

ภาคผนวก ก

คำอธิบายความหมายของสัญลักษณ์ที่แสดงในตารางที่ 2.9

all	: ให้ความละเอียดไม่เท่ากันในแต่ละขบวนการ (processes)
C	: แบบจำลองที่ทำนายระบบอุทกวิทยาแบบต่อเนื่อง (continuous) ซึ่งช่วงเวลาอาจจะเป็นทุกชั่วโมง, วัน หรือเดือน ก็ได้
Cm	: ภูมิอากาศ (climate)
CP	: แบบจำลองที่แยกขบวนการต่าง ๆ ออกจากกันแล้วใช้ขบวนการทาง คณิตศาสตร์อธิบายขบวนการต่าง ๆ เหล่านั้น
Cr	: การหลากในลำน้ำ (channel routing)
D	: แบบจำลองที่ทำนายระบบอุทกวิทยาแบบไม่ต่อเนื่อง (discrete) เช่น พายุฝนลูกเดียว (single storm) เป็นต้น
E-T	: การคายระเหย (evapotranspiration)
G	: ดี (good)
I	: ให้ความละเอียดปานกลาง อธิบายการไหลของน้ำโดยใช้ Kinematic Wave Approximation
In	: การดัก (interception)
Is	: การซึมลงดิน (infiltration)
Ma	: ให้ความสัมพันธ์ของน้ำทำอย่างง่าย รวมกับเทคนิคการหลากของน้ำ (flow routing techniques)
Mi	: ให้ความละเอียดทางอุทกวิทยา โดยใช้สมการไดนามิกแบบสมบูรณ์ (complete dynamic equation)
P	: ไม่ดี (poor)
Pi	: แบบจำลองที่อธิบายการไหลของน้ำในลำน้ำโดยอาศัยสมบัติเพียง ตัวเดียวหรือมากกว่า ซึ่งอาจมีความหมายทางกายภาพ (physical significance) หรือไม่ก็ได้ โดยปกติพารามิเตอร์เหล่านี้มักเป็น แบบเหมารวม (lumped parameters)

- R : แบบจำลองที่ใช้ความสัมพันธ์เชิงสถิติ ของการไหลของน้ำท่าจากลุ่มน้ำ (watershed runoff) ที่ขึ้นอยู่กับตัวแปรอิสระ (independent variable) เพียงตัวเดียว หรือมากกว่า
- RF : ฝน (rainfall)
- S : การตกตะกอน (sedimentation)
- St : การเก็บกักในดิน (storage)
- VG : ดีมาก (very good)
- WQ : แบบจำลองที่ทำนายคุณภาพ (quality) หรือความเข้มข้น (concentration) หรือทั้งสองอย่าง สำหรับบางช่วงเวลา เช่น 1 ปี หรือ 1 เดือน เป็นต้น
- * : ข้อมูลไม่สมบูรณ์

ภาคผนวก ข .

สรุปลักษณะอุทกวิทยาของแบบจำลอง

* คุณสมบัติทั่วไป

- ชนิด : การจำลองสภาพชลภาพน้ำท่า
- ขนาดลุ่มน้ำที่ใช้ : ใช้ได้กับลุ่มน้ำขนาดเล็กและขนาดใหญ่
- การแทนค่าพารามิเตอร์ : แปรเปลี่ยนไปกับสถานที่และเวลา
- ภาษาที่ใช้ในการประมวลผลข้อมูล : MAPS-Geographical
Information System

ขบวนการอุทกวิทยาที่ใช้

- * ขบวนการบนพื้นผิว : ใช้สมการ SCS (ดูตารางที่ 3.1)
การตัด ทาบปริมาณฝนส่วนเกิน
การเก็บกักบนพื้นผิว
การซึม
การเก็บกักความชื้น
- * ขบวนการใต้พื้นผิว : ไม่ได้พิจารณา
- * ขบวนการหลาก : Time-Area method หาเวลาการ
หลากของการไหลบนพื้นผิว โดยใช้
สมการ 3.1 สำหรับการไหลบนผิวดิน
และสมการ 3.3 สำหรับการไหลในลำน้ำ

