)-NRED(K)
IF(NRED(K).LT.LOWK)LOWK=NRED(K)
IF(NBLACK(K+1).GT.HIYX)HIYX=NBLACK(K+1)
IF(NCURRY(K).GT.YNIY)YNIY=NCURRY(K)
CHASTE(E)=-NRED(K)
NBLACK(K+1)=-NBLACK(K+1)
NRED(K)=-NRED(K)
GO TO 930
1050 CONTINUE
CHASTE(E)=NRED(K)
GO TO 930
94 CONTINUE
F=F+1
GO TO 93
95 CONTINUE
IF(MTAG(K).EQ.-1) GO TO 91
MTAG(K)=0
K=K+1
GO TO 91
96 CONTINUE
IF(MELT.EQ.1) GO TO 108
G=G+1
PERSV(G)=NBLACK(K)
GO TO 107
103 CONTINUE
E=E+1
CHASTE(E)=NBLACK(K+1)
107 CONTINUE
IF(MELT.EQ.1) GO TO 89
IF(I2.EQ.1) GO TO 89
89 CONTINUE
PERSV(1)=G
PERSV(2)=MLL
PERSV(3)=YLL
PERSV(4)=SLF
IF(MELT.EQ.1) GO TO 110
CALL SSATCH(0,I0)
IF(DSTATE.EQ.65) GO TO 1020
C
C DISPLAY CONTURE
C
1020 CONTINUE
C
C LOCATE INTERNAL CHARACTERS AND OTHER OBJECTS
C
IF(MELT.EQ.0)F=1
MELT=1
103 CONTINUE
1033 CONTINUE
IF(MTAG(F).EQ.-2) GO TO 106
F=F+1
IF(F.GT.SEG ) GO TO 1099
GO TO 1033
106 CONTINUE
ST=F
F=F+1
K=F
ARAT=0

```

```

YLOY=NCURRY(K)
YLOX=NBLACK(K)
LOWX=NBLACK(K)
HIYX=NBLACK(K)
YHIY=NCURRY(K)
CHASTE(1)=1
CHASTE(2)=NCURRY(K)
CHASTE(3)=NBLACK(K)
E=3
GO TO 91
110 CONTINUE
CALL SSHTCH(4,I4)
ARAS=ARAS+ARAT
YYYY=YHIY-YLOY
XXXX=HIYX-LOWX
XARA=YYYY+XXXX
IF(I4.EQ.2)TYPE 989,XARA
IF(I4.EQ.2)
1TYPE 909,YYYY,XXXX
GO TO 103
IF(XARA.LT.10) GO TO 103
YLOX=(HIYX+LOWX)/2
YLOY=(YHIY+YLOY)/2
CHARCNT=CHARCNT+1
IF(CHARCNT.GT.25) PAUSE > CHARCNT OVERFLOW >
CHASTX(CHARCNT)=YLOX
CHASTY(CHARCNT)=YLOY
IF(I4.EQ.2)
1TYPE 989,YLOX,YLOY,CHARCNT,ARAT,ARAS
C
C PLOT CHARACTER COUTURE
C
GO TO 103
1099 CONTINUE
IF(CHARCNT.LT.2) GO TO 2000
IF(I4.EQ.1) GO TO 2101
TYPE 879
DO 2100 HK=1,CHARCNT
2100 TYPE 909,CHASTX(HK),CHASTY(HK)
2101 CONTINUE
INVT=0
1100 DO 1101 HK=1,CHARCNT-1
IF(CHASTX(HK+1).GE.CHASTX(HK)) GO TO 1101
INVT=1
CHASS=CHASTX(HK)
CHAST=CHASTY(HK)
CHASTX(HK)=CHASTX(HK+1)
CHASTY(HK)=CHASTY(HK+1)
CHASTX(HK+1)=CHASS
CHASTY(HK+1)=CHAST
1101 CONTINUE
IF(INVT.EQ.0) GO TO 1102
INVT=0
GO TO 1100
1102 CONTINUE
IF(I4.EQ.1) GO TO 2103
TYPE 879
DO 2102 HK=1,CHARCNT
2102 TYPE 989,CHASTX(HK),CHASTY(HK)
2103 CONTINUE

```



```

KH=26
DO 1103 HK=CHARCNT,2,-1
KH=KH-1
CHASTX(KH)=CHASTX(HK)
1103 CHASTY(KH)=CHASTY(HK)
KH=27-CHARCNT
1108 CONTINUE
BK=1
DO 1106 HK=KH,25
IF(CHASTX(HK).EQ.0) GO TO 1106
DIFCX=CHASTX(HK)-CHASTX(BK)
DIFAX=IABS(DIFCX)
DIFCY=CHASTY(HK)-CHASTY(BK)
DIFAY=IABS(DIFCY)
IF((DIFAX.GT.5000).OR.(DIFAY.GT.5000)) GO TO 2106
IF(BK.GE.2)DCHNG=FLOAT(DIFCX)*ZDIF
IF(BK.GE.2)DCHNG=DIFCY-DCHNG
IF(BK.GE.2)DCHNG=IABS(DCHNG)
IF((BK.GE.2).AND.(DCHNG.GT.500)) GO TO 2106
BK=BK+1
IF(BK.EQ.2)ZDIF=FLOAT(DIFCY)/FLOAT(DIFCX)
CHASTX(BK)=CHASTX(HK)
CHASTY(BK)=CHASTY(HK)
CHASTX(HK)=0
CHASTY(HK)=0
2106 CONTINUE
1106 CONTINUE
IF(BK.EQ.1) GO TO 1109
CHARCNT=BK
IF(I4.EQ.1) GO TO 1104
TYPE 379
879 FORMAT(I10)
DO 1105 HK=1,CHARCNT
1105 TYPE 989,CHASTX(HK),CHASTY(HK)
1104 CONTINUE
CALL SCNTCH(24,I24)
IF(I24.EQ.2)CHARCNT=10
IF(I24.EQ.2) GO TO 2000
IF(I24.EQ.1)CHARCNT=10
IF(I24.EQ.1) GO TO 2000
1109 CONTINUE
DO 1107 HK=KH,25
IF(CHASTX(HK).EQ.0) GO TO 1107
CHASTX(1)=CHASTX(HK)
CHASTY(1)=CHASTY(HK)
CHASTX(HK)=0
CHASTY(HK)=0
GO TO 1108
1107 CONTINUE
2000 CONTINUE
TK=PK
REP=0
CALL SOLTYDST(CHARCNT,PK)
PK=TK
2001 CONTINUE
CALL QENPTY
IF(REP.EQ.1)PK=PK+1
IF(REP.EQ.1)GO TO 3
RETURN
989 FORMAT(8I10)

```

```

END
SUBROUTINE SOLTYDSP(HK,PK)
IMPLICIT INTEGER (A-Y)
COMMON/QS/QS
COMMON/PP/REP
COMMON/XXY/XXH,XLL,YHH,YLL
COMMON/LFC/PNC(7),LFC(17),DSC(400),DSE(400),DSG(2),FIG(17)
1,ENC(110),DNC(110)
COMMON/SGT/SEG,FGC
COMMON/HIA/HA(4)
COMMON/HHQ/HQ,NT
COMMON/NO/GTAG(9000),PCK(9000),SEGCT
COMMON/HB/BLACK(2500),KRED(2500),NCURRY(2500),HTAG(2500)
1,HTAG(2500)
COMMON/QSID/QSID
COMMON/LK/XLK,XHK,YLK,YHK,YQP,YQP
COMMON/STST/STST(99),SAST(99),SASN(99)
COMMON/WH/WH
COMMON/SSS/STNT(5),STH,FST
COMMON/HB/MBINA(99),MBINXL(99),MBINXH(99),MBINYL(99),MBINYH(99),NQ
COMMON/ENO/XEO,YBO,XDT,YDT,PNO,LNO
COMMON/INZ/CHASTE(100),CHACHT
COMMON/POINT/POINT(100)
COMMON/REFER/
1 LID,LABEL,SEQNO,PROGNO,IDNO,DAY,MONTH,YEAR,
2 X,Y,XPI,YPI,NOP,XLI,YLI,NOL,DENL,BON,HVFLAG,
3 COMCH,CHARIN,BOP,ACTFN,REQFN,IFLAG
COMMON/BUFFS/
1 NBUFF(256),MTBUFF(256),
2 JDLACK(64),JRED(64),KDLACK(64),KRED(64),
3 CURRY,ENCL,LINESP,IP,JI,KI
COMMON/SCFLAG/SCFLAG
COMMON/DIAGEN/SQ,HSQ,DD,HDD,CINCR,F,AVDEN
COMMON/QIV/QPARS(4,11)
DIMENSION APARS(11)
DIMENSION CHASTX(25),CHASTY(25)
DIMENSION DHASTX(25),DHASTY(25)
DIMENSION CHASTZ(50)
EQUIVALENCE (DHASTX,POINT),(DHASTY,POINT(26))
EQUIVALENCE (CHASTZ,POINT(51))
EQUIVALENCE (APARS,X)
COMMON/NTCH/ MAGT(36),SLTP(200),SLTFNO(200)
COMMON/PERS/PERSV(2200),ARAS
COMMON/PERSH/PERSH(550)
COMMON/SOTP/SLNAME(5)
DATA(SLNAME(K),K=1,5)/>SOIL>,>TYPE>,>ACRES>,>NO>,>MULTI>/
C
C DISPLAY SOIL TYPE ,ACRAGE,AND BOUNDARY
C SI
IF(HK.EQ.10)STNT(1)=0
QP=PK
IF(QP.EQ.1)QQ=1
G=PERSV(1)
ARAS=0
DO 25 I=7,G-1
ARAS=ARAS+PERSV(I)
25 CONTINUE
ARAS=-ARAS
ARAS=ARAS+G-7
APAU=((1+ARAS*16000)/(330*330))+5)/10

```

```

CALL QLTB(200,200)
30 CONTINUE
FST=0
CALL QEMPTY
CALL QSIZE(6)
28 CONTINUE
IF(STST(QP).GT.0)HK=4
STW=875
CALL QNOP(460,STW)
CALL QNORD(BNC(II))
CALL QNOP(660,STW)
CALL QNORD(DSC(LL))
32 CONTINUE
CALL QNOP(350,925)
CALL QNORD(SLNAME(1))
CALL QNOP(550,925)
CALL QNORD(SLNAME(2))
CALL QNOP(500,725)
CALL QNORD(SLNAME(3))
ENCODE(5,556,BOP)ARAU
556 FORMAT(I5)
CALL QNOP(500,675)
CALL QNORD(BOP)
STV=800
G=PERSV(1)
LXL=PERSV(2)
LYL=PERSV(3)
SLF=PERSV(4)
CZZ=PERSV(5)
DZZ=PERSV(6)
K=7
CALL SSHTCH(9,I9)
LLL=PERSV(K)
FLS=750
CUT=(LLL-LXL)*FLS/SLF+50
CUZ=(DZZ-LYL)*FLS/SLF+50
CALL QNOP(CUT,CUZ)
GGG=DZZ
HHH=LLL
IF(HHH.LT.1)HHH=1
IF(I7.EQ.2)
1PRINT 909,HHH,LLL,XBO,CINCR,GGG,DZZ,YBO
97 K=K+1
IF(K.GT.6) GO TO 101
IF(PERSV(K).LT.0) GO TO 100
DZZ=DZZ+CZZ
98 LLL=PERSV(K)
CUT=(LLL-LXL)*FLS/SLF+50
CUZ=(DZZ-LYL)*FLS/SLF+50
CALL QVECTOR(CUT,CUZ)
HHH=LLL
IF(HHH.LT.1)HHH=1
IF(I9.EQ.2)
1PRINT 939,HHH,LLL,XBO,CINCR
GO TO 97
99 K=K+1
IF(K.GT.6) GO TO 101
IF(PERSV(K).GT.0) GO TO 98
DZZ=DZZ-CZZ
100 LLL=-PERSV(K)

```

```

CUT=(LLL-LXL)*FLS/SLF+50
CUZ=(DZZ-LYL)*FLS/SLF+50
CALL QVECTOR(CUT,CUZ)
HHH=LLL
IF(HHH.LT.1)HHH=1
HHH=-HHH
IF(I9.EQ.2)
1PRINT 999,HHH,LLL,XBO,CINCR
GO TO 99
101 CONTINUE
SOILTP=0
HALT=0
PERD=0
SFRAC=0
JOIN=0
CALL QNOP(0,0)
CALL QNOP(0,0)
CALL QNOP(0,0)
CALL QNOP(0,0)
CALL QSTART
CALL QREP(4)
ATR=0
102 CALL QRTB(CX,CY)
CALL QNOP(CX-32,CY-32)
CALL QVECTOR(CX+32,CY+32)
CALL QNOP(CX-32,CY+32)
CALL QVECTOR(CX+32,CY-32)
CALL QREP(4)
29 CALL RCHAR(D)
IF(D.EQ.0) GO TO 102
CALL QSTOP
IF((ATR.EQ.1).AND.(D.NE.13)) GO TO 229
IF(D.EQ.32) GO TO 103
IF(D.EQ.35) GO TO 180
IF(D.EQ.45) GO TO 103
IF(D.EQ.33)REP=2
IF(D.EQ.33)RETURN
IF(D.EQ.46) GO TO 1103
IF(D.EQ.13) GO TO 52
IF(D.EQ.47) GO TO 50
229 CONTINUE
SOILTP=SOILTP*10+D-48
CALL QSTART
GO TO 102
50 CONTINUE
ATR=1
IF(SOILTP.EQ.0) GO TO 553
DO 51 I=1,MQ
IF(SOILTP.EQ.DMC(I))GO TO 552
51 CONTINUE
TYPE 910
552 CONTINUE
SOILTP=0
II=I
DSG(1)=II
553 CONTINUE
CALL QSTART
GO TO 102
52 CONTINUE
IF((SOILTP.EQ.2108).OR.(SOILTP.EQ.312).OR.(SOILTP.EQ.317).OR.

```

```

IF(SOILTP.EQ.381) OR (SOILTP.EQ.385) OR (SOILTP.EQ.2102) THEN W9
999 FCRHAT1 - SOIL DOUBLE =>
  AIR=0
  IF(SOILTP.EQ.0) GO TO 518B
  DO 517 I=1,N7
  IF(SOILTP.EQ.00017) GO TO 518B
517 CONTINUE
  TYPE 910
910 FCRHAT1 - NO POINT FOUND =>
518 CONTINUE
  LL=1
  DOO2=ALL
518B CONTINUE
  STG1(0)=SCCH(0006(2),DOO2)
  GO TO 77
780 CALL G200P
  CALL STC(00,0)
  DO 654 I=1,0M
  CALL G71M
654 CONTINUE
  CALL G70VE
  CALL G70VE
  D=0
  CALL G70AP
  GO TO 102
C
C BOUNDARY FILTERING
C
103 CONTINUE
  D00=ACC-001*SEC/FIS*LEV
  D10=CY-001*SEF/FLS*LEV
  DIFF =10
  VAC=LEV
  VV=PERD(07)
  DO 105 O=0,5
  LL=PERD(07)
  IF(ALL.LT.0) GO TO 104
  IF(ANY.GT.0) VAC=VAC+1
  VV=ALL
  GO TO 105
104 LL=LEV
  IF(ANY.LT.0) VAC=VAC-1
  VV=LEV
105 XDIFF=ALL-D00
  XDIFF=IABS(XDIFF)
  XDIFF=VAC-D10
  XDIFF=IABS(XDIFF)
  XDIFF=XDIFF+VVF
  IF(DIFF.GT.DIFF) GO TO 106
  DIFF=XDIFF
  D00=XK
106 CONTINUE
  IF(DIFF.LT.4) GO TO 107
  D=0
998 FCRHAT1 - WINDY CONNECTION =>
  TYPE 900,DIFF
  D=0
  CALL G70AP
  GO TO 102
107 CONTINUE

```



```

K=SKU+KK-1
IF(KK.LT.2)K=SKU
DO 113 I=SKV,G
K=K+1
113 PERSV(K)=PERSV(I)
G=K
PERSV(1)=K
IF(I10.EQ.1) GO TO 1117
DO 115 I=1,G
115 PRINT 909,PERSV(I),I
1117 CONTINUE
1100 CONTINUE
IF(I10.EQ.1) GO TO 130
KP=0
KN=0
DO 118 I=7,G-1
IF(PERSV(I).GT.0)KP=KP+1
118 IF(PERSV(I).LT.0)KN=KN+1
TYPE 989,KP,KN
C
C RECALCULATE AREA
C
130 CONTINUE
G=PERSV(1)
ARAS=ARAS
APAS=0
DO 131 I=7,G-1
131 ARAS=ARAS+PERSV(I)
APAS=-APAS
APAS=APAS+G-7
ARAU=((ARAS+15000)/(330*330))+5)/10
GO TO 30
C
C SPLIT JOINED REGIONS
C
120 CONTINUE
G=PERSV(1)
GG=G
NCURR =PERSV(6)
DO 2121 I=1,2500
NSLACK(I)=0
NRED(I)=0
2121 NCURRY(I)=0
MAXX=0
MAXY=0
MINY=1000000
MINX=1000000
HAPT=1
NT=0
DO 121 I=8,G-1
IF((PERSV(I).GT.0).AND.(PERSV(I-1).GT.0))NCURR =NCURR +1
IF((PERSV(I).LT.0).AND.(PERSV(I-1).LT.0))NCURR =NCURR -1
IF(I.LE.SKU) GO TO 121
IF(I.GT.SKV)GO TO 122
HT=NT+1
IF(PERSV(I).LT.0)NRED(HT)=-PERSV(I)
IF(PERSV(I).GT.0)NSLACK(HT)=PERSV(I)
NCURRY(HT)=NCURR
PERX=IADS(PERSV(I))
IF(PERX.LT.MINX)MINX=PERX

```

```

IF (PEPX .GT. MAXX) MAXX=PEPX
IF (NCURR .LT. MINY) MINY=NCURR
IF (NCURR .GT. MAXY) MAXY=NCURR
121 CONTINUE
122 CONTINUE
YLL=MINX
XHH=MAXX
YHH=MAXY
YLL=MINY
DO 2101 I=1,MT-1
IF (NCURRY(I+1) .GE. NCURRY(I)) GO TO 2101
CHAS=NCURRY(I+1)
CHAT=NBLACK(I+1)
CHAV=NRED(I+1)
NCURRY(I+1)=NCURRY(I)
NBLACK(I+1)=NBLACK(I)
NRED(I+1)=NRED(I)
J=I-1
2099 CONTINUE
IF (J.EQ.0) GO TO 2098
IF (CHAS.GT.NCURRY(J)) GO TO 2098
NCURRY(J+1)=NCURRY(J)
NBLACK(J+1)=NBLACK(J)
NRED(J+1)=NRED(J)
J=J-1
GO TO 2099
2098 CONTINUE
NCURRY(J+1)=CHAS
NBLACK(J+1)=CHAT
NRED(J+1)=CHAV
2101 CONTINUE
K=0
I=0
2104 I=I+1
IF (I.GT.MT) GO TO 2106
IF (NCURRY(I).NE.NCURRY(I+1)) GO TO 2105
K=K+1
IF (NBLACK(I).EQ.0) NBLACK(K)=NBLACK(I+1)
IF (NBLACK(I).NE.0) NBLACK(K)=NBLACK(I)
IF (NRED(I).EQ.0) NRED(K)=NRED(I+1)
IF (NRED(I).NE.0) NRED(K)=NRED(I)
NCURRY(K)=NCURRY(I)
NBL=NBLACK(K)
NRD=NRED(K)
I=I+1
GO TO 2104
2105 CONTINUE
K=K+1
IF (NBLACK(I).EQ.0) NBLACK(K)=NBL
IF (NRED(I).EQ.0) NRED(K)=NRD
IF (NBLACK(I).NE.0) NBLACK(K)=NBLACK(I)
IF (NRED(I).NE.0) NRED(K)=NRED(I)
NCURRY(K)=NCURRY(I)
NBL=NBLACK(K)
NRD=NRED(K)
GO TO 2104
2106 CONTINUE
MT=K
K=0
G=1

```