ขบวนการแสดงคุณสมบัติทางกายภาพ

* กฎการอนุรักษ์ของมวล

สำหรับ grid cell ใด ๆ

ที่พิจารณา โดยการพิจารณา

ใช้หลักการสมมูลย์ของน้ำจะ

ได้ดังนี้

ใช้สมการ SCS (ตารางที่ 3.1) หา

พจน์ส่วนเกิน โดยมีการพิจารณาการเก็บ

กักความชื้น (ตารางที่ 5.1)

$$Q_i = Q_{i-1} + P_i - F_i - I_{ai} - ET_i$$

* กฎการอนุรักษ์โมเมนตัมและพลังงาน

เส้น Isochrones ที่สังเคราะห์

ขึ้นจากเวลาเดินทางของการไหล

ของน้ำผ่าน grid cell แสดง

ถึงการใช้หลักการ momentum

และ energy

เวลาเดินทางของการไหลของน้ำ

สมการที่ 3.1 และ 3.3 เป็นสมการที่มี

พื้นฐานมาจากสมการ momentum และ

energy

* การประยุกต์ใช้

การทำนายชลภาพน้ำท่าของกลุ่มน้ำ

* ข้อมูล/ผลลัพธ์

ข้อมูล ปริมาณฝน, คุณสมบัติทางภูมิ

ศาสตร์, คุณสมบัติทางกายภาพของกลุ่มน้ำ

ผลลัพธ์ ชลภาพน้ำท่าของกลุ่มน้ำ

ภาคผนวก ค.

โปรแกรมคอมพิวเตอร์

```

C
C      GISENTRY.FOR
C      Grid cell data entry for GIS programs
C
C      This program allows the user to enter values by row and
C      column for grid cell GIS systems. It includes facilities for
C      interrupting entry and then resuming later, also for entering
C      runs of same value.
C
C
C      LAST UPDATE      15/JUN/86
C
C      INTEGER * 4 IROW,ISCOL,IECOL,IVALUE,IREC,NREC,NNREC,ICODE(9000)
C      LOGICAL FIRST
C      CHARACTER * 1 TITLE(80),FMAT(80),FMAT1(80),BLANK(80),IBUF(80)
C      CHARACTER*32 INFILE, OUTFILE
C      DATA BLANK /80*' '/
C      DATA FMAT/('','1','6','I','5','/','1','6','I','5','/','1','6','I',
1      '5','/','1','6','I','5','/','1','6','I','5','/','1','6',
2      'I','5','/','1','6','I','5','/','1','6','I','5','/','1',
3      '6','I','5','/','5','I','5','')',30*' '/
C      DATA FMAT1/ ('','1','X','','I','3','','1','X','','I','3','','',
1      '1','X','','I','3','')',61*' '/
2222 FORMAT(1X,'GISENTRY - Make data base creation file for PC MAP ')
2223 FORMAT(1X,'Remote Sensing Center, Asian Institute of Technology')
2224 FORMAT(1X,'Version 2.0   June 1986'//)
      WRITE(*,2222)
      WRITE(*,2223)
      WRITE(*,2224)

2 CONTINUE
      FIRST = .TRUE.
3 FORMAT(/3X,'NAME OF OUPUT FILE TO BE CREATED? ')
      WRITE(*,3)
4 FORMAT(A32)
      READ(*,4,ERR=2,END=2) OUTFILE
      OPEN(15,FILE=OUTFILE,STATUS='NEW')
8 FORMAT(16I5/16I5/16I5/16I5/16I5/16I5/16I5/
1 16I5/5I5)
9 FORMAT(1X,I3,1X,I3,1X,I3)
10 CONTINUE
      WRITE(*,11)
11 FORMAT (/3X,'ENTER RECORD NUMBER TO START WITH. IF THIS IS THE',
1 /6X, 'FIRST TIME YOU HAVE ENTERED DATA FOR THIS STUDY AREA,',
1 /6X, 'ENTER RECORD NUMBER 1 (INTEGER): ')
      READ(*,*,ERR=10,END=10) IREC
      IF (IREC .EQ. 0) GO TO 10
      IF (IREC .EQ. 1) GO TO 20
1002 CONTINUE
1003 FORMAT(/3X,'NAME OF FILE HOLDING PREVIOUSLY ENTERED DATA? ')
      WRITE(*,1003)
      READ(*,4,ERR=1002,END=1002) INFILE
      OPEN(14,FILE=INFILE,STATUS='OLD')
CCC      IREC = IREC - 1

```

```

C      eliminated 6-15-96 - doesn't seem to be necessary
      DO 15 M = 1,IREC
        READ(14,12,END=16,ERR=16) (IBUF(I), I=1,80)
        WRITE(15,12,ERR=16) (IBUF(I), I=1,80 )
12      FORMAT(80A1)
15      CONTINUE
16      CLOSE(14)
20      CONTINUE
        WRITE(*,21)
21      FORMAT (/3X,'HOW MANY ROWS AND COLUMNS IN YOUR DATA SET ',
1 /6X, '(INTEGER,INTEGER)? ')
        READ(*,*,ERR=20,END=20) ITROW,ITCOL
        IF (ITROW .EQ. 0 .OR. ITCOL .EQ. 0) GO TO 20
        ITEM = (ITCOL/16) + 1
CCC     IF (ITEM .NE. 0) ITEM = (ITCOL/16) + 1
        MM = ITCOL + 1
22      CONTINUE
        NREC = 0
        IF (IREC .GT. 1 .AND. FIRST ) THEN
          WRITE(*,2300)
2300      FORMAT(/3X,'WANT TO CONTINUE CODING THE SAME MAP LAYER',
1 /6X,'THAT YOU WERE WORKING ON BEFORE YOU STOPPED (Y OR N)?')
          READ(*,12) TITLE(1)
          IF (ICHAR(TITLE(1)).EQ.89.OR.ICHAR(TITLE(1)).EQ.121) THEN
C            skip title query and format printing if same layer as before
              IIREC = 0
              NREC= -1
C              ugh - but we need a flag
              GOTO 27
          ENDIF
        ENDIF
C
        WRITE(*,23)
23      FORMAT(/3X,'ENTER NAME FOR NEXT MAP LAYER TO BE CODED. '
1 /6X,'(MAXIMUM 32 CHARACTERS):')
        READ(*,12,ERR=22,END=22) (TITLE(M),M=1,32)
        IIREC = 0
27      CONTINUE
        FIRST = .FALSE.
        WRITE(*,28)
28      FORMAT(/3X,'ENTER 1 TO CODE GRID DATA, 2 TO CODE POINT DATA '
1 /6X,'(USE POINT DATA FOR ROADS, ETC.) (INTEGER): ')
        READ(*,*,ERR=27,END=27) IANS
        IF (IANS .LT. 1 .OR. IANS .GT. 2) GO TO 27
        IF (IIREC .EQ. 1) GO TO 30
        IF (NREC .EQ. -1) THEN
          NREC = 0
          GOTO 30
        ENDIF
C      skip format and command if we are continuing
        IF (IANS .EQ. 1) THEN
          WRITE(15,129) (TITLE(M),M=1,32)
          WRITE(15,12) (FMAT(I),I=1,80)
        ELSE IF (IANS .EQ. 2) THEN
          WRITE(15,229) (TITLE(M),M=1,32)