```

      DO 2120 I=2,MT
      IF(NCURLY(I).EQ.NCURLY(I-1)) GO TO 2120
      N=I-1
      P=N-G
      IF(P.EQ.0) GO TO 2119
2121 CONTINUE
      INVT=0
      DO 2122 J=G,N-1
      IF(NRED(J+1).GE.NRED(J)) GO TO 2122
      NGL=NRED(J)
      NRED(J+1)=NGL
      INVT=1
2122 CONTINUE
      IF(INVT.EQ.1) GO TO 2123
2119 CONTINUE
      G=I
2120 CONTINUE
      G=1
      DO 2110 I=2,MT
      IF(NCURLY(I).EQ.NCURLY(I-1)) GO TO 2110
      N=I-1
      P=N-G
      IF(P.EQ.0) GO TO 2109
2111 CONTINUE
      INVT=0
      DO 2112 J=G,N-1
      IF(NBLACK(J+1).GE.NBLACK(J)) GO TO 2112
      NGL=NBLACK(J)
      NBLACK(J+1)=NGL
      INVT=1
2112 CONTINUE
      IF(INVT.EQ.1) GO TO 2111
2109 CONTINUE
      G=I
2110 CONTINUE
      DO 2124 I=2,MT
      IF(NCURLY(I).NE.NCURLY(I-1)) GO TO 2124
      IF(NBLACK(I).LT.NRED(I-1)) NBLACK(I)=NRED(I-1)+5
2124 CONTINUE
      SEG=MT
      PIP=1
      G=GGG
      GO TO 2115
C
C DISPLAY REGION ON LEFT SCOPE
C
1103 CONTINUE
      CALL SWITCH(32,132)
      CALL SWITCH(33,133)
      IF((132.EQ.1).AND.(133.EQ.1))RETURN
      IF(133.EQ.2) GO TO 300
      IF(STSQP).EQ.0)TYPE 905
905 FORMAT(> SOIL MUST BE IDENTIFIED>)
      IF(STSQP).EQ.0) GO TO 300
300 CONTINUE
      S(STSQP)=ARAS
      SASH(QP)=G-8
      TQ=QP

```

```

      IF(QQ.NE.1) GO TO 1108
      DO 1107 I=1,550
1107 PERSM(I)=0
      QQ=0
      QD=0
      PP=2
      PERSM(1)=QSID
      IF(I24.EQ.2)PERSM(2)=ICONB(FNO,LNO)
      SQS= 4000000000
1108 CONTINUE
      QS=QS+1
      FP=FP+1
      IF(FP.LT.551)PERSM(FP)=SQS+QS
      DO 1105 I=6,6-1,2
      P=PP+1
      IF(FP.LE.550) GO TO 1110
      IF(I33.EQ.2)CALL MT7M(PERSM,550)
      IF(I32.EQ.2)CALL MT9M(PERSM,550)
      PRINT 939,PERSM(1),PERSM(2)
      DO 4313 K=3,550
      IF(PERSM(K).GE.SQS)PRINT 942,PERSM(K-1),PERSM(K),PERSM(K+1)
4313 CONTINUE
      942 FORMAT(30I3)
      DO 1112 K=1,550
1112 PERSM(K)=0
      PERSM(1)=QSID
      PERSM(2)=1
      P=FP
      IF(FP.EQ.552) TYPE 932
      932 FORMAT ( > PFP IS EQUAL TO 552 > )
      P=3
      IF(FP.EQ.552)PERSM(3)=SQS+QS
      IF(FP.EQ.552)P=4
1110 PERSM(FP)=ICONB(PERSV(I),PERSV(I+1) )
1105 CONTINUE
      RETURN
      939 FORMAT(8I12)
      END
      SUBROUTINE CLOSOUT(TOT,PK)
      IMPLICIT INTEGER (A-Y)
      COMMON/BDC/BDC(4)
      COMMON/QS/QS
      COMMON/TART/TART
      COMMON/XPZ/XQT,YQT
      COMMON/QSID/QSID
      COMMON/LK/XLK,YHK,YLK,YHK,XQP,YQP
      COMMON /INZ/ CHASTE(100),CHACT
      COMMON/STST/STST(99),SAST(99),SASN(99)
      COMMON/SSS/STNT(5),STN,FST
      COMMON /DIANEH/ SQ,NSQ,DD,HDD,CINCR,F,AVDEN
      COMMON/BND/XBD,YBD,XDT,YDT,FNO,LNO
      COMMON /REFER/
      1   LTD,LABEL,SEQNO,FROGNO,ICHO,DAY,MONTH,YEAR,
      2   X,Y,XPI,YPI,NOP,XLI,YLI,NCL,DENL,BOW,HVFLAG,
      3   COMMON,CHARIN,BOP,ACTFN,REQFN,IFLAG
      COMMON /BUFFS/
      1   NCUFF(256),MTBUFF(256),
      2   JBLACK(64),JRED(64),KBLACK(64),KRED(64),
      3   CURRY,ENCL,LINESP,IP,JI,KI
      COMMON /SCFLAG/ SCFLAG

```

```

COMMON /BPARS/ MMMM,MINLEN,MINA
COMMON /QTV/ QPARS(4,11)
COMMON /MB/MBINA(99),MBINXL(99),MBINXH(99),MBINYL(99),MBINYH(99),NQ
DIMENSION APARS(11)
COMMON /PERD/PERSV(2200),ARAS
COMMON /PERM/PERSM(550)
EQUIVALENCE (APARS(1),X)

C
C RECALCULATE AREA
C
      NQ=PK
      IF(QS.NE.PK)TYPE 969
969  FORMAT(> QS NE PK>)
      ACQV=0
      ACQT=(1600*PND*LNQ)/(330*330)
      TART=((TART+16000)/(330*330))+5)/10
      ACQS=ACQT-TART-20
      DO 1 K=1,NQ
1     ACQV=ACQV+SAST(K)
      ACQV=((ACQV*16000)/(330*330))+5)/10
      DIFAC=ACQT-ACQV
      TYPE 989,ACQT,ACQV,ACQS,TART
4     CONTINUE
      ARSUM=0
      DO 5 K=1,NQ
5     ARSUM=ARSUM+SAST(K)
      ARSUM=((ARSUM*16000)/(330*330))+5)/10
      IF(ARSUM.GT.ACQS) GO TO 7
      DO 6 K=1,NQ
6     SAST(K)=SAST(K)+SASH(K)
      GO TO 4
7     CONTINUE
      ARSUM=0
      DO 8 K=1,NQ
      SAST(K)=((SAST(K)*16000)/(330*330))+5)/10
      ARSUM=ARSUM+SAST(K)
8     CONTINUE
      CALL SSHTCH(24,I24)
      IF(I24.EQ.1) GO TO 81
      SASUM=0
      DO 80 K=1,NQ
      SAST(K)=(SAST(K)*36/16+5)/10
      SASUM=SASUM+SAST(K)
80    CONTINUE
      TYPE 989,SASUM
81    CONTINUE

C
C IF SSHTCH 32 IS ON PACK AREA AND OUTPUT TO MAGTAPE
C
      CALL SSHTCH(33,I33)
      CALL SSHTCH(32,I32)
      IF((I32.EQ.1).AND.(I33.EQ.1))RETURN
4113 CONTINUE
102  CONTINUE
      RFL=0
      DO 203 I=3,550
      IF(PERSM(I).EQ.0) GO TO 204
203  CONTINUE
      RFL=1
      TYPE 925

```

```

925 FORMAT ( > LAST BUFFER FULL PERSH > )
GO TO 205
204 PERSH(1)=_4000000000
SAS= 4000000000
205 CONTINUE
IF(I12.EQ.2)CALL MT9W(PERSH,550)
IF(I13.EQ.2)CALL MT7W(PERSH,550)
PRINT 909,PERSH(1),PERSH(2)
DO 4313 K=3,550
IF(PERSH(K).GE.SAS)PRINT 932,PERSH(K-1),PERSH(K),PERSH(K+1)
4313 CONTINUE
932 FORMAT(8013)
IF(PEL.EQ.0) GO TO 204
PERSH(3)=_4000000000
IF(I12.EQ.2)CALL MT9W(PERSH,550)
IF(I13.EQ.2)CALL MT7W(PERSH,550)
206 CONTINUE
PERSH(1)=OSID
PERSH(2)=2
PERSH(3)=ICOMB(PND,LND)
K=4
COMPA=0
DO 8000 I=1,N0
IF(STST(I).EQ.99) GO TO 8000
I=I+1
COMPA=OSARA+SAST(I)
PERSH(K)=SAST(I)
8000 CONTINUE
PERSH(4)=K
DO 3001 I=K+1,550
3001 PERSH(I)=0
IF(I12.EQ.2)CALL MT9W(PERSH,550)
IF(I13.EQ.2)CALL MT7W(PERSH,550)
3002 CONTINUE
PERSH(1)=OSID
PERSH(2)=3
PERSH(3)=0
K=4
DO 8003 I=1,N0
IF(STST(I).EQ.99) GO TO 8003
I=I+1
PERSH(K)=STST(I)
8003 CONTINUE
PERSH(4)=K
DO 95 I=1,4
K=K+1
95 PERSH(K)=DCC(I)
DO 8004 I=K+1,550
8004 PERSH(I)=0
IF(I12.EQ.2)CALL MT9W(PERSH,550)
IF(I13.EQ.2)CALL MT7W(PERSH,550)
RETURN
999 FORMAT(8I10)
END
SUBROUTINE DTA
IMPLICIT INTEGER (A-Y)
COMMON/LFC/PIC(7),LFC(17),DSC(400),DSE(400),DSG(2),FIG(17)
1,DNC(110),DNC(110)
COMMON/PND/YQ,NT
DATA YQ,NT/99,350/

```

DATA(DMC(K),K=1,99)/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
 117,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,
 237,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,
 357,53,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
 473,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99/
 DATA(BMC(K),K=1,99)/>1>,>2>,>3>,>4>,>5>,>6>,>7>,>8>,>9>,>10>,>11>,
 1>12>,>13>,>14>,>15>,>16>,>17>,>18>,>19>,>20>,>21>,>22>,>23>,>24>,
 2>25>,>26>,>27>,>28>,>29>,>30>,>31>,>32>,>33>,>34>,>35>,>36>,
 3>37>,>38>,>39>,>40>,>41>,>42>,>43>,>44>,>45>,>46>,>47>,>48>,
 4>49>,>50>,>51>,>52>,>53>,>54>,>55>,>56>,>57>,>58>,>59>,>60>,
 5>61>,>62>,>63>,>64>,>65>,>66>,>67>,>68>,>69>,>70>,>71>,>72>,
 6>73>,>74>,>75>,>76>,>77>,>78>,>79>,>80>,>81>,>82>,>83>,>84>,
 7>85>,>86>,>87>,>88>,>89>,>90>,>91>,>92>,>93>,>94>,>95>,>96>,
 8>97>,>98>,>99>/

DATA(DSE(K),K=1,300)/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
 117,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,
 237,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,
 357,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,
 478,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,
 5100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,
 6116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,
 7132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,
 8148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,
 9164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,
 10180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,
 11196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,
 12212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,
 13228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,
 14244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,
 15260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,
 16276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,
 17292,293,294,295,296,297,298,299,300/
 DATA(DSE(K),K=301,350)/301,302,303,304,305,306,307,308,309,310,
 1311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,
 14327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,
 15343,344,345,346,347,348,349,350/
 DATA(DSC(K),K=1,206)/>1>,>2>,>3>,>4>,>5>,>6>,>7>,>8>,>9>,>10>,>11>
 1>12>,>13>,>14>,>15>,>16>,>17>,>18>,>19>,>20>,>21>,>22>,>23>,>24>,
 2>25>,>26>,>27>,>28>,>29>,>30>,>31>,>32>,>33>,>34>,>35>,>36>,
 3>37>,>38>,>39>,>40>,>41>,>42>,>43>,>44>,>45>,>46>,>47>,>48>,
 4>49>,>50>,>51>,>52>,>53>,>54>,>55>,>56>,>57>,>58>,>59>,>60>,
 5>61>,>62>,>63>,>64>,>65>,>66>,>67>,>68>,>69>,>70>,>71>,>72>,
 6>73>,>74>,>75>,>76>,>77>,>78>,>79>,>80>,>81>,>82>,>83>,>84>,
 7>85>,>86>,>87>,>88>,>89>,>90>,>91>,>92>,>93>,>94>,>95>,>96>,
 8>97>,>98>,>99>,>100>,>101>,>102>,>103>,>104>,>105>,>106>,>107>,
 9>108>,>109>,>110>,>111>,>112>,>113>,>114>,>115>,>116>,>117>,>118>,
 A>119>,>120>,>121>,>122>,>123>,>124>,>125>,>126>,>127>,>128>,>129>,
 B>130>,>131>,>132>,>133>,>134>,>135>,>136>,>137>,>138>,>139>,>140>,
 C>141>,>142>,>143>,>144>,>145>,>146>,>147>,>148>,>149>,>150>,>151>,
 D>152>,>153>,>154>,>155>,>156>,>157>,>158>,>159>,>160>,>161>,>162>,
 E>163>,>164>,>165>,>166>,>167>,>168>,>169>,>170>,>171>,>172>,>173>,
 F>174>,>175>,>176>,>177>,>178>,>179>,>180>,>181>,>182>,>183>,>184>,
 G>185>,>186>,>187>,>188>,>189>,>190>,>191>,>192>,>193>,>194>,>195>,
 H>196>,>197>,>198>,>199>,>200>,>201>,>202>,>203>,>204>,>205>,>206>/
 DATA(DSC(K),K=207,350)/
 I>207>,>208>,>209>,>210>,>211>,>212>,>213>,>214>,>215>,>216>,>217>,
 J>218>,>219>,>220>,>221>,>222>,>223>,>224>,>225>,>226>,>227>,>228>,
 K>229>,>230>,>231>,>232>,>233>,>234>,>235>,>236>,>237>,>238>,>239>,
 L>240>,>241>,>242>,>243>,>244>,>245>,>246>,>247>,>248>,>249>,>250>,
 M>251>,>252>,>253>,>254>,>255>,>256>,>257>,>258>,>259>,>260>,>261>,

```

N>262>,>263>,>264>,>265>,>266>,>267>,>268>,>269>,>270>,>271>,>272>,>
O>273>,>274>,>275>,>276>,>277>,>278>,>279>,>280>,>281>,>282>,>283>,>
P>284>,>285>,>286>,>287>,>288>,>289>,>290>,>291>,>292>,>293>,>294>,>
Q>295>,>296>,>297>,>298>,>299>,>300>,>301>,>302>,>303>,>304>,>
1>305>,>306>,>307>,>308>,>309>,>310>,>311>,>312>,>313>,>314>,>
2>315>,>316>,>317>,>318>,>319>,>320>,>321>,>322>,>323>,>324>,>
3>325>,>326>,>327>,>328>,>329>,>330>,>331>,>332>,>333>,>334>,>
4>335>,>336>,>337>,>338>,>339>,>340>,>341>,>342>,>343>,>344>,>
4>345>,>346>,>347>,>348>,>349>,>350>/

```

```
END
```

```
C   MAIN      MAIN      MAIN      MAIN      MAIN      MAIN      MAIN      MAIN
```

```
C
```

```
THIS PROGRAM DISPLAYS SOIL TYPES
```

```
C
```

```
C
```

```
IMPLICIT INTEGER (A-Y)
```

```
COMMON /MNQ/MQ,NT
```

```
COMMON /MS/MS
```

```
COMMON /MT7F/ MT7F
```

```
COMMON /MT9F/MT9F
```

```
COMMON /QSIO/OSIO
```

```
COMMON /MN/MN,MO, POS
```

```
COMMON /MXY/MINX,MINY,MAXX,MAXY,RA,RB,RO,BP
```

```
COMMON /BDC/BDC(4)
```

```
COMMON /OAE/POI,NR,SOI,QERP,MTQ
```

```
COMMON /XB/XBUF(2200),YBUF(2200),XBS(50)
```

```
COMMON /DH/DIND(9999),DIME(9999)
```

```
COMMON /SEF/SOB(550),FRT(100),FRS(100),FRSS(100)
```

```
COMMON /BF/Q3(6000),OB(6000)
```

```
COMMON /LFC/PMC(7),LFC(17),DSC(99),DSE(99),DSG(2),FIG(17)
```

```
1,DNC(110),DMC(110)
```

```
DIMENSION PERSV(2200),SLTF(200),STN(200),FRS(100)
```

```
DIMENSION GST(5),QID(3),STIT(12),STL(26),STOL(26),BUF(15)
```

```
SSW20 SELECTS MODE FOR MINOR ATTRIBUTE IDENT
```

```
SSW32 ON OUTPUT TO 9TK TAPE
```

```
PROGRAM OPERATION
```

```
TYPE IN A MAJOR ATTRIBUTE, MOVE CROSS WITH TRACK BALL BY  
DEPRESSING SPACE BAR WHENIN POLYGON WITH ATTRIBUTE. POLYGON WILL  
BE DELETED FROM SCREEN. CONTINUE THIS OPERATION UNTIL
```

```
ALL POLYGONS WITH SHOWN ATTRIBUTE HAVE BEEN REMOVED FROM SCREEN.  
TYPE IN NEXT ATTRIBUTE AND CONTINUE ABOVE OPERATION. IF POLYGON
```

```
CANT BE FOUND WHICH IS SIGNIFIED BY TTY MESSAGE MOVE CROSS  
TO POINT ON POLYGON DEPRESS SPACE BAR AGAIN. WHEN ALL POLYGONS H
```

```
HAVE BEEN REMOVED FROM SCREEN. DEPRESS SLASH. PROGRAM ENTERS  
CHECKING MODE ALL POLYGONS OF GIVEN ATTRIBUTE ARE SHOWN.
```

```
DEPRESS SLASH UNTIL POLYGONS ALL ATTRIBUTES ARE SHOWN.
```

```
PROGRAM THEN ENTERS MINOR ATTRIBUTE MODE. WITH SSW 20 ON
```

```
TYPE 2N MINOR ATTRIBUTE DEPRESS SPACE BAR AS BEFORE. WHEN
```

```
ALL POLYGONS HAVE BEEN REMOVED TYPE IN PERIOD PROGRAM
```

```
ENTERS CHECKING MODE. DEPRESS KEY PERIOD UNTIL COMPLETE
```

```
NOTE AT ANY PLACE ALL POLYGONS CAN BE VIEWED BY KEYING LINEFEED.
```

```
WITH SSW20 OFF EACH POLYGON IS ENHANCED WITH ALL POLYGONS SHOWN.
```

```
TYPE IN MINOR ATTRIBUTE ASSOCIATED WITH ENHANCED POLYGON
```

```
WHEN ALL POLYGONS HAVE BEEN IDENTIFIED PERIOD KEY PERIOD
```

```
PROGRAMS ENTERS MINOR ATTRIBUTE CHECKING MODE AS BEFORE
```

```
WH=103
```

```
BSQ=_4000000000
```

```
C
```

```
CALL SSWTCH(32,132)
```

```
CALL MT7OPEN
```

```
CALL MT9OPEN
```

```

S=1
MG=22
NOP=330
NOL=330
NR=1
NTOQ=1
QEPB=1
TYPE 1100
1100 FORMAT(>>DO YOU WANT TO POSITION 7 TK TAPE Y OR N >)
1102 CALL RCHAR(FEF)
      IF(FEF.EQ.0) GO TO 1102
      IF(FEF.EQ.78) GO TO 1
      IF(FEF.EQ.89) GO TO 1104
      TYPE 1705
      GO TO 1102
1104 CONTINUE
      TYPE 987
      987 FORMAT( > TYPE LAST QSECT NUMBER >)
      TYPE 1714
1714 FORMAT(> IDX=>*)
      CALL RDEC(QQ,IDX)
      TYPE 1715
1715 FORMAT(> IDY=>*)
      CALL RDEC(QQ,IDY)
      IDEN=TCONS(IDX,IDY)
1717 CALL NT7R(SQB,550)
      IF(SQB(1).NE.IDEN) GO TO 1717
      IF(SQB(2).NE.3) GO TO 1717
      CALL SSHTCH(30,130)
      IF(130.EQ.1) GO TO 1
1718 CALL NT0R(SQB,550)
      IF(SQB(1).NE.IDEN) GO TO 1718
      IF(SQB(2).NE.3) GO TO 1718
      1 CONTINUE
      CALL SSHTCH(20,120)
      TYPE 1801
1801 FORMAT( > READ IN NEXT QSECT , Y OR N >)
1802 CALL RCHAR(D)
      IF(D.EQ.0) GO TO 1802
      IF(D.EQ.78 ) GO TO 123
      MN=0
      MO=0
      FOS=0
      CALL SSHTCH(24,124)
      CALL QSTOP
      CALL QEMPTY
101 CONTINUE
C
C
C   START THE UNPACKING OF THE DATA
C
ONE=0
2 CONTINUE
  ENT=0
3 CONTINUE
  ENT=ENT+1
  CALL NT7R(SQB,550)
  IF(SQB(1).EQ.5100273664) GO TO 3
  IF(ENT.EQ.2) GO TO 123
  CALL SSHTCH(19,119)

```