```

```

        WRITE(15,12) (FMAT1(I),I=1,80)
    ENDIF
129 FORMAT(1X,'GRID FOR ',32A1)
229 FORMAT(1X,'POINT FOR ',32A1)
    NREC = NREC + 2
30 CONTINUE
    IF (IREC .NE. 1) NREC = NREC + IREC
    IF (IANS .EQ. 2) GO TO 100

C
C     FOR GRID CODING....
C
38 CONTINUE
    WRITE(*,39)
39 FORMAT (//3X,'ENTER ROW NUMBER (INTEGER) ',/6X,
1 '(ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER',/6X
2 ' FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER) ')
    READ(*,*,ERR=38,END=38) IROW
    DO 4000 I = 1,MM
        ICODE(I)=0
4000 CONTINUE
        IF (IROW .EQ. 0) GO TO 70
        IF (IROW .EQ. -1) GO TO 60
        IF (IROW .LE. ITROW) GO TO 41
        WRITE(*,40)
40 FORMAT (/3X,'*** ROW NUMBER TOO BIG!! TRY AGAIN! ')
        GO TO 38
41 ICODE(1) = IROW
42 CONTINUE
        WRITE(*,43) IROW
43 FORMAT(/3X,'ROW ',I3,' : ENTER STARTING AND ENDING COLUMN NUMBERS,'
1 /6X,'AND VALUE TO ASSIGN TO THIS RANGE ',/6X,
1 '(ENTER 0,0,0 TO CHOOSE NEW ROW)',/6X,
2 '(INTEGER, INTEGER, INTEGER): ')
        READ(*,*,ERR=42,END=42) ISCOL,IECOL,IVALUE
        IF (ISCOL .LT. 1) GOTO 55
        IF (IECOL .GE. ISCOL .AND. IECOL .LE. ITCOL ) GOTO 48
        IF (IECOL .GT. ITCOL) WRITE(*,44)
44 FORMAT (/3X,'**** ENDING COLUMN NUMBER TOO BIG!! TRY AGAIN! ')
        IF (IECOL .LT. ISCOL) WRITE(*,45)
45 FORMAT (/3X,'**** INCONSISTENT COLUMN NUMBERS!! TRY AGAIN! ')
        GO TO 42
48 CONTINUE
49 FORMAT(/3X,'**** CODE VALUE OUT OF RANGE!! TRY AGAIN! ')
        IF (IVALUE .GT. 32000 .OR. IVALUE .LT. -32000 ) THEN
            WRITE(*,49)
            GOTO 42
        ENDIF
        DO 50 K = ISCOL,IECOL
            ICODE(K+1) = IVALUE
50 CONTINUE
        GOTO 42
55 WRITE(15,8) (ICODE(I), I=1,MM)
        NREC = NREC + ITEM
        GO TO 38
60 CONTINUE

```

```

NREC = NREC + ITEM
JJ = ITEM - 1
WRITE(15,62) IROW
62 FORMAT(I2)
DO 65 J = 1,JJ
    WRITE(15,12) (BLANK(I),I=1,80)
65 CONTINUE
66 CONTINUE
    WRITE(*,67)
67 FORMAT(/3X,'ENTER 1 TO CODE NEW ENTITY, 2 TO STOP (INTEGER): ')
    READ(*,*,ERR=66,END=66) IANS
    IF (IANS .LT. 1 .OR. IANS .GT. 2) GO TO 66
    IF (IANS .EQ. 2) GO TO 74
    IREC = NREC
    GO TO 22
70 CONTINUE
74 WRITE(*,75) NREC
75 FORMAT (/3X,'PLEASE REMEMBER THE LAST RECORD NUMBER ',
1 /6X,'IF YOU PLAN TO CONTINUE DATA ENTRY LATER. ',
1 //6X,'*** THE LAST RECORD NUMBER WAS: ',I4)

    GO TO 999

C
C   FOR POINT CODING....
C
100 CONTINUE
101 CONTINUE
    WRITE(*,102)
102 FORMAT (/3X,'ENTER ROW NUMBER, COLUMN NUMBER, AND DATA CODE ',
1 /6X,'(INTEGER,INTEGER,INTEGER)',
1 /6X,'(ENTER 0,0,0 TO STOP FOR A WHILE AND RESUME ENTRY LATER',
2/6X,'FOR CURRENT LAYER; -1,0,0 TO COMPLETE CURRENT LAYER.)')
    READ(*,*,ERR=101,END=101) IROW,ISCOL,IVALUE
    IF (IROW .EQ. 0) GOTO 70
    IF (IROW .EQ. -1) GO TO 103
    IF (IROW .GT. ITROW .OR. ISCOL .GT. ITCOL .OR. IROW .LT. 1 .OR.
1 ISCOL .LT. 1 ) THEN
1045    FORMAT (/3X,'**** INVALID ROW OR COLUMN NUMBER!! TRY AGAIN! ')
        WRITE(*, 1045)
        GOTO 101
    END IF
    IF (IVALUE .GT. 32000 .OR. IVALUE .LT. -32000 ) THEN
        WRITE(*,49)
        GOTO 101
    ENDIF
    NREC = NREC + 1
    WRITE(15,9) IROW,ISCOL,IVALUE
C   NOTE: Mainframe version wrote in the wrong order!! col then row !!
    GO TO 101
103 CONTINUE
    NREC = NREC + ITEM
    WRITE(15,62) IROW
    JJ = ITEM - 1
    DO 1065 J = 1,JJ
        WRITE(15,12) (BLANK(I),I=1,80)

```

```
1065 CONTINUE  
      GO TO 66  
998  FORMAT(1X,'READ ON 5 ')  
999  WRITE(15,998)  
      CLOSE(15)  
      END
```

Program MAP

```

common/ref/  mname, mapnam
COMMON/SAM/  NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER      BUF(2000),W
integer*2    ia(32640),ib(32640)
character*1   mapnam(11,160), title(8), c1, c2
character*2   name, mname
character*4   catnam(4,120)
character*12  fname
data         fname/'      .-  '/

open (1,file='titles.tmp',status='old')
C  SKIP OVER THE FIRST FEW LINES ABOVE THE LIST OF OVERLAY TITLES
do 020 n = 1, 4
020      read(1,030)
030      format(8a1,2x,a2)

C  SEARCH THROUGH TITLES FOR MAP NAMES
max = 0
do 050 n = 1, 9999
    read(1,030,end=060) (title(mm),mm=1,8), name

C  COMPARE EACH NEW NAME TO ALL PREVIOUS NAMES
do 040 m = 1, max
    if(name(1:1).ne.mapnam(1,m).or.
1      name(2:2).ne.mapnam(2,m)) go to 40
        ia(m) = ia(m) + 1
        go to 50
040      continue

C  RECORD ANY NAME NOT YET ENCOUNTERED
max      = max + 1
mapnam(1,max) = name(1:1)
mapnam(2,max) = name(2:2)
do 045 mm = 1, 8
045      mapnam(mm+2,max) = title(mm)
        ia(max)      = 1
050      continue
060      rewind (1)
max = max - 1

C  LIST AVAILABLE MAP NAMES
if (max.eq.0) then
065      write (0,065)
        format(' (SINCE NO MAPS ARE ON FILE, '/
1          ' ONE WILL HAVE TO BE CREATED)')//)
    else
        write (0,069)
069      format(22(/),' AVAILABLE DATA:  MAP NAME  # OF OVERLAYS'//)
        do 070 m = 1, max
070          write(0,080) (mapnam(n,m),n=1,2), ia(m)
080          format(24x,2a1,4x,i3/)
endif