```

IF(I19.EQ.1) GO TO 4170
ONE=ONE+1
PRINT 989,ONE
PRINT 981,SQB
981 FORMAT(1X,10013)
4170 CONTINUE
IF(SQB(2).GT.1) GO TO 108
IF(SQB(2).EQ.0)CALL CLPBF
IF(I32.EQ.2)CALL HT9H(SQB,550)
PERSV(1)=SQB(1)
PERSV(2)=1
BN=3
IF(SQB(2).EQ.1) GO TO 102
CALL QEMPTY
PERSV(3)=SQB(3)-BSQ
BN=4
M=4
102 CONTINUE
DO 103 K=BN,550
IF(SQB(K).LT.BSQ) GO TO 105
104 CONTINUE
PERSV(4)=M
CALL IRLEFT(PRT(PERSV(3)+4),POI)
C
C          PACK MAG TAPE DUFFERS INTO QB
C          POI IS THE QB POINTER
C          POI IS PLACED INTO THE LEFT HALF OF PRT
C
DO 4 I=1,PERSV(4)
QB(POI)=PERSV(I)
POI=POI+1
IF(POI.GT.6000)TYPE 959
959 FORMAT(> POI GT 6000>)
4 CONTINUE
522 CONTINUE
IF(SQB(K).EQ.BSQ) GO TO 2
PERSV(3)=SQB(K)-BSQ
BSQ=PERSV(3)
M=4
GO TO 103
105 CONTINUE
M=M+1
PERSV(M)=SQB(K)
103 CONTINUE
GO TO 2
108 CONTINUE
DO 107 K=1,550
107 PERSV(K)=SQB(K)
IF(PERSV(2).EQ.3) GO TO 30
C
20 DO 22 I=1,PERSV(4)
PRV(I)=PERSV(I)
22 CONTINUE
MINY=1
MINX=1
MAXX=IRLEFT(PERSV(3))
MAXY=IRIGHT(PERSV(3))
IF(I32.EQ.2)CALL HT9H(SQB,550)
GO TO 2
C

```



```

C           THE SOIL TYPE IS PLACED IN THE RIGHT HALF OF PRT
C
30 CONTINUE
   DO 32 I=1,4
     PRT(I)=PERSV(I)
32 CONTINUE
   IDB=0
   DO 36 I=5,PERSV(4)
     PRP=IPRIGHT(PERSV(I))
     PST=ILEFT(PERSV(I))
     CALL IPRIGHT(PRT(I),PRP)
     CALL ILEFT(PRS(I),PST)
     IF(PST.NE.0)IDB=78
36 CONTINUE
   EST=PERSV(4)-4
   XAB=ILEFT(PERSV(1))
   YAB=IRIGHT(PERSV(1))
   TYPE 999,XAB,YAB
   GG=PERSV(4)+1
   HH=PERSV(4)+4
   J=0
   DO 336 I=GG,HH
     J=J+1
     PBC(J)=PERSV(I)
336 CONTINUE
C
37 CONTINUE
   CALL SSATCH (26,I26)
   IF (IC6.EQ.1) GO TO 55
   DO 106 I=1,550
     PRP=ILEFT(QB(I))
     PRK=IRIGHT(QB(I))
106 PRINT 999,PRP,PRK,I
     DO 54 I=5,PRT(4)
       PRP=ILEFT(PRT(I))
       PST=IRIGHT(PRT(I))
       PRINT 1711, PRP,PST
54 CONTINUE
55 CONTINUE
   NN=0
C
C           UNPACK DATA AND PUT IN XY FORMAT
C
   DO 5 I=5,100
     PRP=ILEFT(PRT(I))
     IF (PRP.EQ.0) GO TO 10
     PRK=IRIGHT(QB(PRP+3))
     YBUF(1)=ILEFT(QB(PRP+4))
     XBUF(1)=IRIGHT(QB(PRP+4))
     L=1
     PRK=PRP+PRK-1
     DO 6 K=PRP+5,PRK
       L=L+1
       YBUF(L)=ILEFT(QB(K))
       XBUF(L)=IRIGHT(QB(K))
6 CONTINUE
     LL=L
     IF(LL.GT.2200)TYPE 971
971 FORMAT(> LL GT 2200>)

```

```

DO 7 L=2,LL
YBUF(L)=YBUF(L-1)
IF((XBUF(L).LT.0).AND.(XBUF(L-1).LT.0))YBUF(L)=YBUF(L-1)-1
IF((XBUF(L).GT.0).AND.(XBUF(L-1).GT.0))YBUF(L)=YBUF(L-1)+1
7 CONTINUE
DO 76 L=1,LL
76 XBUF(L)=IABS(XBUF(L))
DO 77 L=1,LL
IF((XBUF(L-1).EQ.XBUF(L+1)).AND.(XBUF(L-1).EQ.XBUF(L)))GO TO 77
IF((YBUF(L-1).EQ.YBUF(L+1)).AND.(YBUF(L-1).EQ.YBUF(L)))GO TO 77
MN=MN+1
DIMO(MN)=ICOMB(XBUF(L),YBUF(L))
DIME(MN)=I-4
77 CONTINUE
5 CONTINUE
10 CONTINUE
TYPE 989,MN
QUICK=0
RATT=-1
NOTE=0
RABT=-1
ATB=0
697 CONTINUE
DO 698 I=1,9999
698 QB(I)=0
CALL QLTB(100,100)
699 CONTINUE
CALL QEMPTY
CALL QSIZE(6)
CALL QNOP(350,975)
CALL QNORD(ATB)
CALL QNOP(350,975)
CALL QNORD(ATB)
CALL QNOP(350,975)
CALL QNORD(ATB)
DO 785 I=1,MN
IF((QB(I).NE.0).AND.(RABT.NE.1)) GO TO 785
IF(NOTE.EQ.0) GO TO 6785
IF(QB(I).NE.QSD) GO TO 785
QB(I)=0
6785 CONTINUE
XA=ILEFT(DIMO(I))
YB=IRIGHT(DIMO(I))
IF(DIME(I).NE.DIME(I-1)) GO TO 784
CALL QVECTOR(XA*MG/10+50,YB*MG/10+50)
GO TO 785
784 CONTINUE
CALL QNOP(XA*MG/10+50,YB*MG/10+50)
785 CONTINUE
CALL QREADI(IPAR)
IF(QUICK.EQ.1) GO TO 158
CFP=0
783 CONTINUE
SOILIP=0
786 CONTINUE
CALL QRTB(CX,CY)
CALL QSETI(IPAR)
CALL QNOP(CX-32,CY-32)
CALL QVECTOR(CX+32,CY+32)
CALL QNOP(CX-32,CY+32)

```

```

CALL QVECTORCK=32,CY=32)
CALL QNCE
CALL QNAT
CALL PCHURIDA
IF(D.EQ.0) GO TO 786
CALL QSTOP
IF(D.EQ.10)RBT=-RBT
IF(D.EQ.10) GO TO 699
IF(D.EQ.13) GO TO 62
IF(D.EQ.27) GO TO 700
  IF(D.EQ.33) GO TO 95
IF(D.EQ.55) GO TO 320
IF(D.EQ.47) GO TO 45
SDILTP=SDILTP+10-D-48
GO TO 705
95 CONTINUE
CALL PCKCHQ,QM)
DO 645 I=1,M
CALL WFILM
645 CONTINUE
CALL QMOVE
CALL QMOVE
CALL QSTART
GO TO 705
62 CONTINUE
IF(PATT.EQ.1) GO TO 64
DO 63 I=1,M2
IF(SDILTP.EQ.QMCH(I)) GO TO 64
63 CONTINUE
TYPE 910
210 FCHMTI> NO ATTRIBUTE FOUND->
GO TO 783
64 CONTINUE
ATB=QMCH(I)
ATC=1
GO TO 699
65 CONTINUE
NOTE=1
RBT=1
DO 66 I=1,M1
IF(CO(I).EQ.0) GO TO 665
QSD=CO(I)
ATB=QMCH(QSD)
GO TO 699
665 CONTINUE
NOTE=0
RBT=-RBT
RATT=-RATT
IF(I20.EQ.2) GO TO 697
QUICK=1
GO TO 697
158 CONTINUE
CALL QSETI(IPAR)
GG=1
DO 165 I=2,MH+1
IF(DIME(I).EQ.DIME(I-1)) GO TO 165
MH=I-1
EE=0
164 CONTINUE
EE=CE+1

```

```

DPHSE DT,40 DT DT 16Z
  H=CLERT(DDPHDSE)
  H=DPD(DDPHDSE)
  CALL DDPH(40*H0/10+50,H0*H0/10+50)
  GO TO 1400*1,400
  H=CLERT(DDPHDSE)
  H=DPD(DDPHDSE)
763 C=1, 1=DDPH(40*H0/10+50,H0*H0/10+50)
  TO TO 100
762 C=1, 1=1
  CALL DDPH(40)
761 C=1, 1=1
  C=1, 1=1 GO TO 764
  1=1, 1=1
  D=1, 1=1 GO TO 772
  1=1, 1=1 GO TO 76
  1=1, 1=1 GO TO 76
  GO TO 761
76 CONTINUE
  CALL DDPH(40,40)
  GO 400 1=1,400
  CALL DDPH
764 CONTINUE
  CALL DDPH
  CALL DDPH
  CALL DDPH
  GO 45 761
773 CONTINUE
  C=1, 1=1, 1=1
  D=1, 1=1, 1=1 GO TO 779
768 CONTINUE
  TYPE 112
  GO TO 112
759 CONTINUE
  D=1, 1=1, 1=1
  ATC=1
  CALL DDPH(40,40,40,40,40)
  GO 117 1=1,40,40
757 C=1, 1=1, 1=1
  C=1
  CALL DDPH(40,40)
765 CONTINUE
  QUICK=0
  GO TO 699
766 CONTINUE
  GO 67 I=1,NT
  IF(SOILTP.EQ.DSE(I)) GO TO 68
767 CONTINUE
  TYPE 910
  GO TO 783
768 CONTINUE
  ATD=DSE(I)
  ATC=1
  GO TO 699
700 CONTINUE
  D=1=(CX-50)*10/MG
  D=1=(CY-50)*10/MG
  K=0
  GO 701 I=2,MN

```

```

IF(DIME(I-1).NE.DIME(I)) GO TO 701
YB=IRIGHT(DIMO(I-1))
YC=IRIGHT(DIMO(I))
IF((YB.GE.DAV).AND.(YC.LT.DAV)) GO TO 702
IF((YC.GE.DAV).AND.(YB.LT.DAV)) GO TO 702
GO TO 701
702 CONTINUE
XA=ILEFT(DIMO(I))
K=K+1
SLTP(K)=DIME(I)
STN(K)=XA
701 CONTINUE
KK=K
DO 705 I=1,KK-1
IF(STN(I+1).GE.STN(I)) GO TO 703
TMP=STN(I+1)
THK=SLTP(I+1)
STN(I+1)=STN(I)
SLTP(I+1)=SLTP(I)
J=I-1
704 IF(J.EQ.0) GO TO 705
IF(TMP.GT.STN(J)) GO TO 705
STN(J+1)=STN(J)
SLTP(J+1)=SLTP(J)
J=J-1
GO TO 704
705 CONTINUE
STN(J+1)=TMP
SLTP(J+1)=THK
703 CONTINUE
DO 706 I=2,KK
IF(SLTP(I-1).NE.SLTP(I)) GO TO 706
XA=STN(I-1)
YD=STN(I)
IF((XA.LE.DAV).AND.(XB.GE.DAV)) GO TO 707
706 CONTINUE
CFP=CFP+1
IF(CFP.EQ.2) GO TO 711
TYPE 959
969 FORMAT(> CANT FIND POLYGON<)
GO TO 783
711 CONTINUE
DIFB=100
DO 710 I=1,KN
XA=ILEFT(DIMO(I))
YB=IRIGHT(DIMO(I))
XDIF=XA-DAV
XDIF=IABS(XDIF)
YDIF=YB-DAV
YDIF=IABS(YDIF)
DIF=XDIF+YDIF
IF(DIF.GT.DIFB) GO TO 710
DIFB=DIF
II=I
710 CONTINUE
DLP=DIME(II)
707 CONTINUE
IF(CFP.NE.2)DLP=SLTP(I)
CFP=0
DO 709 I=1,KN

```

```
709 IF(DIME(I).EQ.DLP)QB(I)=ATC
      IF(RATT.NE.1)CALL IRRIGHT(SQB(DLP+4),ATC)
      IF(RATT.EQ.1)CALL IRLEFT(SQB(DLP+4),ATC)
      GO TO 699
120 CONTINUE
      NOTE=1
      RABT=1
      DO 122 I=1,NN
      IF(QB(I).EQ.0) GO TO 122
      QSD=QB(I)
      ATB=DSC(QSD)
      GO TO 699
122 CONTINUE
      RABT=-RABT
      NOTE=0
      CALL SSWTCH(32,I32)
      IF(I32.EQ.2)CALL MT9W(SQB,550)
      DO 124 I=1,550
124  SQB(I)=0
      GO TO 1
123 CONTINUE
      CALL QSTOP
      CALL MT7CLOSE
      CALL MT9CLOSE
      CALL EXIT
1701 FORMAT (>+SOIL TYPE ERROR>)
1702 FORMAT (>+ *>,$)
1703 FORMAT (>+ >>,$)
1704 FORMAT (>+WRITE 7TK? (Y OR N) >,$)
1705 FORMAT (>+? >,$)
1711 FORMAT (8I3)
939  FORMAT(8I3)
      END
```

****ALICE FOUNDATION SUMMARY****

Foundation Title: CHAIN

Date of Construct: 1978

Code Format: Fortran IV

Operating Computer: DEC PDP-10

Size Required: 55K

Number of Subroutines: 5

Foundation/Module/Task Summary:

The CHAIN foundation accepts ALICE generated polygon data on 9 track tape. CHAIN converts the polygons to a chain and node format and outputs the data to 7 track magnetic tape. The chain and node data is then windowed to fit within boundaries formed by the map markers.

Type of Input Data Required by Foundation:

ALICE generated polygons on 9 track tape

Type of Output Data Required by Foundation:

Chain and node data on 7 track tape

References:

Section 4.0 of this report, Volume I; Table 3 of this report, Volume II

```

C      MAIN      MAIN      MAIN      MAIN      MAIN      MAIN      MAIN      MAIN
C
C      THIS PROGRAM DISPLAYS SOIL TYPES
C
      IMPLICIT INTEGER (A-Y)
      COMMON/MS/MS
      COMMON/MT7F/ MT7F
      COMMON/MT9F/MT9F
      COMMON/QSID/QSID
      COMMON/MN/MN,MD,POS
      COMMON/MXY/MINX,MINY,MAXX,MAXY,RA,RB,RO,BP
      COMMON/BDC/BDC(4)
      COMMON/DAE/POI,NR,SOI,QERR,HTOQ
      COMMON/XB/XBUF(2200),YDUF(2200),XBS(50)
      COMMON/DH/DHDI(9000),DIHE(9000)
      COMMON/SEF/SQB(550),FRT(100),FRS(100),PRSS(100)
      COMMON/EF/CE(6000),CB(6000)
      DIMENSION PERSV(2200),SLTP(200),STN(200),FRTS(100)
      DIMENSION GST(5),QID(3),STIT(12),STL(26),STDL(26),BUF(15)
      EQUIVALENCE (PERSV,DHDI),
      1(SLTP,DHDI(2401)),(STN,DHDI(2601)),
      2(FRTS,DHDI(2901)),(BUF,DHDI(3001)),(QID,DHDI(3016))
      N=103
      BQQ=_4000000000
C
      CALL SWTCH(34,134)
      IF(134.EQ.2)
      1CALL MT7OPEN
      CALL MT9OPEN
      S=1
      MS=18
      MDP=330
      MDL=330
      NR=1
      NTD=1
      QCR=1
      TYPE 1100
1100 FORMAT(>>SEARCH FOR EOF ON 7TK? (Y OR N) >,>)
1102 CALL RCHAR(PEF)
      IF(PEF.EQ.0) GO TO 1102
      IF(PEF.EQ.78) GO TO 1
      IF(PEF.EQ.89) GO TO 1104
      TYPE 1705
      GO TO 1102
1104 CONTINUE
      CALL TCR
      TYPE 1106
1106 FORMAT(>>LIST QS ON 7TK? (Y OR N) >,>)
      LLST=0
1108 CALL RCHAR(LQSS)
      IF(LQSS.EQ.0) GO TO 1108
      IF(LQSS.EQ.78) GO TO 1120
      IF(LQSS.EQ.89) LLST=1
      IF(LQSS.EQ.89) GO TO 1120
      TYPE 1705
      GO TO 1108
1120 CONTINUE
      QDT=0
1219 CONTINUE
      CALL MT7R(PERSV,550)

```



```

IF(MT7F.EQ.3) GO TO 1134
IF(LLST.EQ.0) GO TO 1219
DO 1220 I=3,550
QBT=QBT+1
1220 QB(QBT)=PERSV(I)
IF(PERSV(1).EQ.1) GO TO 1219
CALL UMFA
XIDN=ILEFT(QID(1))
YIDN=IRIGHT(QID(1))
PRINT 989,XIDN,YIDN,QID(2),QID(3)
GO TO 1129
1134 CONTINUE
CALL MT7SB
CALL TCR
TYPE 989,XIDN,YIDN
1715 FORMAT(>+LAST QS ON 7TK->,$)
TYPE 1713,QID(1),QID(2),QID(3)
1713 FORMAT(1H+,3A5)
DO 1136 I=1,6000
QB(I)=0
1136 CONTINUE
MT7F=1
CALL TCR
TYPE 854
1 CONTINUE
MN=0
MO=0
POS=0
CALL SSHTCH(24,I24)
CALL QSTOP
CALL QEMPTY
CALL SSHTCH(31,I31)
IF(I31.EQ.1) GO TO 101
TYPE 987
987 FORMAT( > TYPE LAST QSECT NUMBER >)
100 TYPE 1714
1714 FORMAT( > IDNX= >$)
CALL RDEC(QQ,IDNX)
TYPE 1814
1814 FORMAT(> IDNY=>$)
CALL RDEC(QQ,IDNY)
IDEN=ICOMB(IDNX,IDNY)
TYPE 1700,IDEN,IDNX,IDNY
1700 FORMAT(8I8)
TYPE 2701
2701 FORMAT( > IF OK TYPE A NOT TYPE Z >)
1716 CALL RCHAR(D)
IF(D.EQ.65) GO TO 1717
IF(D.EQ.90)GO TO 101
GO TO 1716
1717 CALL MT9R(SQB,550)
PRINT 989,SQB(1)
IF(SQB(1) .NE.IDEN) GO TO 1717
IF(SQB(2).NE.3) GO TO 1717
101 CONTINUE
CALL SSHTCH(30,I30)
IF(I30.EQ.1) GO TO 1101
MEND=0
2102 CONTINUE
CALL MT7R(PERSV,550)

```



```

IF(YBUF(1).EQ.0)YBUF(1)=1
XBUF(1)=IRIGHT(PERSV(5))
N=1
DO 501 I=6,PERSV(4)
N=N+1
XBUF(N)=ILEFT(PERSV(I))
IF(XBUF(N).EQ.0)XBUF(N)=1
N=N+1
XBUF(N)=IRIGHT(PERSV(I))
IF(YBUF(N).EQ.0)XBUF(N)=1
501 CONTINUE
PMAX=0
DO 500 I=2,N-1
MLP=XBUF(I)*XBUF(I-1)
IF(MLP.LT.0) GO TO 503
IF(XBUF(I).LT.0)YBUF(I)=YBUF(I-1)-1
IF(XBUF(I).GT.0)YBUF(I)=YBUF(I-1)+1
GO TO 502
503 YBUF(I)=YBUF(I-1)
502 CONTINUE
IF(YBUF(I).GT.PMAX)PMAX=YBUF(I)
IF(YBUF(I).EQ.PMAX)PM=I
500 CONTINUE
P=1
PLK=0
508 R=1
XBS(1)=XBUF(P)
DO 504 I=P+1,N-1
IF(XBUF(I).EQ.0) GO TO 504
IF(YBUF(I).NE.YBUF(P)) GO TO 504
R=R+1
IF(R.GT.50)PAUSE > XBS OVERFLOW >
XBS(R)=IABS(XBUF(I))
XBUF(I)=0
504 CONTINUE
506 MAZ=0
DO 505 I=2,R
IF(XBS(I).GE.XBS(I-1)) GO TO 505
MAZ=1
CSV=XBS(I-1)
XBS(I-1)=XBS(I)
XBS(I)=CSV
505 CONTINUE
IF(MAZ.EQ.1) GO TO 506
PLK=PLK+1
IF(PLK.LT.2) GO TO 524
DO 507 I=2,R,2
CALL QNOP(XBS(I-1)*MG/10,YBUF(P)*MG/10)
CALL QVECTOR(XBS(I)*MG/10,YBUF(P)*MG/10)
507 CONTINUE
PLK=0
524 CONTINUE
XBUF(P)=0
509 P=P+1
IF((XBUF(P).EQ.0).AND.(P.LT.N)) GO TO 509
IF(P.LE.FN) GO TO 503
522 CONTINUE
IF(SQB(K).EQ.BSQ) GO TO 2
PERSV(3)=SQB(K)-BSQ
BSS=PERSV(3)

```