```

```

C      SELECT MAP NAME
090    write(0,100)
100    format(// ' PLEASE ENTER MAP NAME: '//)
      read (0,110) c1, c2
110    format(2a1)
      nc1 = ichar(c1)
      nc2 = ichar(c2)
      if(nc1.eq.32.or.nc2.eq.32) go to 090
      if(nc1.ge.97.and.nc1.le.122) c1 = char(nc1-32)
      if(nc2.ge.97.and.nc2.le.122) c2 = char(nc2-32)
      name(1:1) = c1
      name(2:2) = c2
      mname = name

C      TRY TO FIND THE REQUESTED MAP
      do 120 m = 1, max
          if(mname(1:1).ne.mapnam(1,m).or.
1          mname(2:2).ne.mapnam(2,m)) go to 120

C      IF MAP EXISTS, GET ITS DIMENSIONS FROM 1ST OVERLAY
      close (1)
      do 115 mm = 1, 8
115      fname(mm:mm) = mapnam(mm+2,m)
          fname(11:12) = mname
          open (1,file=fname,status='old',
1          access='direct',recl=4)
          read (1,rec=449) nr
          read (1,rec=450) nc
          read (1,rec=451) nwide
          close (1)
          go to 180

120      continue

C      OR IF THE REQUESTED MAP DOES NOT EXIST, GET ITS DIMENSIONS FROM
      close (1,status='delete')
      write(0,130) mname
130    format(/ ' ',A2,' " MUST BE NEW MAP'//
1      ' HOW MANY (horizontal) ROWS SHOULD IT HAVE?'//)
      read (0,*) nr
      n = 32640 / nr
150    write (0,160) n
160    format(/ ' HOW MANY (vertical) COLUMNS? (no more than ',i5,')'//)
      read (0,*) nc
      if (nr*nc.gt.32640) go to 150
      write (0,170)
170    format(/ ' HOW WIDE (IN GEOGRAPHIC UNITS SUCH AS FEET ',
1      ' OR METERS) IS EACH ROW OR COLUMN?'//)
      read (0,*) nwide
      m = 0
      goto 275

C      DESCRIBE AND THE CURRENT MAP AND LOAD ITS OVERLAY TITLES
180    write(0,190) mname,nr,nc,nwide
190    format(// ' (LOADING ',a2,' OVERLAYS)'//)

```

```

1          20x,'      Number of Rows: ',i5/
1          20x,' Number of Columns: ',i5/
1          20x,'      Standard Scale: ',i5///
1          20x,'      Overlay      Protection'/
1          20x,'      Title      Scale      Status'//
open (1,file='titles.tmp',status='old')
do 200 n = 1, 4
200      read(1,030)
      m = 0
do 260 n = 1, 9999
      read(1,030,end=270) (title(mm),mm=1,8), name
      if (name.ne.mname) go to 260
          m = m + 1
          do 240 mm = 1,8
240              mapnam(mm,m) = title(mm)
              call iotit (m, 1)
              iscale = (256 * ichar(mapnam(11,m)))
                  + ichar(mapnam(10,m))
1              write(0,250) title, iscale, mapnam(9,m)
250              format(25x,8a1,3x,i5,3x,a1)
260          continue
270      close (1,status='delete')
275      nc1      = nwide / 256
      nc2      = nwide - (256*nc1)
      c1      = char(nc1)
      c2      = char(nc2)
do 290 n = m+1, 160
      do 280 mm = 1,8
280          mapnam(mm,n) = char(32)
          mapnam(09,n) = 'E'
          mapnam(10,n) = c2
290          mapnam(11,n) = c1
      NP      = 0
      NI      = 0
      NCL     = nc * nr
      NMAPS   = 160
      NBUF    = 2000
      JC      = NC
      JR      = NR
      JCL     = ( ( ( nc1-1 )/960 )+1 )*960
      JMAPS   = NMAPS

c      COPY OVERLAY(S), IF ANY, FROM PREVIOUS MAP
      open (3,file='copies.tmp',status='old',err=299)
      read (3,*) ic, ir, icl, irow, lrow, icol, lcol
      write (np,292)
292      format(///' (COPYING DATA FROM PREVIOUS MAP)'///' PLEASE',
1          ' IDENTIFY THE AREA INTO WHICH DATA SHOULD BE COPIED'//
1          ' TOP ROW NUMBER ...'//)
      read (ni,*) iro
      write (np,293)
293      format('/' BOTTOM ROW NUMBER ...'//)
      read (ni,*) lro
      write (np,294)
294      format('/' LEFT COLUMN NUMBER ...'//)