```

M=4
GO TO 103
108 CONTINUE
IF(I20.EQ.1) GO TO 523
CALL QSTART
510 CALL RCHAR(D)
IF(D.EQ.0) GO TO 510
CALL QSTOP
523 CONTINUE
DO 107 K=1,550
107 PERSV(K)=SQB(K)
IF(PERSV(2).EQ.2) GO TO 20
PST=0
L=1
DO 1107 K=1,PERSV(4)
FRSS(K)=PERSV(K)
IF(K.LT.5) GO TO 1107
L=L+1
SQB(L)=PERSV(K)
CALL IRLEFT(PERSV(K),PST)
1107 CONTINUE
SQB(1)=L
GO TO 30
C
20 DO 22 I=1,PERSV(4)
FRS(I)=PERSV(I)
22 CONTINUE
MINY=1
MINX=1
MAXX=ILEFT(PERSV(3))
MAXY=IRIGHT(PERSV(3))
GO TO 2
C
C THE SOIL TYPE IS PLACED IN THE RIGHT HALF OF PRT
C
30 CONTINUE
DO 32 I=1,4
PRT(I)=PERSV(I)
32 CONTINUE
IDB=0
DO 36 I=5,PERSV(4)
PRP=IRIGHT(PERSV(I))
PST=ILEFT(PERSV(I))
CALL IRRIGHT(PRT(I),PRP)
CALL IRLEFT(FRS(I),PST)
IF(PST.NE.0)IDB=78
36 CONTINUE
EST=PERSV(4)-4
CELY=IRIGHT(PERSV(1))
CELX=ILEFT(PERSV(1))
TYPE 939,CELX,CELY
GG=PERSV(4)+1
HH=PERSV(4)+4
J=0
DO 336 I=GG,HH
J=J+1
EPC(J)=PERSV(I)
336 CONTINUE
C
37 CONTINUE

```

```

CALL SSWTCH (26,I26)
IF (I26.EQ.1) GO TO 55
DO 106 I=1,550
FRP=ILEFT(Q3(I))
PRR=IRIGHT(Q3(I))
106 PRINT 989,FRP,PRR,I
DO 54 I=5,FRT(4)
FRP=ILEFT(FRT(I))
PRR=IRIGHT(FRT(I))
PRINT 1711, FRP,PRR
54 CONTINUE
55 CONTINUE
7006 CONTINUE
K=1
DO 4106 I=5,FRT(4)
K=K+1
KBT=ILEFT(SQB(K))
KBU=IRIGHT(FRT(I))
SQB(K)=ICOMB(KBT,KBU)
4106 CONTINUE
CALL DIN
1800 CONTINUE
TYPE 1801
1801 FORMAT( > READ IN NEXT QSECT , Y OR N >)
1802 CALL RCHAR(D)
IF(D.EQ.0) GO TO 1802
IF(D.EQ.89) GO TO 1
IF(D.EQ.78 ) GO TO 120
TYPE 1705
GO TO 1802
120 CONTINUE
IF(I34.EQ.1)CALL EXIT
CALL MT7EOF
CALL MT7EOF
CALL QSTOP
CALL MT7CLOSE
CALL HTSCLOSE
CALL EXIT
854 FORMAT (>+7TK READY>)
1701 FORMAT (>+SOIL TYPE ERROR>)
1702 FORMAT (>+ *>,$)
1703 FORMAT(>+ >>,$)
1704 FORMAT(>+WRITE 7TK? (Y OR N) >,$)
1705 FORMAT(>+? >,$)
1711 FORMAT (8I8)
1712 FORMAT(1X,3A5,3I8)
989 FORMAT(8I8)
END
SUBROUTINE DIN
IMPLICIT INTEGER (A-Y)
COMMON/GG/GG
COMMON/TCC/TCC(3)
COMMON/HH/HH
COMMON/MG/MG
COMMON/MN/MN,MO,POS
COMMON/MXY/MINX,MINY,MAXX,MAXY,RA,RB,RD,BP
COMMON/EDC/BDC(4)
COMMON/CR/CRT(550),CRX(550),CRP(550)
COMMON/SBF/SQB(550),FRT(100),PRR(100),PRSS(100)
COMMON/DH/DIMD(9000),DINE(9000)

```

```

COMMON/BF/QB(6000),OB(6000)
COMMON/YB/XBUF(2200),YBUF(2200),XBS(50)
DIMENSION MAX(628),MAY(628),TAL(628),TAR(628),TAM(628),TAI(628)
1,TAQ(628)
DIMENSION GRA(4),GRT(10)
DIMENSION FIMO(6000),FIME(6000)
EQUIVALENCE (FIMO,QB),(FIME,OB)
EQUIVALENCE (MAX,XBUF),(MAY,XBUF(629)),(TAL,XBUF(1257)),
1(TAR,XBUF(1835)),(TAM,YBUF(313)),(TAI,YBUF(941)),(TAQ,YBUF(1569))
C
C   GENERATE NODE TABLE WITH POTENTIAL NODES
C   INTERNAL NODES
C
WRT=0
MAP=0
150 CONTINUE
DNO=IABS(DIMO(1))
XA=ILEFT(DNO)
YB=IRIGHT(DNO)
TB=IRIGHT(DIME(1))
TA=ILEFT(DIME(1))
H=1
MAX(H)=XA
MAY(H)=YB
TAL(H)=TA
TAR(H)=TB
TAM(H)=0
TAI(H)=1
TAQ(H)=DIME(1)
DO 236 I=2,MN
IF(DIMO(I).GT.0) GO TO 236
TA=ILEFT(DIME(I-1))
TB=IRIGHT(DIME(I-1))
XA=ILEFT(DIMO(I-1))
YB=IRIGHT(DIMO(I-1))
H=H+1
TAR(H)=TB
TAL(H)=TA
MAY(H)=YB
MAX(H)=XA
TAI(H)=0
TAI(H)=I-1
TAQ(H)=DIME(I-1)
TB=IRIGHT(DIME(I))
TA=ILEFT(DIME(I))
DNO=IABS(DIMO(I))
XA=ILEFT(DNO)
YB=IRIGHT(DNO)
H=H+1
TAR(H)=TB
TAL(H)=TA
MAY(H)=YB
MAX(H)=XA
TAM(H)=0
TAI(H)=I
TAQ(H)=DIME(I)
236 CONTINUE
YB=IRIGHT(DIMO(MN))
TA=ILEFT(DIME(MN))
TB=IRIGHT(DIME(MN))

```

```

XA=ILEFT(DIMO(MN))
H=H+1
TAR(H)=TB
TAL(H)=TA
MAY(H)=YB
MAX(H)=XA
TAM(H)=0
TAI(H)=IN
TAG(H)=DIME(MN)
HH=H
G=0
TYPE 989,HH
IF(HH.GT.628) PAUSE > HH GT 628>
C   FIND AND TAG ALL NODES
C
TAR(HH+1)=0
TAL(HH+1)=0
TAM(HH+1)=0
I=0
KG=1
CALL SSWTCH(9,I9)
126 CONTINUE
I=I+1
IF(I.GE.HH) GO TO 128
IF(TAM(I).NE.0) GO TO 126
IF(TAR(I).NE.TAR(I+1)) GO TO 129
IF(TAL(I).EQ.TAL(I+1)) GO TO 126
IF(TAM(I+1).NE.0) GO TO 126
KE=I+1
GO TO 130
129 CONTINUE
KE=KG
KG=I+1
IF(TAM(KE).NE.0) GO TO 126
130 CONTINUE
G=G+1
TAG=TAL(KE)
TAH=TAR(I)
TAJ=TAL(I)
TCC(1)=0
TCC(2)=TAJ
TCC(3)=TAH
CALL THREEPAC(TCC,TCD)
IF((TAM(I).NE.0).OR.(TAM(KE).NE.0))GO TO 131
TAM(I)=G
TAM(KE)=G
II=I
KK=KG
WARD=1
123 CONTINUE
WARD=WARD+1
IF(WARD.GT.5) GO TO 131
MARK=0
WARD=0
DO 124 K=2,HH+1
IF((WARD.EQ.0).AND.(TAM(K).NE.0)) GO TO 124
IF((MARK.EQ.0).AND.(TAR(K).NE.TAG))GO TO 124
IF((MARK.EQ.1).AND.(TAR(K).NE.TAG)) GO TO 125
IF(MARK.EQ.0)KT=K
MARK=1

```

```

IF(TAL(K).NE.TAH) GO TO 124
MARD=1
IF(TAL(K).EQ.TAL(K+1)) GO TO 124
IF(TAR(K).NE.TAR(K+1)) GO TO 124
IF((TAM(K).NE.0).OR.(TAM(K+1).NE.0))GO TO 131
TAM(K)=G
TAM(K+1)=G
TAG=TAL(K+1)
TAH=TAR(K)
TAJ=TAL(K)
TCC(1)=0
TCC(2)=TAH
TCC(3)=TAG
CALL THREEPAC(TCC,TCE)
II=K
KK=K+1
IF(TCE.EQ.TCD) GO TO 126
GO TO 123
125 CONTINUE
IF(MARD.EQ.0) GO TO 131
IF(TAM(KT).NE.0) GO TO 131
IF((TAM(KT).NE.0).OR.(TAM(K-1).NE.0))GO TO 131
TAM(K-1)=G
TAM(KT)=G
TAH=TAR(KT)
TAG=TAL(KT)
TAJ=TAL(K-1)
TCC(1)=0
TCC(2)=TAH
TCC(3)=TAG
CALL THREEPAC(TCC,TCE)
II=K-1
KK=KT
IF(TCE.EQ.TCD) GO TO 126
GO TO 123
124 CONTINUE
131 CONTINUE
DO 127 E=1,HH
127 IF(TAM(E).EQ.G)TAM(E)=0
G=G-1
GO TO 126
128 CONTINUE
3341 CONTINUE
GG=G
G=1
CALL SSHTCH(3,I3)
IF(I3.EQ.1) GO TO 5341
DO 4341 I=1,HH
4341 PRINT 989,MAX(I),MAY(I),TAL(I),TAR(I),TAQ(I),TAM(I),I
1,CRT(I),CRP(I)
5341 CONTINUE
153 CONTINUE
K=0
DO 6154 I=1,HH
IF(TAM(I).NE.G) GO TO 6154
IF(TAL(I).LT.TAR(I)) GO TO 6154
K=K+1
CT=TAL(I)
CR=TAR(I)
GRA(1)=ILEFT(PRSS(CT+4))

```



```

GRA(2)=IRIGHT(PRSS(CR+4))
GRA(3)=ILEFT(PRSS(CR+4))
GRA(4)=IRIGHT(PRSS(CR+4))
CALL FOURPAC(GRA,GR(K))
K=K+1
GR=GRA(1)
GRA(1)=GRA(3)
GRA(3)=GR
GRA(2)=GRA(4)
GRA(4)=GR
CALL FOURPAC(GRA,GR(K))
6154 CONTINUE
KK=K
DO 6155 I=1,KK-1
DO 6155 K=I+1,KK
IF(GR(K).NE.GR(I)) GO TO 6156
DO 6158 L=1,HH
6153 IF(TAM(L).EQ.G)TAM(L)=0
GO TO 5157
6156 CONTINUE
6155 CONTINUE
YSUM=0
XSUM=0
H=0
DO 154 I=1,HH
IF(TAM(I).NE.G) GO TO 154
H=H+1
XSUM=XSUM+MAX(I)
YSUM=YSUM+MAX(I)
154 CONTINUE
XSUM=XSUM/H
YSUM=YSUM/H
DO 5155 I=1,HH
IF(TAM(I).NE.G) GO TO 5155
XDIF=XSUM-MAX(I)
XDIF=I*ABS(XDIF)
YDIF=YSUM-MAX(I)
YDIF=I*ABS(YDIF)
DIF=XDIF+YDIF
IF(DIF.LT.25) GO TO 5155
TYPE 937
937 FORMAT(> NODE LOGIC WRONG)
DO 5156 K=1,HH
IF(TAM(K).EQ.G)TAM(K)=0
5156 CONTINUE
GO TO 5157
5155 CONTINUE
DO 155 I=1,HH
IF(TAM(I).NE.G) GO TO 155
MAX(I)=XSUM
MAX(I)=YSUM
TAM(I)=-TAM(I)
155 CONTINUE
5157 CONTINUE
G=G+1
IF(G.LE.GG) GO TO 153

```

```

C
C   ELIMINATE ABAB COMBINATIONS
C

```

```

CALL SSHTCH(1,11)
IF(I1.EQ.2) GO TO 1200
I=2
MODK=0
343 CONTINUE
I=I+1
IF(I.GT.HH-3) GO TO 344
IF(TAM(I).NE.0) GO TO 343
IF(TAM(I+1).NE.0) GO TO 343
IF(TAM(I+2).NE.0) GO TO 343
IF(TAM(I+3).NE.0) GO TO 343
IF(TAQ(I).NE.TAQ(I+3)) GO TO 343
IF(TAQ(I+1).NE.TAQ(I+2)) GO TO 343
IF(TAI(I+3)-TAI(I+1).NE.2) GO TO 343
MODK=1
TAQ(I+1)=0
TAQ(I+2)=0
DIMD(TAI(I+1))=0
DIMD(TAI(I+2))=0
I=I+3
GO TO 343
344 CONTINUE
IF(MDK.EQ.0) GO TO 1200
TYPE 932
932 FORMAT(' ABAB CONV>')
K=0
DO 345 I=1,HH
IF(TAQ(I).EQ.0) GO TO 345
K=K+1
TAK(K)=MAX(I)
MAK(K)=MAY(I)
TAL(K)=TAL(I)
TAR(K)=TAR(I)
TAM(K)=TAM(I)
TAQ(K)=TAQ(I)
345 CONTINUE
M=K
K=0
DO 346 I=1,MN
IF(DIMD(I).EQ.0) GO TO 346
K=K+1
DIMD(K)=DIMD(I)
DIME(K)=DIME(I)
346 CONTINUE
M=K
K=1
DO 3346 I=2,MN
IF(DIMD(I).GT.0) GO TO 3346
K=K+1
TAIK(I)=I-1
K=K+1
TAIK(I)=I
3346 CONTINUE
M=K
TAM(K)=M
TYPE 939,K,HH
IF(I9.EQ.1) GO TO 1200
DO 939 I=1,HH
939 PRINT 939,MAK(I),MAY(I),TAL(I),TAR(I),TAM(I),I
9399 CONTINUE

```

```

C
C   ELIMINATE DOUBLE CHAINS
C
  FIMO(1)=DIMO(1)
  FIME(1)=DIME(1)
  K=1
  DO 242 I=2,MN
    TA=ILEFT(DIME(I))
    TB=IRIGHT(DIME(I))
    IF(TA.LT.TB) GO TO 242
    K=K+1
    FIMO(K)=DIMO(I)
    FIME(K)=DIME(I)
242 CONTINUE
    MM=K
    K=0
    DO 243 I=1,HH
      IF(TAM(I).GE.100) GO TO 243
      IF(TAL(I).LT.TAR(I)) GO TO 243
      K=K+1
      MAX(K)=MAX(I)
      MAY(K)=MAY(I)
      TAL(K)=TAL(I)
      TAR(K)=TAR(I)
      TAM(K)=TAM(I)
      TAQ(K)=TAQ(I)
243 CONTINUE
      MM=K
      TAI(1)=1
      L=1
      DO 2242 I=2,MN
        IF(FIMO(I).GT.0) GO TO 2242
        L=L+1
        TAI(L)=I-1
        L=L+1
        TAI(L)=I
        IF(FIME(I).NE.TAQ(L)) GO TO 1243
        GO TO 2242
1243 CONTINUE
        DO 1244 K=MM,L-1,-1
          MAX(K+2)=MAX(K)
          MAY(K+2)=MAY(K)
          TAL(K+2)=TAL(K)
          TAR(K+2)=TAR(K)
          TAQ(K+2)=TAQ(K)
1244 TAM(K+2)=TAM(K)
          MAX(L-1)=ILEFT(FIMO(I-1))
          MAY(L-1)=IRIGHT(FIMO(I-1))
          TAL(L-1)=ILEFT(FIME(I-1))
          TAR(L-1)=IRIGHT(FIME(I-1))
          TAQ(L-1)=FIME(I-1)
          TAM(L-1)=0
          FMO=IABS(FIMO(I))
          MAX(L)=ILEFT(FMO)
          MAY(L)=IRIGHT(FMO)
          TAL(L)=ILEFT(FIME(I))
          TAR(L)=IRIGHT(FIME(I))
          TAQ(L)=FIME(I)
          TAM(L)=0
          MM=MM+2

```

```

2242 CONTINUE
      L=L+1
      TAI(L)=MN
      K=HH
      IF(L.NE.K)TYPE 969,L,HH
969  FORMAT(> L,NE,HH>,8I10)
      IF(L.EQ.K) GO TO 2245
      DO 9 I=1,HH
9     PRINT 989,MAY(I),MAY(I),TAL(I),TAR(I),TAQ(I),TAM(I),TAI(I),I
      DO 8 I=1,NN
      FMO=I*DS(FMO(I))
      XA=ILEFT(FMO)
      YA=IRIGHT(FMO)
      TA=ILEFT(FIME(I))
      TB=IRIGHT(FIME(I))
8     PRINT 989,XA,YA,TA,TB,FMO(I),I
2245 CONTINUE
C
C     TIE ISLANDS AND TWO POINT NODES
C
      MAM=0
4128 CONTINUE
      DO 4133 I=1,HH-1
      IF(TAM(I).NE.0) GO TO 4133
      CT=TAL(I)
      CR=TAR(I)
      GRA(1)=ILEFT(PRSS(CT+4))
      GRA(2)=IRIGHT(PRSS(CT+4))
      GRA(3)=ILEFT(PRSS(CR+4))
      GRA(4)=IRIGHT(PRSS(CR+4))
      CALL FOURPAC(GRA,GRT(1))
      GR=GRA(1)
      GRA(1)=GRA(3)
      GRA(3)=GR
      GR=GRA(2)
      GRA(2)=GRA(4)
      GRA(4)=GR
      CALL FOURPAC(GRA,GRT(2))
      BIG=25
      DO 4132 J=I+1,HH
      IF(TAM(J).NE.0) GO TO 4132
      IF((MAM.EQ.0).AND.(TAQ(J).EQ.TAQ(I))) GO TO 4132
      CT=TAL(J)
      CR=TAR(J)
      GRA(1)=ILEFT(PRSS(CT+4))
      GRA(2)=IRIGHT(PRSS(CT+4))
      GRA(3)=ILEFT(PRSS(CR+4))
      GRA(4)=IRIGHT(PRSS(CR+4))
      CALL FOURPAC(GRA,GRT(3))
      GR=GRA(1)
      GRA(1)=GRA(3)
      GRA(3)=GR
      GR=GRA(2)
      GRA(2)=GRA(4)
      GRA(4)=GR
      CALL FOURPAC(GRA,GRT(4))
      DO 4131 P=1,3
      DO 4130 Q=P+1,4
      IF(GRT(P).EQ.GRT(Q)) GO TO 4129
4130 CONTINUE

```

```

4131 CONTINUE
GO TO 4132
4129 CONTINUE
MI=MAX(I)-MAX(J)
MI=ABS(MI)
MD=MAX(I)-MAX(J)
MD=ABS(MD)
MC=MI+MD
IF(MC.GT.BIG) GO TO 4132
BIG=MC
JJ=J
II=I
4132 CONTINUE
IF(BIG.GE.25) GO TO 4133
GG=GG+1
TAM(JJ)=GG
TAM(II)=GG
AX=(MAX(JJ)+MAX(II))/2
AY=(MAX(JJ)+MAX(II))/2
MAX(JJ)=AX
MAX(II)=AX
MAY(JJ)=AY
MAY(II)=AY
4133 CONTINUE
IF(MM.EQ.1) GO TO 4127
MM=1
GO TO 4128
4127 CONTINUE
IF(123.EQ.2) GO TO 3342
C
C   BOUNDARY NODES
C
RI=30
DO 342 I=1,MM
IF(TAM(I).NE.0) GO TO 342
MR=MAX(I)
MA=MAX(I)
TXT=0
MYL=MAX(I)-MINX
MYH=MAX(I)+MAX(I)
NYL=MAX(I)-MINY
MYH=MAX(I)+MAX(I)
IF(MXL .LT. RI) GG=GG+1
IF(MXL .LT. RI) TAM(I)=GG
IF(MXH .LT. RI) GG=GG+1
IF(MXH .LT. RI) TAM(I)=GG
IF(MYL .LT. RI) GG=GG+1
IF(MYL .LT. RI) TAM(I)=GG
IF(MYH .LT. RI) GG=GG+1
IF(MYH .LT. RI) TAM(I)=GG
342 CONTINUE
3342 CONTINUE
C   LINE FILTER
C
I=1
133 CONTINUE
I=I+1
IF(I.GE.MN) GO TO 135
IF((FIM(I)).LT.0).OR.(FIM(I+1)).LT.0) GO TO 134
FMD=ABS(FIM(I)-1)

```

```

XA=ILEFT(FMO)
XB=ILEFT(FIMO(I))
XC=ILEFT(FIMO(I+1))
XD=(YA+2*XB+XC)/4
YE=IRIGHT(FMO)
YB=IRIGHT(FIMO(I))
YC=IRIGHT(FIMO(I+1))
YD=(YA+2*YB+YC)/4
FIMO(I)=ICOMB(XD,YD)
GO TO 133
134 CONTINUE
I=I+1
GO TO 133
135 CONTINUE
909 FORMAT(10I10)
CALL DIO
END
SUBROUTINE DIO
IMPLICIT INTEGER (A-Y)
COMMON/GG/GG
COMMON/TCC/TCC(3)
COMMON/HH/HH
COMMON/MS/MS
COMMON/MN/MN,MO,PDS
COMMON/MY/MY,MINX,MINY,MAXX,MAXY,RA,RB,RD,BP
COMMON/ECC/ECC(4)
COMMON/CP/CRT(550),CRX(550),CRP(550)
COMMON/SEF/SEF(550),FRT(100),FRS(100),FRSS(100)
COMMON/DM/DIMD(9000),DIME(9000)
COMMON/BF/QB(6000),OB(6000)
COMMON/XB/XBUF(2200),YBUF(2200),XBS(50)
DIMENSION MAX(628),MAY(628),TAL(628),TAR(628),TAM(628),TAI(628)
1,TAD(600)
DIMENSION DOD(4)
DIMENSION FIMO(6000),FIME(6000)
EQUIVALENCE (FIMO,QB),(FIME,OB)
EQUIVALENCE (MAX,XBUF),(MAY,XBUF(629)),(TAL,XBUF(1257)),
1,TAR,XBUF(1885)),(TAM,YBUF(313)),(TAI,YBUF(941)),(TAQ,YBUF(1569))
CALL SSWTCH(5,15)
N=NN
MIE=0
GG=-GG
GG=GG-1
6359 CONTINUE
CALL SSWTCH(9,19)
IF(19.EQ.1) GO TO 6360
PRINT 989,GG
DO 272 K=1,HH
272 PRINT 909,MAX(K),MAY(K),TAL(K),TAR(K),TAM(K),K,TAI(K)
6360 CONTINUE
CALL QEMPTY
C
C   INSERT NODES BETWEEN CHAINS
C
DIME(1)=ICOMB(TAL(1),TAR(1))
DIMD(1)=-ICOMB(MAX(1),MAY(1))
DIME(2)=FIME(1)
DIMD(2)=IASS(FIMO(1))
I=1
K=2

```

```

L=1
2252 CONTINUE
I=I+1
IF(I.GT.MN) GO TO 2254
IF((IS.EQ.2).AND.(FINE(I).NE.TLR)) GO TO 2252
IF(FIMO(I).LT.0) GO TO 2253
K=K+1
DIME(K)=FINE(I)
DIMO(K)=IABS(FIMO(I))
GO TO 2252
2253 CONTINUE
L=L+1
K=K+1
DIME(K)=ICOMB(TAL(L),TAR(L))
DIMO(K)=ICOMB(MAX(L),MAY(L))
TAI(L)=K
L=L+1
K=K+1
DIME(K)=ICOMB(TAL(L),TAR(L))
DIMO(K)=-ICOMB(MAX(L),MAY(L))
TAI(L)=K
GO TO 2252
2254 CONTINUE
K=K+1
L=L+1
DIME(K)=ICOMB(TAL(L),TAR(L))
DIMO(K)=ICOMB(MAX(L),MAY(L))
TAI(L)=K
MI=K
C
C   PLOT CHAINS WITH INSERTED NODES
C
6459 CONTINUE
I=1
DMO=IABS(DIMO(I))
XA=ILEFT(DMO)
YB=IRIGHT(DMO)
CALL QNOP(XA*MG/10+50,YB*MG/10+50)
DO 260 I=2,MN
DMO=IABS(DIMO(I))
XA=ILEFT(DMO)
YB=IRIGHT(DMO)
IF(DIMO(I).LT.0) GO TO 261
CALL QVECTOR(XA*MG/10+50,YB*MG/10+50)
GO TO 260
261 CALL QNOP(XA*MG/10+50,YB*MG/10+50)
260 CONTINUE
CALL QNOP(MINX*MG/10+50,MINY*MG/10+50)
CALL QVECTOR(MINX*MG/10+50,MAXY*MG/10+50)
CALL QVECTOR(MAXX*MG/10+50,MAXY*MG/10+50)
CALL QVECTOR(MAXX*MG/10+50,MINY*MG/10+50)
CALL QVECTOR(MINX*MG/10+50,MINY*MG/10+50)
C
C   PLOT NODES SA XES
C
CALL QSIZE(2)
DO 250 I=1,HH
IF(TAM(I).EQ.0) GO TO 250
XT=MAX(I)
YT=MAY(I)

```

```

CALL QNOPI(XT*MG/10+45, YT*MG/10+43)
CALL QCHAR(>X>)
250 CONTINUE
TANH(HH+1) = 0
DO 252 I=1, HH+1
IF(I.EQ. HH+1) TYPE 9936
IF(TAN(I).NE.0) GO TO 252
CX=MAX(I)*MG/10+50
CY=MAX(I)*MG/10+50
CALL QLTBI(CX, CY)
CALL QNOPI(0, 0)
CALL QNOPI(0, 0)
CALL QNOPI(0, 0)
CALL QNOPI(0, 0)
CALL QSTART
CALL QREP(4)
YOUN=0
YOUN=0
NCNT=0
251 CALL QRTBI(CX, CY)
CALL QNOPI(CX-32, CY-32)
CALL QVECTOR(CX+32, CY+32)
CALL QNOPI(CX-32, CY+32)
CALL QVECTOR(CX+32, CY-32)
CALL QREP(4)
CALL QCHAR(0)
IF(D.EQ.0) GO TO 251
CALL QSTOP
IF(D.EQ.32) GO TO 253
IF(D.EQ.65) GO TO 253
IF(D.EQ.66) GO TO 253
IF(D.EQ.70) GO TO 180
IF(D.EQ.74) GO TO 253
IF(D.EQ.83) GO TO 830
IF(D.EQ.84) GO TO 270
IF(D.EQ.87) GO TO 803
IF(D.EQ.90) GO TO 2522
CALL QSTART
GO TO 251
180 CONTINUE
CALL QDEC(QQ, QH)
DO 644 Q=1, QH
CALL QFILM
644 CONTINUE
CALL QMOVE
CALL QMOVE
CALL QSTART
GO TO 251
803 CONTINUE
FPINT 989, GG
DO 802 L=1, HH
2802 PRINT 989, MAX(L), MAY(L), TAL(L), TAR(L), TAN(L), TAQ(L), TAI(L), L
CALL QSWTCH(2, I2)
IF(I2.EQ.2) GO TO 805
DO 804 L=1, MH
DMD=IABS(DIND(L))
XA=ILEFT(DMD)
YA=IRIGHT(DMD)
TA=ILEFT(DIME(L))
TB=IRIGHT(DIME(L))

```



```

804 PRINT 989,XA,YA,TA,TB,L,DIHO(L)
      IF(WRT.EQ.1) GO TO 6358
805 CONTINUE
      CALL QSTART
      GO TO 251
830 CONTINUE
      IF(I5.EQ.2) GO TO 908
      TYPE 904
904  FORMAT(> TYPE CHAIN NUMBER>)
      TYPE 906
906  FORMAT(> TL=>%)
      CALL RDEC(QQ,TL)
      TYPE 989,TL
      TYPE 907
907  FORMAT(> TR=>%)
      CALL RDEC(QQ,TR)
      TYPE 989,TR
      TLR=ICOMB(TL,TR)
908  CALL SSNTCH(5,I5)
      GO TO 6359

C
C   MANUAL TOING OF NODES
C
253  CONTINUE
      DAV=(CX-50)*10/MG
      DAV=(CY-50)*10/MG
      SUMD=100
      DO 254 K=1,HH
      IF((D.EQ.65).AND.(TAM(K).EQ.0)) GO TO 254
      IF((D.EQ.66).AND.(TAM(K).EQ.0)) GO TO 254
      IF((D.EQ.32).AND.(TAM(K).NE.0)) GO TO 254
      DIFX=DAU-MAX(K)
      DIFX=IABS(DIFX)
      DIFY=DAV-MAY(K)
      DIFY=IABS(DIFY)
      DIFS=DIFX+DIFY
      IF(DIFS.GT.SUMD) GO TO 254
      SUMD=DIFS
      KI=K
254  CONTINUE
      IF(SUMD.GT.RA)TYPE 981
981  FORMAT(> OUT OF RANGE>)
      IF(SUMD.GT.RA)GO TO 2270
      NCNT=NCNT+1
      IF(D.EQ.65) GO TO 280
      IF(D.EQ.66) GO TO 273
      IF(D.EQ.74) GO TO 274
      XSUM=XSUM+MAX(KI)
      YSUM=YSUM+MAY(KI)
      TAM(KI)=GG
2270 CONTINUE
      CALL QSTART
      KJ=KI
      GO TO 251
270  CONTINUE
      IF(NCNT.EQ.0)TYPE 905
905  FORMAT(> NCNT EQ 0>)
      IF(NCNT.EQ.0)GO TO 6359
      XSUM=XSUM/NCNT
      YSUM=YSUM/NCNT

```

```

DO 271 K=1,HH
IF(TAM(K).NE.GG) GO TO 271
MAX(K)=XSUM
MAY(K)=YSUM
271 CONTINUE
IF(NONT.NE.2) GO TO 5272
J=0
DO 5271 K=1,HH
IF(TAM(K).NE.GG) GO TO 5271
J=J+1
DOD(J)=PRSS(TAL(K)+4)
J=J+1
DOD(J)=PRSS(TAR(K)+4)
5271 CONTINUE
IF((DOD(1).EQ.DOD(3)).AND.(DOD(2).EQ.DOD(4))) GO TO 5272
IF((DOD(1).EQ.DOD(4)).AND.(DOD(2).EQ.DOD(3))) GO TO 5272
TYPE 925,TAL(K),TAR(K)
925 FORMAT(> ATTRIBUTE MISMATCH>,2I5)
5272 CONTINUE
GG=GG-1
NN=NN
GO TO 6359
280 CONTINUE
TCC(NONT)=KI
IF(NONT.EQ.2) GO TO 281
CALL RSTART
GO TO 251
281 CONTINUE
TYPE 929
929 FORMAT(> LEFT POLY N=>*)
CALL RDEC(QQ,LPN)
TYPE 923
923 FORMAT(> RIGHT POLY N=>*)
CALL RDEC(QQ,RPN)
DO 282 B=1,2
HH=HH+1
MAX(HH)=MAX(TCC(B))
MAY(HH)=MAY(TCC(B))
TAL(HH)=LPN
TAR(HH)=RPN
TAQ(HH)=ICOMB(LP,RPN)
TAN(HH)=GG
282 CONTINUE
NN=NN+1
FIND(NN)=-ICOMB(MAX(HH-1),MAY(HH-1))
FIME(NN)=TAQ(HH-1)
NN=NN+1
FIND(NN)=ICOMB(MAX(HH),MAY(HH))
FIME(NN)=TAQ(HH)
GG=GG-1
NN=NN
GO TO 6359
273 CONTINUE
HJ=TAM(KI)
DO 2373 J=1,HH
IF(TAM(J).NE.HJ) GO TO 2373
TAR(J)=0
JP=TAI(J)
JQ=1
IF(DIM0(JP).GE.0)JQ=-1

```

```

MAX(J)=ILEFT(DIMO(JP+JQ))
MAY(J)=IRIGHT(DIMO(JP+JQ))
2373 CONTINUE
IF=NN
GO TO 6359
274 CONTINUE
TYPE 909,KJ,KI
TAM(KJ)=TAM(KI)
MAX(KJ)=MAX(KI)
MAY(KJ)=MAY(KI)
NN=NI
GO TO 6359
252 CONTINUE
2522 CONTINUE
9936 FORMAT(> ALL NODES ARE TIED Z TO CONT>)
H=1
DO 650 I=2,MN
  IF(DIMO(I).GT.0) GO TO 650
  H=H+1
  TAI(H)=I-1
  H=H+1
  TAI(H)=I
650 CONTINUE
H=H+1
TAI(H)=NN
IF(H.NE.NN)TYPE 970,H,HH
970 FORMAT(> FIRST H .NE.HH>,8I10)
6358 CONTINUE
909 FORMAT(10I10)
DO 5353 I=1,MN
  DIMO(I)=IABS(DIMO(I))
5358 CONTINUE
976 FORMAT(> TAG CONFLICT>)
CALL DIP
END
SUBROUTINE DIP
IMPLICIT INTEGER (A-Y)
COMMON/TCC/TCC(3)
COMMON/HH/HH
COMMON/NS/NS
COMMON/NI/MN,MO,POS
COMMON/NIY/MINX,MINY,MAXX,MAXY,RA,RB,RD,BP
COMMON/EDC/BDC(4)
COMMON/CR/CRT(550),CRX(550),CRP(550)
COMMON/SBF/SQB(550),FRT(100),FRS(100),PRSS(100)
COMMON/DH/DIMO(9000),DIME(9000)
COMMON/BF/Q3(6000),CB(6000)
COMMON/XB/XBUF(2200),YBUF(2200),XBS(50)
DIMENSION MAX(628),MAY(628),TAL(628),TAR(628),TAM(628),TAI(628)
1,TAQ(628)
DIMENSION FIMO(6000),FIME(6000)
DIMENSION BDX(4),BDY(4)
DIMENSION PERSV(550)
EQUIVALENCE (PERSV,DIME)
EQUIVALENCE (FIMO,Q3),(FIME,OB)
EQUIVALENCE (MAX,XBUF),(MAY,XBUF(629)),(TAL,XBUF(1257)),
1(TAR,XBUF(1885)),(TAM,YBUF(313)),(TAI,YBUF(941)),(TAQ,YBUF(1569))
C
C   CHOP CHAINS OFF TO FIT BOUNDARY
C   BDC CONTAINS MAP MARKERS

```

```

C
  CALL SSWTCH(18,I18)
  DO 4 I=1,MN
    FIME(I)=1
  4 FIMO(I)=0
    DO 15 I=1,4
      BDX(I)=ILEFT(BDC(I))
    15 BDY(I)=IRIGHT(BDC(I))
  16 CONTINUE
    INVT=0
    DO 17 I=1,3
      IF(BDX(I+1).GE.BDX(I)) GO TO 17
      CCX=BDX(I)
      BDX(I)=BDX(I+1)
      BDX(I+1)=CCX
      INVT=1
    17 CONTINUE
    IF(INVT.EQ.1) GO TO 16
  18 CONTINUE
    INVT=0
    DO 19 I=1,3
      IF(BDY(I+1).GE.BDY(I)) GO TO 19
      CCX=BDY(I)
      BDY(I)=BDY(I+1)
      BDY(I+1)=CCX
      INVT=1
    19 CONTINUE
    IF(INVT.EQ.1) GO TO 18
    CALL QEMPTY
    X1=ILEFT(BDC(1))
    Y1=IRIGHT(BDC(1))
    X2=ILEFT(BDC(2))
    Y2=IRIGHT(BDC(2))
    X3=ILEFT(BDC(3))
    Y3=IRIGHT(BDC(3))
    X4=ILEFT(BDC(4))
    Y4=IRIGHT(BDC(4))
    ZXD=X2-X1
    ZYD=Y2-Y1
    ZM1=ZYD/ZXD
    ZND=X3-X2
    ZYD=Y3-Y2
    ZM2=ZYD/ZXD
    ZXD=X3-X4
    ZYD=Y3-Y4
    ZM3=ZYD/ZXD
    ZXD=X4-X1
    ZND=Y4-Y1
    ZM4=ZYD/ZXD
    DO 20 I=1,MN
      XA=ILEFT(DIMO(I))
      YB=IRIGHT(DIMO(I))
      IF(XA.LT.BDX(1)) GO TO 21
      IF(XA.GT.BDX(4)) GO TO 21
      IF(YB.LT.BDY(1)) GO TO 21
      IF(YB.GT.BDY(4)) GO TO 21
      IF((XA.GT.BDX(2)).AND.(XA.LT.BDX(3))) GO TO 26
      GO TO 27
    26 IF((YB.GT.BDY(2)).AND.(YB.LT.BDY(3))) GO TO 20
    27 CONTINUE

```

```

ZYA=XA-X1
ZYD=YB-Y1
XC=ZYB/ZH1+0.5
XC=XC+X1
YC=ZXA+ZH4+0.5
YC=YC+Y1
ZXA=YA-X4
ZYB=YB-Y4
XD=ZYD/ZH3+0.5
XD=XD+X4
ZXA=YA-X2
ZYB=YB-Y2
YD=ZXA+ZH2+0.5
YD=YD+Y2
IF((XA.LT.XC).OR.(XA.GT.XD)) GO TO 21
IF((YB.LT.YC).OR.(YB.GT.YD)) GO TO 21
GO TO 20
21 CONTINUE
FINE(I)=0
20 CONTINUE
DO 429 I=2,MN-1
IF(DIME(I-1).NE.DIME(I+1)) GO TO 429
FIN=FINE(I-1)+FINE(I+1)
IF((FIN.EQ.0).AND.(FINE(I).EQ.1))FINE(I)=0
IF((FIN.EQ.2).AND.(FINE(I).EQ.0))FINE(I)=1
429 CONTINUE
DO 430 I=2,MN
IF((FINE(I).EQ.0).AND.(FINE(I-1).NE.0))FIMO(I-1)=1
IF((FINE(I).NE.0).AND.(FINE(I-1).EQ.0))FIMO(I)=1
430 CONTINUE
K=0
DO 421 I=1,MN
IF(FINE(I).EQ.0) GO TO 421
K=K+1
DIMO(K)=DIME(I)
DIMS(K)=DIME(I)
FIMO(K)=FIMO(I)
FINE(K)=FINE(I)
421 CONTINUE
MN=K
H=0
DO 422 I=1,MN
IF(DIME(I).EQ.DIME(I-1)) GO TO 422
H=H+1
TAL(H)=ILEFT(DIME(I))
TAR(H)=IRIGHT(DIME(I))
TAN(H)=I
422 CONTINUE
HH=H+1
TAL(HH)=0
TAR(HH)=0
TAN(HH)=MN+1
DO 4423 H=2,HH
LNG=TAN(H)-TAN(H-1)
IF(LNG.NE.1) GO TO 4423
MM=TAN(H-1)+1
DO 4424 I=MN,MM,-1
DIME(I+1)=DIME(I)
DIMO(I+1)=DIMO(I)
FINE(I+1)=FINE(I)

```

```

      FIMO(I+1)=FIMO(I)
4424 CONTINUE
      NN=TAN(H-1)
      XA=ILEFT(DIMO(NN))+1
      YB=IRIGHT(DIMO(NN))+1
      DIMO(NN+1)=ICOMB(XA,YB)
      DIME(NN+1)=DIME(NN)
      FIMO(NN+1)=1
      FIME(NN+1)=1
      FIMO(NN)=0
      NN=NN+1
      DO 4425 J=H,HH
      TAN(J)=TAN(J)+1
4425 CONTINUE
4423 CONTINUE
C
C   TEST FOR OPEN POLYGONS
C
      XA=ILEFT(BDC(1))
      YA=IRIGHT(BDC(1))
      CALL QNOP(XA*MG/10+50,YA*MG/10+50)
      DO 25 I=2,4
      XB=ILEFT(BDC(I))
      YB=IRIGHT(BDC(I))
25  CALL QVECTOR(XB*MG/10+50,YB*MG/10+50)
      CALL QVECTOR(XA*MG/10+50,YA*MG/10+50)
      CALL QREADI(IPAB)
      XA=ILEFT(DIMO(1))
      YB=IRIGHT(DIMO(1))
      CALL QNOP(XA*MG/10+50,YB*MG/10+50)
      DO 23 I=2,MN
      XA=ILEFT(DIMO(I))
      YB=IRIGHT(DIMO(I))
      IF(DIME(I).NE.DIME(I-1)) GO TO 24
      CALL QVECTOR(XA*MG/10+50,YB*MG/10+50)
      GO TO 23
24  CONTINUE
      CALL QNOP(XA*MG/10+50,YB*MG/10+50)
23  CONTINUE
      CALL QREADI(IPAR)
      CALL SRTCH(4,I4)
      PCN=0
      MK=0
3401 CONTINUE
      PCN=PCN+1
      IF(PCN.GT.SQB(1)-1) GO TO 3400
      IF(I4.EQ.2)
      1FFINT 939,PCN
      K=0
      MARG=0
      MARK=0
      CR=0
      DO 3402 I=1,MN+1
      TA=ILEFT(DIME(I))
      TB=IRIGHT(DIME(I))
      IF((TA.EQ.PCN).OR.(TB.EQ.PCN)) GO TO 3404
      IF(MARK.EQ.0) GO TO 3402
      MARK=0
      XA=ILEFT(DIMO(I-1))
      YB=IRIGHT(DIMO(I-1))

```

```

L=L+1
IF(L.LE.BFS) GO TO 785
READ (QS) MUXH
L=1
785 LOAT(I)=MUXH(L)
790 CONTINUE
C          GENERATE FILE NAMES
DO 870 I=1,CLCT
IF(CLPC(I).EQ.0) GO TO 800
CLX=I-(I-1)/MAPX)*MAPX
CLY=(I-1)/MAPX*1
IF(CLX.GT.MAPX) PAUSE > MAPX OVERFLOW>
IF(CLY.GT.MAPY) PAUSE > MAPY OVERFLOW>
CLX1=CLX/10
CLX2=CLX-CLX1*10
CLY1=CLY/10
CLY2=CLY-CLY1*10
INCODE(5,901,CEL(I)) CLX1,CLX2,CLY1,CLY2
800 CONTINUE
901 FORMAT(1HC,4I1)
C          EQUIVALATE POLYGONS
DO 190 I=1,PGCT
EPG=2-ICHT(PGTG(I))
IF (I.EQ.EPG) GO TO 180
IF (EPG.EQ.0) GO TO 190
IF (ILEFT(PGTG(I)).NE.0) GO TO 190
PGT(I)=ICOMB(I,PGT(I))
160 CONTINUE
PGT(EPG)=ICOMB(I,PGT(EPG))
IF (IRIGHT(PGT(EPG)).EQ.I) GO TO 190
EPG=IRIGHT(PGT(EPG))
GO TO 160
180 CONTINUE
IF (ILEFT(PGT(I)).NE.0) GO TO 190
PGT(I)=ICOMB(I,PGT(I))
190 CONTINUE
OPI=0
DO 210 I=1,PGCT
IF (PGT(I).EQ.0) GO TO 210
IF (ILEFT(PGT(I)).NE.I) GO TO 200
OPI=OPI+1
OPG(I)=OPI
GO TO 210
200 OPG(I)=OPG(ILEFT(PGT(I)))
210 CONTINUE
C          SW1 ON LOADS ATTRIBUTES FROM CARDS
C          SW1 OFF LOADS ATTRIBUTES FROM TABLES
CALL SWTCH (1,1)
IF (I1.EQ.1) GO TO 450
READ 401,DHCE,DSEE
DO 404 I=1,DHCE,16
READ 405,(DHC(K),K=I,I*15)
404 CONTINUE
DO 403 I=1,DSEE,16
READ 405,(DSE(K),K=I,I*15)
403 CONTINUE
GO TO 450
401 FORMAT (7I5)
405 FORMAT (16A5)

```



```

150 CBF(K,D)=IABS(MUCH(L))
120 CONTINUE
C      LOAD ALL THE CHAIN POINTERS
FSTPG=ILEFT(CLFG(IN))
PGFF=(FSTPG/4096)*4096
DO 130 J=1,NOCN(D)
L=L+1
IF(L.LE.BFS) GO TO 140
READ(QS) MUCH
L=L-1
140 CALL UPTHREE(MUCH(L),QID)
CP(J,D)=QID(1)
RAI(J,D)=QID(2)+PGFF
IF (RAI(J,D).LT.FSTPG) RAI(J,D)=RAI(J,D)+4096
LAI(J,D)=QID(3)+PGFF
IF (LAI(J,D).LT.FSTPG) LAI(J,D)=LAI(J,D)+4096
130 CONTINUE
CP(NOCN(D)+1,D)=NOFT(D)+1
C      PRINT ATTRIBUTES IF SQ22 ON
CALL SWITCH (22,22)
IF (22.EQ.1) GO TO 180
PRINT 173
DO 170 I=1,NOCN(D)
DL=LEFT(CBF(CPI I,D),D)+XTRAIN(N)
DR=DRIGHT(CBF(CPI I,D),D)+XTRAIN(N)
LI=LEFT(CBF(CPI I+1,D)-1,D)+XTRAIN(N)
RD=DRIGHT(CBF(CPI I+1,D)-1,D)+XTRAIN(N)
PRINT 171,NL,I,LI,DR,RAI I,D,LAI I,D,DL,DR,LI,RD
170 CONTINUE
180 CONTINUE
171 FORMAT ('808')
173 FORMAT ('811',> CELL CHAIN PGL PGR CBF1 STEP Y1
180 XSEP XSEP)
RETURN
END
SUBROUTINE DAVIES
IMPLICIT INTEGER (4-7)
COMMON/CL/CP 1000,1,PH 10,1,CLCH 400,1,CC 1000,1,CHCN 1000)
COMMON/CLF/CLF 400,1,BFS,CL,PGCT,CLCT,N
COMMON/CP/CP 200,2,RAI 200,2,LAI 200,2,NOCN 2,NOFT 2,
TAM 200,2,NOCN 3)
COMMON/PG/PG 700,1,PG 700,1,XTRAI 400,1,XTRAI 400,1,CELL 400)
CD=0
CLC=0
D=1
NR=1
180 CONTINUE
CLCH(N)=CCMB(CLC,NOCN 1)
CLC=CLC+NOCN 1)
DO 170 I=1,NOCN 1)
CD=CD+1
DL=LEFT(CBF(CPI I,D),D)+XTRAIN(N)
DR=DRIGHT(CBF(CPI I,D),D)+XTRAIN(N)
CPI(CD)=CCMB(DL,DR)
CC(CD)=CD
DADD(CD)=CCMB(N,2)
CD=CD+1
DL=LEFT(CBF(CPI I+1,D)-1,D)+XTRAIN(N)
DR=DRIGHT(CBF(CPI I+1,D)-1,D)+XTRAIN(N)
CPI(CD)=CCMB(DL,DR)

```

```

      CC(CI)=CI
      CHID(CI)=ICOM(SN ,I)
120 CONTINUE
      CALL SWITCH (28,128)
      IF (128.EQ.1) GO TO 124
      IL=LEFT(CLCH(N))
      IR=RIGHT(CLCH(N))
      PRINT 123,N ,CI,IL,IR
124 I=I+1
C
      ORDER THE END POINT COORDINATES
      DO 160 J=1,CI-1
      IF (CEPI(J+1).GE.CEPI(J)) GO TO 160
      GG=CEPI(J+1)
      HH=CC(J+1)
      FF=CHID(J+1)
      I=J+1
152 IF (I.EQ.1) GO TO 156
      IF (CEPI(I-1).LE.GG) GO TO 156
      CEPI(I)=CEPI(I-1)
      CC(I)=CC(I-1)
      CHID(I)=CHID(I-1)
      I=I-1
      GO TO 152
156 CEPI(I)=GG
      CC(I)=HH
      CHID(I)=FF
160 CONTINUE
      DO 166 I=1,CI
166 CC(CCI(I))=DECOMBZ,CC(CCI(I))
      DO 168 I=1,CI
168 CC(I)=DLEFT(CCI(I))
C
      PRINT END POINTS IF SWITCH 24 ON
      CALL SWITCH (24,224)
      IF (224.EQ.1) GO TO 140
      PRINT 171)
      DO 134 K=1,CI,8
      I=1
      DO 130 J=1,7
      FB(I)=DLEFT(CEPI(J+K))
      I=I+1
      FB(I)=DRIGHT(CEPI(J+K))
      I=I+1
130 I=I+1
      PRINT 123,(FB(I),I=1,16)
134 CONTINUE
123 FORMAT (8I20,20X)
140 CONTINUE
C
      SWITCH 25 ON PRINTS NODES (X1,Y1 X2,Y2 ...)
      CALL SWITCH (25,225)
      IF (225.EQ.1) GO TO 174
      PRINT 173
      I=0
      DO 190 J=1,CI-1
      IF (CEPI(J).NE.CEPI(J+1)) GO TO 182
      IF (I.NE.0) GO TO 174
      I=I+1
      FB(I)=DLEFT(CEPI(J))
      I=I+1
      FB(I)=DRIGHT(CEPI(J))
      I=I+1
174 I=I+1
      FB(I)=DLEFT(CEPI(J+1))

```

```

      I=I+1
      PB(I)=IRIGHT(CEP(J+1))
      GO TO 190
182 IF (I.EQ.0) GO TO 186
      PRINT 185,(PB(K),K=1,I)
      I=0
      GO TO 190
186 I=I+1
      PB(I)=ILEFT(CEP(J))
      I=I+1
      PD(I)=IRIGHT(CEP(J))
      PRINT 185,(PB(K),K=1,I)
      I=0
190 CONTINUE
194 CONTINUE
C      SSWTCH 26 ON PRINTS CHID--CELL NO., CHAIN NO.
      CALL SSWTCH (26,I26)
      IF (I26.EQ.1) GO TO 220
      PRINT 173
      DO 214 K=1,CJ,8
      I=1
      DO 210 J=0,7
      FB(I)=ILEFT(CHID(J+K))
      I=I+1
      FB(I)=IRIGHT(CHID(J+K))
210 I=I+1
      PRINT 123,(PB(L),L=1,16)
214 CONTINUE
220 CONTINUE
C      SSWTCH 27 ON PRINTS CC--CEP INDEX
      CALL SSWTCH (27,I27)
      IF (I27.EQ.1) GO TO 230
      PRINT 173
      PRINT 123,(CC(L),L=1,CI)
230 CONTINUE
173 FORMAT (1H1)
185 FORMAT (8(2X,216))
      RETURN
      END
      SUBROUTINE LNKCH (I)
      IMPLICIT INTEGER(A-Y)
      COMMON/CE/CEP(1000),PB(16),CLCH(480),CC(1000),CHID(1000)
      COMMON/CLPG/CLPG(480),BFS,QS,PGCT,CLCT,N
      COMMON/CP/CP(200,2),RA(200,2),LA(200,2),NOCH(2),NOPT(2),
      TCBF(2000,2),QID(3)
      COMMON/PCB/PCB(2000)
      COMMON/FG/FGTB(7000),FGTG(7000),XTRA(480),YTRA(480),CEL(480)
      COMMON/CHP/CHP(2),CEID(2),CI,130
      COMMON/OP/OPG(7000),LWY(500),OPI
      COMMON/SSF/SSF(126),SQB(550)
      COMMON/CRS/CLNH,CR(4),OCLI,OCUI
C      CEP(I)=ORDERED (X,Y) CHAIN END POINTS
C      CHID(I)=IDENTITY (CELL,CHAIN) OF END POINT (I)
C      CC(I)=POSITION OF CHAIN END POINT IN CEP
C      FIND END POINTS OF WORKING CHAIN
      CCI=2*(ILEFT(CLCH(N))+1)
      CHEP(1)=CCI-1
      CHEP(2)=CCI
C      CHECK TO SEE IF CHAIN ALREADY DONE
      IF (CC(CHEP(1)).LT.0) GO TO 500

```

```

C                                LOAD FIRST SEGMENT
  L=1000
  OFS=ICOMB(XTRA(N),YTRA(N))
  DO 152 M=CP(I,1),CP(I+1,1)-1
  L=L+1
  PCB(L)=CBF(M,1)+OFS
152 CONTINUE
153 FORMAT (10X,I6,6X,0I2)
  PLI=1001
  PUI=L
  CC(CHEP(1))=-CC(CHEP(1))
  CC(CHEP(2))=-CC(CHEP(2))
C                                CHECK NODE COMPLEXITY
  E=1
162 CXYI=IABS(CC(CHEP(E)))
166 WXYI=CXYI
  CXYC=1
  CXY=CEP(CXYI)
170 CXYI=CXYI-1
  IF (CEP(CXYI).NE.CXY) GO TO 174
  CXYC=CXYC+1
  MXYI=CXYI
  GO TO 170
174 CXYI=WXYI
178 CXYI=CXYI+1
  IF (CEP(CXYI).NE.CXY) GO TO 182
  CXYC=CXYC+1
  MXYI=CXYI
  GO TO 178
182 CONTINUE
183 FORMAT (2X,4I4)
  IF (CXYC.EQ.1) GO TO 300
  IF (CXYC.EQ.2) GO TO 220
C                                END NODE
186 E=E+1
  IF (E.GT.2) GO TO 400
  GO TO 162
C                                TWO CHAIN NODE
220 CONTINUE
  RCL=ILEFT(CHID(MXYI))
  RCH=IRIGHT(CHID(MXYI))
  CCCI=2*(ILEFT(CLCH(RCL))+RCH)
  IF (CC(CCCI).LT.0) GO TO 186
  IF (RCL.EQ.CEID(1)) GO TO 232
  IF (RCL.EQ.CEID(2)) GO TO 236
  PRINT 221,CEID(1),RCL
221 FORMAT (2I8)
  F=1
  GO TO 250
232 F=1
  GO TO 250
236 F=2
C                                IF E=1--PCB COUNTS DOWN,LI=-1
C                                IF E=2--PCB COUNTS UP,LI=1
C                                IF CEP=CBF(S)--CBF COUNTS UP,MI=1
C                                IF CEP=CBF(E)--CBF COUNTS DOWN,NI=-1
250 LI=-1
  IF (E.EQ.2) LI=1
  L=PLI
  IF (E.EQ.2) L=PUI

```



```

ITMSZ=PL-PS+1
SFN=SFN+1
C
DO 6 K=1,80
6 EB(K)=32
C
CALL UPFIVE (HDR,EB(1))
E=8
ENCODE (10,33,TEM) SFN,ITMSZ
DO 10 K=1,2
CALL UPFIVE (TEM(K),EB(E))
10 E=E+6
C
DSET=CNTY
DO 130 K=24,22,-1
EB(K)=MOD(DSET,10)+48
DSET=DSET/10
IF (DSET.EQ.0) GO TO 132
130 CONTINUE
C
132 DSET=TWP
DO 134 K=27,25,-1
EB(K)=MOD(DSET,10)+48
DSET=DSET/10
IF (DSET.EQ.0) GO TO 136
134 CONTINUE
C
136 DSET= RNG
DO 138 K=30,28,-1
EB(K)=MOD(DSET,10)+48
DSET=DSET/10
IF (DSET.EQ.0) GO TO 140
138 CONTINUE
140 CONTINUE
C
EB(36)=QSECT+48
DSET=SECT
DO 142 K=33,32,-1
EB(K)=MOD(DSET,10)+48
DSET=DSET/10
IF (DSET.EQ.0) GO TO 144
142 CONTINUE
144 CONTINUE
E=38
ENCODE (10,33,TEM) OPG(PGL),OPG(PGR)
DO 26 K=1,2
CALL UPFIVE (TEM(K),EB(E))
26 E=E+6
DSES=54
DSET=ILEFT(PGTB(PGL))
C
CALL UPFIVE (DSE(DSET),TEM(1))
DO 20 K=5,1,-1
IF (TEM(K).EQ.32) GO TO 20
EB(DSES)=TEM(K)
DSES=DSES-1
20 CONTINUE
DSES=60
DSET=ILEFT(PGTB(PGR))
CALL UPFIVE (DSE(DSET),TEM(1))

```

HEADER CARD

ENCODE COUNTY

ENCODE TOWNSHIP NORTH

ENCODE TOWNSHIP EAST

ENCODE QSECT, SECT

CHECK FOR A BLANK SPACE

```

DO 24 K=5,1,-1
IF (TEM(K).EQ.32) GO TO 24
EB(DSES)=TEM(K)
DSES=DSES-1
24 CONTINUE
E=1
DO 12 K=1,16
CALL FIVEPAC (EB(E),ACARD(K))
12 E=E+5
CALL CONEBDIC (ACARD,ECARD,16)
CALL BLKN(1)
DO 14 K=1,80
14 EB(K)=32
180 CONTINUE
C
E=3
P=PS
PE=P+2
IF (PL-PS.EQ.1) PE=PS+1
C
30 CONTINUE
DO 38 J=P,PE
ILE =ILEFT(PCB(J))*ZX*100+SPX
IRI=IRIGHT(PCB(J))*ZY*100+SPY
ENCODE (20,37,TEM) ILE,IRI
CALL UPFIVE (TEM(1),EB(E))
E=E+5
CALL UPFIVE (TEM(2),EB(E))
E=E+7
CALL UPFIVE (TEM(3),EB(E))
E=E+5
CALL UPFIVE (TEM(4),EB(E))
E=E+7
38 CONTINUE
C
E=1
DO 42 K=1,16
CALL FIVEPAC (EB(E),ACARD(K))
42 E=E+5
CALL CONEBDIC (ACARD,ECARD,16)
CALL BLKN(1)
C
DO 44 K=1,80
44 EB(K)=32
E=3
P=P+3
PE=P+2
IF (P.GT.PL) GO TO 50
IF (PE.LE.PL) GO TO 30
C
DO 46 K=1,80
46 EB(K)=32
PE=PL
GO TO 30
31 FORMAT (I5)
33 FORMAT (2I5)
35 FORMAT (4I5)
37 FORMAT (2I10)
C
50 CONTINUE

```

ENCODE 3 POINTS PER DATA CARD

PACK, CONVERT AND WRITE

INCREMENT AND TEST

LAST DATA CARD


```

RETURN
200 CONTINUE
DO 250 K=1,80
250 EB(K)=32
EB(5)=45
EB(6)=49
EB(11)=45
EB(12)=50
E=1
DO 254 K=1,16
CALL FIVEPAC (EB(E),ACARD(K))
254 E=E+5
CALL CONEBDIC (ACARD,ECARD,16)
CALL BLKN(1)
RETURN

```

C

OUTPUT AREAS

```

300 CONTINUE
IF (ATBI.NE.0) GO TO 310
SFN=SFN+1
DO 304 K=1,80
304 EB(K)=32
CALL UPFIVE (ADD,EB(1))
ENCODE (5,31,TEM) SFN
CALL UPFIVE (TEM(1),EB(8))
EB(18)=50
E=1
DO 308 K=1,16
CALL FIVEPAC (EB(E),ACARD(K))
308 E=E+5
CALL CONEBDIC (ACARD,ECARD,16)
CALL BLKN(1)
310 DO 312 K=1,80
312 EB(K)=32
EB(6)=49
ENCODE (5,31,TEM) OPG(PGL)
CALL UPFIVE (TEM(1),EB(8))
ENCODE (10,321,TEM) PGR
CALL UPFIVE (TEM(1),EB(21))
CALL UPFIVE (TEM(2),EB(26))
E=1
DO 316 J=1,16
CALL FIVEPAC (EB(E),ACARD(J))
316 E=E+5
CALL CONEBDIC (ACARD,ECARD,16)
CALL BLKN(1)
ATBI=ATBI+1
RETURN
321 FORMAT (I10)
END
SUBROUTINE BLKN(DNE)
IMPLICIT INTEGER (A-Y)
COMMON/NT9F/NT9F
COMMON/AC/ACARD(16),ECARD(20),EB(80),TEM(6),ATB(200)
DIMENSION ECD(200)
DATA ECDI/0/
IF (DNE.EQ.2) GO TO 40
DO 20 I=1,20
ECDI=ECDI+1
ECD(ECDI)=ECARD(I)
20 CONTINUE

```

```

IF (ECDI.LT.200) RETURN
  CALL MT9W (ECD,200)
  ECDI=0
  RETURN
40 CONTINUE
  IF (ECDI.EQ.0) GO TO 60
  CALL MT9W (ECD,ECDI)
60 CONTINUE
  CALL MT9EOF
  CALL MT9EOF
  RETURN
  END
SUBROUTINE SKL (SPC1,SPC2)
  IMPLICIT INTEGER (A-Y)
  COMMON/CLFG/CLFG(480),BFS,QS,PGCT,CLCT,N
  COMMON/PG/PGB(7000),PGTG(7000),XTRA(480),YTRA(480),CEL(480)
  COMMON/SBF/MUCH(126),SQB(550)
  COMMON/HHL/MAPH(10),HIAT(350),LOAT(200),MAPX,MAPY
  COMMON/CRS/CLNM,CR(4),OCLI,OCUI
  COMMON/ZXY/ZX,ZY,ZASF,SPX,SPY,EPX,EPY
  NN=SPC1
10 CONTINUE
  FNAME=CEL(NN)
  CALL IFILE(QS,FNAME)
  READ(QS)MUCH
  IF(MUCH(1).NE.NN) PAUSE > CELL NO. ERROR>
  CR(1)=MUCH(9)
  CR(2)=MUCH(10)
  CR(3)=MUCH(11)
  CR(4)=MUCH(12)
  C1=ILEFT(CR(1))+XTRA(NN)
  C3=ILEFT(CR(2))+XTRA(NN)
  C5=ILEFT(CR(3))+XTRA(NN)
  C7=ILEFT(CR(4))+XTRA(NN)
  C2=IRIGHT(CR(1))+YTRA(NN)
  C4=IRIGHT(CR(2))+YTRA(NN)
  C6=IRIGHT(CR(3))+YTRA(NN)
  C8=IRIGHT(CR(4))+YTRA(NN)
  IF (NN.EQ.SPC1) X169=C1
  IF (NN.EQ.SPC1) Y169=C2
  IF (NN.EQ.SPC2) X1=C5
  IF (NN.EQ.SPC2) Y1=C6
  PRINT 111,FNAME,C1,C2,C3,C4,C5,C6,C7,C8
111 FORMAT (10X,>SKL>,10X,A5,8X,4(2I6,4X))
  IF (SPC2.NE.144) GO TO 200
  IF (NN.EQ.SPC2) GO TO 120
  NN=SPC2
  GO TO 10
120 CONTINUE
  ZEX=IABS(X1-X169)*100.0
  ZEY=IABS(Y1-Y169)*100.0
  ZX=IABS(EPX-SPX)/ZEX
  ZY=IABS(EPY-SPY)/ZEY
  PRINT 121,ZX,ZY
  RETURN
121 FORMAT (10X,>ZX=>,F7.4,10X,>ZY=>,F7.4)
200 CONTINUE
  PRINT 121,ZX,ZY
  RETURN
  END

```



```

      F=IRIGHT(CBF(K+2,1))
      G=ILEFT(CDF(K+2,1))
      IF ((A.GT.B).AND.(F.GT.B)) GO TO 110
      IF ((A.LT.B).AND.(F.LT.B)) GO TO 110
      GO TO 150
110  IF (IABS(D-E).GT.IABS(E-G)) GO TO 130
      IF (G.GE.E) GO TO 100
      EL=E-1
100  EL=E+1
      CALL IRLEFT (CBF(K+1,1),EL)
      GO TO 150
130  IF (D.GT.E) GO TO 140
      E=E-1
      GO TO 150
140  E=E+1
150  CONTINUE
C
      IF (B.EQ.A) GO TO 250
      AB=IABS(B-A)
      ZAB=FLOAT(AB)
      ZD=FLOAT(D)
      ZF=(E-D)/ZAB
      IF (D.LT.A) C=-1
C
      DO 210 L=A,B,C
      DD=ZD*0.5
      ZD=ZD+ZF
C
      DO 230 M=1,10
      IF (Y(M,L).EQ.DD) GO TO 240
      IF (Y(M,L).NE.9999) GO TO 230
      Y(M,L)=DD
      GO TO 210
230  CONTINUE
240  CONTINUE
      GO TO 234
250  CONTINUE
C
      TAKE AVERAGE X
      LOAD AVERAGE POINT
C
      DO 244 M=1,10
      IF (Y(M,B).EQ.D) GO TO 248
      IF (Y(M,B).NE.9999) GO TO 244
      Y(M,B)=D
      GO TO 248
244  CONTINUE
248  DO 252 M=1,10
      IF (Y(M,B).EQ.E) GO TO 252
      IF (Y(M,B).NE.9999) GO TO 256
      Y(M,B)=E
      GO TO 252
252  CONTINUE
256  CONTINUE
264  CONTINUE
280  CONTINUE
C
      TEST TO SEE IF POLYGON IS DONE
700  CONTINUE
720  CONTINUE
      PAREA=0
C
      DO 290 K=0,400

```

```

YAC(K)=0
IF (YAI(1,K).EQ.9999) GO TO 290
DO 274 I=10,1,-1
IF (YAI(I,K).EQ.9999) GO TO 274
EP=I
YAC(K)=EP
GO TO 276
276 CONTINUE
276 CONTINUE
IF (EP.EQ.1) GO TO 285
C
DO 282 L=1,10
N=0
DO 278 I=EP,2,-1
CO=YAI(I,K)
IF (CO.GT.YAI(I-1,K)) GO TO 278
YAI(I,K)=YAI(I-1,K)
YAI(I-1,K)=CO
N=N+1
278 CONTINUE
IF (N.EQ.0) GO TO 286
282 CONTINUE
285 CONTINUE
286 CONTINUE
C
DO 360 K=0,600
IF (YAI(1,K).EQ.9999) GO TO 360
EP=YAC(K)
C
IF (EP.EQ.1) GO TO 360
IF (EP.EQ.3) GO TO 300
IF (MOD(EP,2).EQ.1) GO TO 312
DO 303 I=1,EP,2
298 PAREA=PAREA+YAI(I+1,K)-YAI(I,K)
GO TO 300
C
IF (YAI(3,K)-YAI(2,K).GT.YAI(2,K)-YAI(1,K)) GO TO 304
PAREA=PAREA+(YAI(2,K)+YAI(3,K))/2-YAI(1,K)
GO TO 300
304 PAREA=PAREA+YAI(3,K)-(YAI(1,K)+YAI(2,K))/2
GO TO 300
C
312 CONTINUE
IF (YAC(K)-YAC(K-1).EQ.1) GO TO 300
IF (YAC(K)-YAC(K+1).EQ.1) GO TO 304
GO TO 360
320 L=1
GO TO 328
324 L=L-1
328 P=10000
DO 330 I=1,YAC(K)-L
N=0
DIF=0
DO 332 J=1,YAC(K)-L
IF (J.EQ.1) GO TO 330
DIF=DIF+I*ABS(YAI(J,K-L)-YAI(J+M,K))
GO TO 332
330 N=N+1
YCA=(YAI(J,1)+YAI(J+1,K))/2
DIF=DIF+I*ABS(YAI(J,K-L)-YCA)

```

```

340 CONTINUE
350 IF (100.EQ.1) GO TO 355
355
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570
580
590
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

SSW23--PRINTS POLYGON DATA
PRINT MATRIX

```

C
C
CALL SSWTCH (23,123)
IF (123.EQ.1) GO TO 300
PRINT 173
PRINT 271
DO 370 K=0,400
IF (Y(K),0).EQ.9999) GO TO 370
PRINT 273,P,K,(Y(I,K),I=1,YAC(K))
370 CONTINUE
300 CONTINUE
N1=LEFT(FGID(P))
L1=RIGHT(FGID(P))
N1A(N1)=N1A(N1)+PAREA
L1A(L1)=L1A(L1)+PAREA
TAR(N1)=TAR(N1)+PAREA
POND(P)=PAREA
000 CONTINUE
TAR=TAR+TAREA
C
C
SSW29--PRINTS AREAS
CALL SSWTCH (29,129)
IF (129.EQ.1) RETURN
PRINT 201,SECT,QSECT
801 FORMAT (1H0,10X,>SECTION >,12,5X,>QUARTERSECTION >,11)
ZTARA=0.0
DO 850 K=1,DSEE
IF (N1A(K).EQ.0) GO TO 850
ZATT=N1A(K)/100.0
ZPCT=ZATT/TAREA
ZARA=N1A(K)/(ZASF*100.0)
ZTARA=ZTARA+ZARA
PRINT 253,DSE(K),ZARA,ZPCT
850 CONTINUE
253 FORMAT (12X,A5,5X,F 8.2,7X,F7.3)
C
PRINT 277,ZTARA
RETURN
277 FORMAT (20X,F8.2)

```



```

      30
      31
      32
      33
      34
      35
      36
      37
      38
      39
      40
      41
      42
      43
      44
      45
      46
      47
      48
      49
      50
      51
      52
      53
      54
      55
      56
      57
      58
      59
      60
      61
      62
      63
      64
      65
      66
      67
      68
      69
      70
      71
      72
      73
      74
      75
      76
      77
      78
      79
      80
      81
      82
      83
      84
      85
      86
      87
      88
      89
      90
      91
      92
      93
      94
      95
      96
      97
      98
      99
      100
      101
      102
      103
      104
      105
      106
      107
      108
      109
      110
      111
      112
      113
      114
      115
      116
      117
      118
      119
      120
      121
      122
      123
      124
      125
      126
      127
      128
      129
      130
      131
      132
      133
      134
      135
      136
      137
      138
      139
      140
      141
      142
      143
      144
      145
      146
      147
      148
      149
      150
      151
      152
      153
      154
      155
      156
      157
      158
      159
      160
      161
      162
      163
      164
      165
      166
      167
      168
      169
      170
      171
      172
      173
      174
      175
      176
      177
      178
      179
      180
      181
      182
      183
      184
      185
      186
      187
      188
      189
      190
      191
      192
      193
      194
      195
      196
      197
      198
      199
      200
      201
      202
      203
      204
      205
      206
      207
      208
      209
      210
      211
      212
      213
      214
      215
      216
      217
      218
      219
      220
      221
      222
      223
      224
      225
      226
      227
      228
      229
      230
      231
      232
      233
      234
      235
      236
      237
      238
      239
      240
      241
      242
      243
      244
      245
      246
      247
      248
      249
      250
      251
      252
      253
      254
      255
      256
      257
      258
      259
      260
      261
      262
      263
      264
      265
      266
      267
      268
      269
      270
      271
      272
      273
      274
      275
      276
      277
      278
      279
      280
      281
      282
      283
      284
      285
      286
      287
      288
      289
      290
      291
      292
      293
      294
      295
      296
      297
      298
      299
      300
      301
      302
      303
      304
      305
      306
      307
      308
      309
      310
      311
      312
      313
      314
      315
      316
      317
      318
      319
      320
      321
      322
      323
      324
      325
      326
      327
      328
      329
      330
      331
      332
      333
      334
      335
      336
      337
      338
      339
      340
      341
      342
      343
      344
      345
      346
      347
      348
      349
      350
      351
      352
      353
      354
      355
      356
      357
      358
      359
      360
      361
      362
      363
      364
      365
      366
      367
      368
      369
      370
      371
      372
      373
      374
      375
      376
      377
      378
      379
      380
      381
      382
      383
      384
      385
      386
      387
      388
      389
      390
      391
      392
      393
      394
      395
      396
      397
      398
      399
      400
      401
      402
      403
      404
      405
      406
      407
      408
      409
      410
      411
      412
      413
      414
      415
      416
      417
      418
      419
      420
      421
      422
      423
      424
      425
      426
      427
      428
      429
      430
      431
      432
      433
      434
      435
      436
      437
      438
      439
      440
      441
      442
      443
      444
      445
      446
      447
      448
      449
      450
      451
      452
      453
      454
      455
      456
      457
      458
      459
      460
      461
      462
      463
      464
      465
      466
      467
      468
      469
      470
      471
      472
      473
      474
      475
      476
      477
      478
      479
      480
      481
      482
      483
      484
      485
      486
      487
      488
      489
      490
      491
      492
      493
      494
      495
      496
      497
      498
      499
      500
      501
      502
      503
      504
      505
      506
      507
      508
      509
      510
      511
      512
      513
      514
      515
      516
      517
      518
      519
      520
      521
      522
      523
      524
      525
      526
      527
      528
      529
      530
      531
      532
      533
      534
      535
      536
      537
      538
      539
      540
      541
      542
      543
      544
      545
      546
      547
      548
      549
      550
      551
      552
      553
      554
      555
      556
      557
      558
      559
      560
      561
      562
      563
      564
      565
      566
      567
      568
      569
      570
      571
      572
      573
      574
      575
      576
      577
      578
      579
      580
      581
      582
      583
      584
      585
      586
      587
      588
      589
      590
      591
      592
      593
      594
      595
      596
      597
      598
      599
      600
      601
      602
      603
      604
      605
      606
      607
      608
      609
      610
      611
      612
      613
      614
      615
      616
      617
      618
      619
      620
      621
      622
      623
      624
      625
      626
      627
      628
      629
      630
      631
      632
      633
      634
      635
      636
      637
      638
      639
      640
      641
      642
      643
      644
      645
      646
      647
      648
      649
      650
      651
      652
      653
      654
      655
      656
      657
      658
      659
      660
      661
      662
      663
      664
      665
      666
      667
      668
      669
      670
      671
      672
      673
      674
      675
      676
      677
      678
      679
      680
      681
      682
      683
      684
      685
      686
      687
      688
      689
      690
      691
      692
      693
      694
      695
      696
      697
      698
      699
      700
      701
      702
      703
      704
      705
      706
      707
      708
      709
      710
      711
      712
      713
      714
      715
      716
      717
      718
      719
      720
      721
      722
      723
      724
      725
      726
      727
      728
      729
      730
      731
      732
      733
      734
      735
      736
      737
      738
      739
      740
      741
      742
      743
      744
      745
      746
      747
      748
      749
      750
      751
      752
      753
      754
      755
      756
      757
      758
      759
      760
      761
      762
      763
      764
      765
      766
      767
      768
      769
      770
      771
      772
      773
      774
      775
      776
      777
      778
      779
      780
      781
      782
      783
      784
      785
      786
      787
      788
      789
      790
      791
      792
      793
      794
      795
      796
      797
      798
      799
      800
      801
      802
      803
      804
      805
      806
      807
      808
      809
      810
      811
      812
      813
      814
      815
      816
      817
      818
      819
      820
      821
      822
      823
      824
      825
      826
      827
      828
      829
      830
      831
      832
      833
      834
      835
      836
      837
      838
      839
      840
      841
      842
      843
      844
      845
      846
      847
      848
      849
      850
      851
      852
      853
      854
      855
      856
      857
      858
      859
      860
      861
      862
      863
      864
      865
      866
      867
      868
      869
      870
      871
      872
      873
      874
      875
      876
      877
      878
      879
      880
      881
      882
      883
      884
      885
      886
      887
      888
      889
      890
      891
      892
      893
      894
      895
      896
      897
      898
      899
      900
      901
      902
      903
      904
      905
      906
      907
      908
      909
      910
      911
      912
      913
      914
      915
      916
      917
      918
      919
      920
      921
      922
      923
      924
      925
      926
      927
      928
      929
      930
      931
      932
      933
      934
      935
      936
      937
      938
      939
      940
      941
      942
      943
      944
      945
      946
      947
      948
      949
      950
      951
      952
      953
      954
      955
      956
      957
      958
      959
      960
      961
      962
      963
      964
      965
      966
      967
      968
      969
      970
      971
      972
      973
      974
      975
      976
      977
      978
      979
      980
      981
      982
      983
      984
      985
      986
      987
      988
      989
      990
      991
      992
      993
      994
      995
      996
      997
      998
      999
      1000

```

C

```

      IF (EC01.EQ.0) CALL NT9R (ECD,200)
      IF (NT9F.EQ.3) GO TO 40
      ONE=1
      DO 20 I=1,20

```



```
      KIDD=KIDD+1  
      KCARDN1=KIDPFKIDN  
20  CONTINUE  
      IF (KIDD.LT.200) RETURN  
      KIDD=0  
      RETURN  
40  CONTINUE  
      KID=1  
      RETURN  
      END
```

B-0

APPENDIX B

OPERATIONAL KEY/SUBROUTINE TABLES

TABLE 1.
MAP: Teletype Key Commands

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	RESULT	ROUTINE USED
/	Disconnects any program from track-ball			DISTTY
A	Control is transferred from MAIN to AL2	Scanning constants entered	Starts cell scanning mode	DISTTY
C	Changes horizontal scan to vertical scan			DISTTY
D	Connects density threshold to track-ball		Sets density threshold	DISTTY
E	Connects east boundary of scanning aperture to track-ball		Changes east boundary	DISTTY
G	Transfers control to IDENK		Enters map page constants	DISTTY
H	Changes scanning station		Not used in MAP program	DISTTY
I	Initializes scanning parameters			DISTTY
J	Saves current parameters			DISTTY
L	Reads DEC tapes		Not used in MAP program	DISTTY
M	Writes DEC tapes			DISTTY
N	Connects track-ball to north boundary		Changes north boundary	DISTTY

TABLE 1. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	RESULT	ROUTINE USED
P	Carriage movement control			DISTTY
PH	Carriage home reference		Map page starting position	DISTTY
PT	Connects track-ball to carriage		Cell adjustment	DISTTY
P(a,a)	Moves carriage to initial cell	where (a,a) is the cell position in the overall matrix		DISTTY
PI	Resets carriage reference position		Carriage movement with track-ball	DISTTY
Q	Changes scan search from black space to white space or vice versa		To better observe scanning operation	DISTTY
S	Connects track-ball to south boundary		To change south boundary	DISTTY
U	Control to TTY to enter scanning parameters		Replaces track-ball at user's discretion	DISTTY
V	Positions 9 track tape		Not used in MAP program	DISTTY
W	Connects track-ball to west boundary		To change west boundary	DISTTY
X	Connects track-ball to 'X' increment		Not used in MAP program	DISTTY

TABLE 1. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	RESULT	ROUTINE USED
Y	Connects track-ball to 'Y' increment		Not used in MAP program	DISTTY
!	Transfers control to program DENSITY		Uses density threshold for threshold setting	DISTTY
CSX	Cell size in 'X'		Types in cell size in 'X' direction	IDENK
CSY	Cell size in 'Y'		Types in cell size in 'Y' direction	IDENK
PAX	Page number 'X'		Types in page identifier	IDENK
PAY	Page number 'Y'		Types in page identifier	IDENK
PMX	Page matrix 'X'		Types in page dimension in 'X'	IDENK
PMY	Page matrix 'Y'		Types in page dimension in 'Y'	IDENK
A	Continue		Operators option	AL2
Z	Return to MAIN		Operators option	AL2
S	Skip cell		Operators option	AL2

TABLE 1. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	RESULT	ROUTINE USED
A	Accepts displayed polygon			NBINDISPLAY
R	Displays all polygons		Used to locate a polygon	NBINDISPLAY
Z	Returns to single polygon mode			NBINDISPLAY
!	Rejects polygon			NBINDISPLAY
space bar	Used with track-ball and cursor for polygon boundary filtering			SOLTYDSP
K	Separates two joined polygons	space bar		SOLTYDSP
R	Returns to NBINDISPLAY			SOLTYDSP
space bar	Used with track-ball and cursor to remove polygons from screen in major attribute mode		Associates polygon with entered attributes	ATT
/	Displays all polygons with common attributes in major attribute mode		Used for checking polygon attributes	ATT
:	Displays all polygons with common attributes in minor attribute mode		Used for checking polygon attributes	ATT

TABLE 2.
MAP: Subroutine Functions

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
MAIN	Initializing mode - continually scans selected region and displays scanning constants for operator observation and intervention	Scanning station, TTY, display station	X and Y coordinates of crossings		
DISTTY	Operates in conjunction with MAIN. Interprets operator commands and directs control to proper programs	Scanning station, TTY, display station	Operator entered TTY commands	Control transfers as directed by operator commands (Table 1)	See Teletype Key Commands
MOVERL	Sends commands to carriage on reflected light scanner	Reflected light scanner	Command information from various sources	Interpreted command information	
BOUNDS	Draws box on slave scope around scanning region	Slave display station	Coordinates of box from MAIN	Interpreted coordinates	-
DENSITY	Commands scanner in density mode to scan region - assembles density points and plots histogram on graphic display terminal	Graphic display station	Commands from DISTTY	Visual display of histogram	!
IDENK	Accepts map page information, page number, cell matrix size, cell size		Operator inputs map page information	Program constants	G

TABLE 2. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
AL2	Control program for map processing operation: transfers control to ISOLATE for scanning and polygon generation, transfers control to NBINDISPLAY for polygon separation and operator intervention, and transfers control to MOVERL for carriage movement				A Z S
ISOLATE	Assembles X and Y coordinate data and forms line segments of surface crossings. Tags all line segments with unique tag number for each surface or polygon		X and Y coordinate data from NEXTL	Line segment data tagged according to polygon number	B S
NEXTL	Directs scanner to scan one line of map. Collects X and Y coordinates of surface crossings. Works with ISOLATE	Reflected light scanner	X and Y coordinate data from reflected light scanner	X and Y coordinate data to ISOLATE	
SOLTYDSP	Polygon perimeter and area are displayed at graphic display station. Operator function: 1. boundary filtering; and 2. attribute identification	Graphic display station, TTY, trackball, magnetic tape unit	Unprocessed polygon data	Processed data with attributes identified, output to 9 TK tape	space bar R Z K

TABLE 2. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
EXAMINE	Transforms line segments of polygon data to points on perimeter of polygon - Data is in incremental format		Line segment polygon data	Incremental perimeter data - Area of polygon	
NBINDISPLAY	Displays line segment data to operator for observation and decision making, segments of four corners are displayed for map marker location and coordinate input. Segments for each polygon are displayed for operator's acceptance or rejection	Graphic display station, track-ball, TTY	Line segment data	Map marker locations - line segments separated according to polygon number	space bar Z R
CLOSEOUT	Map cell is terminated, area is corrected for line width, area data and attributes are formatted and output to magnetic tape	Magnetic tape unit	Unformatted area data and polygon attributes	Formatted output to 9 TK tape	
ATT	An alternate method of attribute identification	7 TK magnetic tape unit, 9 TK magnetic tape unit, Graphic display station, track-ball, TTY	Polygon data less attributes from from 7 TK tape	Polygon data with attributes to 9 TK tape	/ .

TABLE 3.
CHAIN: Subroutine Function

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
CHA	Reads 550 word records from 9 track tape - unpacks buffers and separates perimeter information from area, attributes and map markers	9 Tk tape unit	ALICE polygon data	ALICE polygon perimeter data	Directed by system
CLRBF	Clears all buffers and prepares system to accept a cell of map data				
DIM	Accepts ALICE polygon data and converts to chain DATA		Polygon data	Chain data	Automatic unless in test mode
DIN	Accepts chain data and forms nodes between all chains		Chained data with out nodes	Chain data with inserted nodes	Directed by system
DIP	Tries cell data to area bounded by map markers. Formats and outputs finished cell data	7 Tk tape unit	Chain and node data	Formatted chain and node data	Directed by system

or
to

TABLE 4.
FIX: General Map Information

VARIABLE NAME	TYPE	TITLE	DESCRIPTION
PGTB	Array	Polygon attribute Index table	For each number associated with a scanned polygon, there is an entry consisting of two separate indices which index two different attribute tables. This permits dual attribute assignment for each polygon (i.e., county/soil, soil/slope, state/ecosystem type, etc.)
PGTG	Array	Equivalent polygon tag table	For each number associated with a scanned polygon, there is a numeric pointer to equivalent polygons in the map. Although each scanned polygon is uniquely identified, other pieces of the same physical polygon may appear in more than one cell. For this reason, they are all tagged as being equivalent.
FIAT	Array	First attribute table	For every possible first attribute index in PGTB there is an entry whose value can be up to five ASCII characters. (Usually a geographic code such as a state or county abbreviation, is used as the first attribute for a polygon.)
LOAT	Array	Second attribute table	Same as above, but for every possible second attribute index. (Usually the physical characteristic, such as soil type, is used as the second attribute for a polygon.)
CELL	Array	Cell file name table	For each scanning aperture cell imaged on the map there is a unique cell number which indicates the cell's location in the map's scanning cell matrix. The entry for each cell number is the ASCII character file name used for that cell on the disk.
CLPG	Array	Cell polygon index table	For each cell number, the table contains the numbers of the first and the last polygons contained in that scanned cell.

TABLE 4. (Cont'd)

VARIABLE		TITLE	DESCRIPTION
NAME	TYPE		
XTRA	Array	Global X translation table	These tables contain the global X and Y translations for each cell. Since the X,Y coordinates in each cell are normalized to begin at zero, the translation values are added to the X and Y values for each point in each cell in order to accurately position the cells before displaying or plotting more than one scanned cell.
YTRA	Array	Global Y translation table	
PGCT	Scaler	Polygon count	Contains the total number of polygons in the scanned cell. This is not the number of unique polygons in the map due to polygon equivalence.
CLCT	Scaler	Cell count	Contains the number of the last scanned cell in the map's scanning cell matrix.
MAXGX	Scaler	Maximum global X value	These are the largest X and Y values after all map values are normalized so that the lower left hand corner is the origin.
MAXGY	Scaler	Maximum global Y value	
MAPX	Scaler	Map matrix X dimension	These are the map's scanning cell matrix dimensions. They are used in creating the scanned cell file name and in some cell number error checking.
MAPY	Scaler	Map matrix Y dimension	
MAPN	Array	Map name	This is a fifty-character array used to store a name for the map.

TABLE 5.
FIX: Scanned Cell File Information

VARIABLE		TITLE	DESCRIPTION
NAME	TYPE		
CLID	Scaler	Cell identifier	Contains the scanning matrix X and Y coordinates of the cell so as to uniquely identify each cell.
CLST	Scaler	Cell status	Contains a cell status word: 0 = NEW, never displayed with another cell. No translation values generated. 1 = OLD, has been displayed with other cells. Translation values have been generated. 2 = SET, has been positioned to its final location in the map. Translation values should not be changed.
NOCN	Scaler	Number of chains	Contains the total number of chains in this cell.
NOPT	Scaler	Number of points	Contains the total number of points in this cell.
MAXX	Scaler	Maximum X value	Contains the maximum X and Y coordinate values that occur in the cell.
MAXY	Scaler	Maximum Y value	
CR	Array	Cross coordinates	Contains the X and Y coordinates of the four corners of the cell corresponding to the original locations of the crosses that were marked on the map.
CP	Array	Chain pointer	Contains a pointer to the first X and Y coordinate pair for each chain of polygon data in the scanned cell. Commonly referred to as a beginning node.
ECP	Array	Chain end pointer	Contains a pointer to the last X and Y coordinate pair for each chain of polygon data in the scanned cell. Also referred to as an ending node.

TABLE 5. (Cont'd)

VARIABLE		TITLE	DESCRIPTION
NAME	TYPE		
CBF	Array	Chain buffer	This one large array contains all of the X and Y coordinate pairs for all of the polygon chains contained in a scanned cell. The beginning node and ending node for a chain are located by using the corresponding chain pointers.

TABLE G.
FIX: Current Program Information

VARIABLE		TITLE	DESCRIPTION
NAME	TYPE		
PDPT	Scaler	Data Stack Pointer	Contains the number of elements of information in the data stack.
PDCL	Array	(DATA STACK) Cell number	Each element is the number of the cell containing the chosen chain end point.
PDCN	Array	(DATA STACK) Chain number	Each element is the number of the chain containing the chosen end point.
PDPP	Array	(DATA STACK) Point pointer	Each element is a pointer to the chosen chain end point.
PDX	Array	(DATA STACK) X position	Each element is the cross hair X position.
PDY	Array	(DATA STACK) Y position	Each element is the cross hair Y position.
DSTAT	Scaler	Display Status	Contains the number of cells currently in core.
DPGN	Scaler	Display Polygon Number	Contains the number of the desired polygon to be shown, equals zero if all polygons are to be shown.
DHAT	Scaler	Display First Attribute	Contains the index number of the desired attribute to be shown, equals zero if all attributes are to be shown.
DLAT	Scaler	Display Second Attribute	
FSF	Scaler	Final Scaling Factor	Contains the scaling factor currently being used to display the polygon data.
BXTRA	Scaler	Base X Translation	Contains the minimum global X translation of the cells currently in core.

TABLE 6. (Cont'd)

VARIABLE		TITLE	DESCRIPTION
NAME	TYPE		
BYTRA	Scaler	Base Y Translation	Contains the minimum global Y translation of the cells currently in core.
BX	Scaler	Base X Displacement	Contains the X and Y displacements needed to center the current cells on the display.
BY	Scaler	Base Y Displacement	

TABLE 7.
FIX: Teletype Key Commands for Primary (cell editing) Mode

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
A	Automatically position the displayed cells to their final location	At least one cell on display must have been previously set		All cells are marked as set	ACSET
B	Change display back to normal scale (after a zoom)			New scaling factor and offset values are calculated	RESCAL
C	Copy the displayed data onto the plotter	Plotter ready		Plot of the displayed data	COPY
D	Display a particular subset of polygons		Desired polygon number or polygon attributes	Displays requested polygons	SHOWSM
E	Enter the chain editing mode	Data stack contains chains to be edited	Chain editing mode commands	(See Table 8)	CHNED
F	Copy all of the general map information to the tables file			New tables file generated	FINISH
G	Manually generate a chain	Data stack contains the starting node of the chain	Path of the chain via the track ball	New chain added to the cell	CHNGEN
H	Print a list of the primary mode commands			Listing of commands on printer	MAIN

TABLE 7. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
I	Increment the displayed polygon number or the displayed attribute	A particular polygon number or attribute is already being displayed		Displays polygons with next attribute or polygon number	MAIN
J	Join chain endpoints together to create a node	Data stack contains chain endpoints to be joined		Node is created at average of chain endpoints, polygons are equated if possible	NJOIN
K	Remove a cell's data from the current cell information tables		Number of cell to be removed	Cell data is removed	WRTCEL
L	Print polygon and/or cell information		Which listings desired	Polygon and/or cell information printed on line printer	LIST
M	Move a cell	Data stack contains old and new cell locations	None (unless trying to move a set cell)	Cell moved to new location	CMOVE
N	Move a node	Data stack contains old and new node locations		Node moved to new location	NMOVE
O	Change tag of chain endpoint to indicate border endpoint or not	Data stack contains those endpoints to be affected		Tag of each endpoint is changed	MAIN

TABLE 7. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
P	Plot the desired polygons for the entire map	Desired polygon information already set	Desired maximum dimension of plotted map	Plot of desired polygons for entire map	PLOTAL
Q	Leave the FIX program		Confirmation of being finished	Save tables in table file, leave program	MAIN
R	Load a cell file into the current cell information tables	There is room for another cell	Number of cell to be loaded	New cell's information in tables	RDCELL
S	Mark all cells on display as set			All displayed cells marked as set	MAIN
T	Automatically join border endpoints of displayed cells wherever possible	Cells to be joined must be set		Chain endpoints of adjacent cells joined and untagged wherever possible	ANTIE
U	Verify a border chain endpoint and put information in the data stack	Cross hair at desired location on display, point not previously in data stack		New set of chain endpoint information added to the data stack	MAIN
V	Verify a chain endpoint near this location and save endpoint information in the data stack	Cross hair at desired location on display, point not previously in data stack		New set of chain endpoint information added to the data stack	MAIN

TABLE 7. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
W	Replace a cell file with the current in-core information for that cell, then remove the information from the tables		Number of cell file to be replaced	Old cell file replaced with new data, information removed from in-core	WRTCEL
X	Remove all unnecessary points from the cells currently in core			Original and new number of points for each cell	REMOVE
Y	Remove a node	Data stack contains node to be removed		Endpoints of all chains entering the node are deleted	NMOVE
Z	Zoom in on an area of the display	OPTIONAL: The data stack may contain two points defining the edges of a zoom		Either double the display scale with the cross hair at the center or make zoom window fill display	MAIN
(SPACE)	Save these X and Y coordinates and the closest chain endpoint information in the data stack	Cross hair at desired location		New set of information in data stack	MAIN
1	Position all free cells in the map			All cells positioned to final location	ACPOS

TABLE 7. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
2	Check cells for certain bad conditions		Want all cells or only those on display, which conditions to check for	Location and type of bad condition	CHECK
4	Check polygons for bad conditons starting with the displayed polygon			Number of bad polygon, type and location of bad condition	CHECKP
5	Copy all files from one device to another (i.e. DEC tape to disk or disk to DEC tape)	No cells on display, sense switches set as desired		New copy of cell files and table file	MAIN
6	Copy all files to 9 track tape for storage			All files saved on 9 track tape	MAGRW
7	Programmed for the particular task				MAKE
8	Programmed for the particular task				MAKE
9	Programmed for the particular task				MAKE

TABLE 8.
 FIX: Teletype Key Commands for Secondary (chain editing) Mode

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT	ROUTINE USED
A	Change a polygon attribute of this chain		Right or left polygon, which attribute, new attribute	Change made in PGTB for this polygon	CHNED
B	Show the polygon data again, changing display mode		Change sense switch 8 if desired to change display mode	Desired polygon data displayed	CHNED
D	Delete this chain			Chain removed from cell and data stack	CHNED
E	Set two polygons equivalent	Polygon attributes matched	Polygon numbers of the two polygons to be set equivalent	PGTG changed to show polygons equivalent	CHNED
F	Find an unused polygon number in this cell	Unused number available		Number of available polygon listed on TTY	CHNED
H	Print a list of possible commands			List of commands on printer	CHNED
L	List the information for this chain			Number and attributes of polygons on either side of the chain listed on Teletype	CHNED

TABLE 8. (Cont'd)

TELETYPE KEY COMMAND	FUNCTION PERFORMED	NECESSARY PRE-CONDITIONS	OPERATOR INPUT	RESULT.	ROUTINE USED
M	Make a node along the chain thereby making the chain into two chains		Location of new node	Old chain removed, new chains added to cell	CHNED
N	Edit the next chain in the stack			Next chain is put in editing position	CHNED
P	Change a polygon number of this chain		Which polygon to change, new polygon number	Polygon data changed for this chain	CHNED
Q	Leave the chain editing mode and return to the primary (cell editing) mode			Data stack cleared, control returned to primary mode	CHNED
S	Search the chains in this cell for another chain with the same endpoint		Choose chain	Desired chain is put into editing position	CHNED
T	Change the polygon tag		Polygon to change, new tag number	Polygon tag changed	CHNED
U	Mark a polygon as not being equivalent to any other polygon		Polygon number of polygon to be marked not equivalent	Polygon set equivalent to itself	CHNED
+	Copy displayed information onto film	Camera ready	Number of exposures to be taken	Picture taken and film advanced	Picture

TABLE 9.
FIX: Sense Switch Options

SWITCH NUMBER	SWITCH POSITION	FUNCTION	ROUTINES AFFECTED
0	Off On	Use the disk as the cell file input device Use the DEC tape as the cell file input device	RDCELL, WRCELL READ1, FINISH
1	Off On	Use the disk as the cell file output device Use the DEC tape as the cell file output device	RDCELL, WRCELL LOAD1
2	Off On	Do not display the locations of the cell corners Mark the locations of the cell corners on the display	SHOWAL
4	Off On	Display in-core cell information at the top of the display screen Do not display the in-core cell information	SHOWAL
6	Off On	Do not mark untied border chain endpoints on the display screen Mark all untied border chain endpoints on the display screen	SHOWAL
8	Off On	Show all chains contained in the current cells Show only the chosen subset of the current polygons	SHOWAL
10	Off On	Automatically calculate the map matrix cell coordinates while loading cells Manually assign cell coordinates while loading cells	LOAD1
12	Off On	Check all nodes when checking for polygon closure Do not check those polygons which have open border chain endpoints when checking for polygon closure	CHECKP
14	Off On	Move all the chain endpoints of a node when moving the node. Move only a single chain endpoint of a node when moving the node	MOVE

107

TABLE 9. (Cont'd)

SWITCH NUMBER	SWITCH POSITION	FUNCTION	ROUTINES AFFECTED
16	Off On	Use map information from the tables file Allow for inputting new general map information before going to the cell editing mode	MAIN
17	Off On	List switch options on line printer during start-up Do not list switch options.	MAIN
18	Off On	Plot information with X-axis across paper Plot information with X-axis along paper	PLOTAL
19	Off On	Check for closed polygons in CHECKP Clear unused polygon numbers only in CHECKP	CHECKP
20	Off On	Suppress printing of diagnostic data Print diagnostic data	LOADT, REMOVE, REDUCE
21	Off On	Suppress printing of extensive diagnostic data Print extensive diagnostic data	LOADT, REDUCE,
22	Off On	Assign new polygon numbers to the cells as they are loaded If reloading a previously loaded cell, use previous cell's polygon numbers	LOADT
23	Off On	Load all files from the magnetic tape Load only the specified file from the magnetic tape	MAGRW
24	Off On	Complete the current command Abort the current command	

TABLE 9. (Cont'd)

SWITCH NUMBER	SWITCH POSITION	FUNCTION	ROUTINES AFFECTED
25	Off On	Check those conditions chosen via switches 28-34 Use routine XCHECK to remove specific chains	CHECK
28	Off On	Do not check for single point chains Remove single point chains	CHECK
29	Off On	Do not check for polygon numbers outside of cell Check for poygon numbers outside of cell	CHECK
30	Off On	Do not check for untied border chains Check for untied border chains	CHECK
31	Off On	Do not check for unnecessary chains Check for unnecessary chains	CHECK
32	Off On	Do not check for illegal attribute indices Check for illegal attribute indices	CHECK
33	Off On	Do not check for chain loops Check for chain loops	CHECK
34	Off On	Do not check for inconsistent polygon tags Check for inconsistent poygon tags	CHECK
35	Off On	Clear all in-core tables during start-up procedure Do not clear the in-core tables during start-up and go directly to the cell file editing mode (allows for machine error recovery)	MAIN
All other switches ARE NOT currently used by FIX			

TABLE 10.
FIX: Subroutine Functions

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
LOADT	Reads the cell data output from the CHAIN foundation and creates the cell files on the disk for use with the FIX foundation.	7 track tape, Disk or DEC Tape, Card Reader	Scanned cell data from CHAIN and map information	Cell files	-
READT	Loads the information from the tables file on the disk into the appropriate in-core general map information tables.	Disk or DEC Tape	Table file	General map information tables	-
RDCELL	Loads the information for a specific cell from the cell file into one set of temporary cell tables.	Disk or DEC Tape	Cell file	One set of temporary cell information tables	R
WRTCEL	Writes one set of current cell information tables to a cell file, replacing the old cell file. Also removes the information from core.	Disk or DEC Tape	One set of temporary cell information tables	One cell file	W,K
FINISH	Copies the general map information to the tables file.	Disk or DEC Tape	General map information tables	TABLE file	F
MAGRW	Saves and reloads all files for a map.	9 track tape; Disk or DEC Tape	TABLE file and all cell files	9 track tape file containing all the map information	6

TABLE 10. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
CMOVE	Changes the position of a cell by updating the X and Y translation tables	-	X and Y positions from data stack	X and Y translation values for this cell	M
NMOVE	Changes the location of a node by changing the coordinates of the chain endpoints making up the node.	-	Node location from data stack	New coordinates for the chain endpoints of the node	N
CHECK	Checks for certain bad conditions in the in-core cell data, such as unjoined boundary endpoints, illegal attribute indices, and unnecessary chains.	ALICE display	Scanned cell data from current cell information tables	Location of bad condition	2
CHECKP	Verifies that all polygons in the map are closed and checks for complex polygons.	ALICE display	Beginning polygon number from current program information	Number of bad polygon and location of bad node on polygon	4
RESCAL	Calculates the proper scaling factor and offset values to allow all of the in-core cell data to fit on the ALICE display.	-	Cell translation values and cell MAXX and MAXY values	Scaling factor, base translation values, base display values	B
SHOWAL	Displays all or a subset of the polygon information contained in those cells currently in-core.	ALICE display	Current program information, general map information, in-core cell information	Display of polygon data	-

TABLE 10. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
COPY	Plots the displayed information onto the plotter.	Pen plotter	Current program information, general map information, in-core cell information	Plot of polygon data	C
LIST	Prints general polygon data on the line printer.	Line printer	General map information	List of general polygon information	L
SHOWSM	Sets information for the display of a specific subset of polygons.	-	Temporary program information	Desired polygon number or desired attributes	D
NJOIN	Joins chain endpoints together to create nodes by giving the endpoints a common coordinate. Also checks if possible to equate any polygons.	-	Location of chain endpoints to be joined from data stack	Node located at average of chain endpoint coordinates	J
REMOVE	Reduces the number of points in a scanned cell by removing unnecessary points along straight lines. Considers three points and removes center point if on the line created by the other two points.	-	Chain data from current cell information tables	Reduced chain data	X

TABLE 10. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
ACSET	Automatically positions in-core cells to their final positions by calculating final position of each unset cell considering all set cells and using CMOVE.	-	Locations of the corners for all in-core cells	Final X and Y translations of the in-core cells	A
ANTIE	Automatically joins tagged boundary chain endpoints of adjacent in-core cells by checking for proximity and attribute matching and then calling NJOIN.	-	Coordinates of all in-core tagged chain endpoints	Integrated cell boundaries	T
CHNGEN	Allows operator to manually generate a chain and then calls CHNED so the proper attributes may be assigned to it.	Track-ball	Location of beginning node from data stack	New chain	G
CHNED	Allows editing of chain attributes, deleting of chains, and other data manipulation functions.	Track-ball, Teletype, display	Location of endpoints of chains to be edited, general map information, cell data	Corrected data	E
PLOTAL	Plots the desired polygons for the entire map using COPY	-	General map information, program information	Temporary program information	P

TABLE 10. (Cont'd)

SUBROUTINE NAME	OPERATION PERFORMED	PERIPHERALS	SUBROUTINE DATA		COMMAND(S)
			INPUT	OUTPUT	
ACPOS	Positions all free cells in the map to their final location by using ACSET	-	Corner locations of all the cells	Final X and Y translations of all cells	1
XCHECK	Custom programmed to remove specific chains. Called from CHECK as an option.	-	Location of current chain being checked.	Chain is deleted or left alone	(2)
MAKE	Custom programmed to perform the required function	-	General map information, current cell information	-	7,8,9

TABLE 11.
FIX: Data Organization