```

```

read (ni,*) ico
write (np,295)
295 format(/' RIGHT COLUMN NUMBER ...'//)
read (ni,*) lco
do 297 m = 2, 40
    read (3,*,err=298,end=298) fname
    do 2296 mm = 1, 8
        title(mm) = fname(mm:mm)
        if (title(mm).eq.char(46)) then
            if (mm.eq.1) goto 298
            do 2295 mmm = mm, 8
                title(mmm) = char(32)
2295 mmm = mm
                goto 2297
            endif
2296 continue
        mmm = 9
2297 name(1:2) = fname(mmm+2:mmm+3)
        write (np,2298) title, name, mname
2298 format(//' (COPYING OVERLAY "' ,8a1,'" FROM MAP "' ,a2,
1      '" TO MAP "' ,a2,'"')//)
        errors = 0
        call newmap (jmaps,mapno,title,errors)
        if (errors.eq.0) then
            open(1,file=fname,status='old',access='direct',recl=1920)
            nppb = 960
            nblock = 1
            do 296 n = 1, icl, 960
                nblock = nblock + 1
                if(n+nppb.gt.icl) nppb = icl + 1 - n
296 call ioddata(1,nppb,nblock,ia(n))
            read (1,rec=1) catnam
            close (1)
            call regrid (ic,ir,icl,jc,jr,jcl,1,irow,icol,lrow,lcol,
1      0,1,iro,ico,lro,lco,ia,ib)
            call file (jcl, mapno, 2, ib)
            call iocat (mapno, 2, catnam)
            endif
297 continue
298 close (3,status='delete')

299 write(0,300)
300 format(20(//
1  /'                               The Map Analysis Package'//
1  /'                               developed by'
1  /'                               C. Dana Tomlin'/
1  /'       Laboratory for Computer Graphics and Spatial Analysis'
1  /'       Harvard Graduate School of Design'
1  /'       Cambridge, Massachusetts USA'////
1  /'       (C) Copyright 1986'
1  /'       by The President and Fellows of Harvard College'
1  /'       All Rights Reserved'
1  ////' OK, MAP ANALYSIS PACKAGE READY')
CALL RECOM (JC,JR,JCL,JMAPS,CATNAM,IA,IB)
WRITE (NP,400)

```

```

400  FORMAT (///,' OK, MAP ANALYSIS PACKAGE STOPPED',18(/))
      stop 'Returning from MAP to DOS'
      end

      subroutine file (jcl, mapno, io, ia)
      common/ref/      mname, mapnam
      integer*2        ia(jcl)
      character*1       mapnam(11,160)
      character*2       mname
      character*12      fname
      data              fname/'          .-  '/
      fname(11:12) = mname
      do 299 n = 1, 8
299          fname(n:n) = mapnam(n,mapno)
      if (io.ge.2) then
          open(1,file=fname,status='unknown',access='direct',recl=1920)
      else
          open(1,file=fname,status='old',access='direct',recl=1920)
      endif
      nppb = 960
      nblock = 1
      do 301 n = 1, jcl, 960
          nblock = nblock + 1
          if(n+nppb.gt.jcl) nppb = jcl + 1 - n
301          call iodata(io,nppb,nblock,ia(n))
      close(1)
      return
      end

      subroutine iodata (io,nppb,nblock,block)
      integer*2 block(nppb)
      if(io.ge.2) then
          write(1,rec=nblock) block
      else
          read(1,rec=nblock) block
      endif
      return
      end

      subroutine iocat (mapno, io, legend)
      common/ref/      mname, mapnam
      character*1       mapnam(11,160), legend(1600)
      character*2       mname
      character*12      fname
      data              fname/'          .-  '/
      fname(11:12) = mname
      do 299 n = 1, 8
299          fname(n:n) = mapnam(n,mapno)
      if (io.ge.2) then
          open(1,file=fname,status='unknown',access='direct',recl=1600)
          write (1,rec=1) legend
      else
          open (1,file=fname,status='old',access='direct',recl=1600)
          read (1,rec=1) legend
      endif

```

```

close(1)
return
end

subroutine iotit (mapno,io)
common/ref/      mname, mapnam
COMMON/SAM/      NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
character*1      mapnam(11,160)
character*2      mname
character*12     fname
data             fname/'      .-  '/
fname(11:12) = mname
do 100          n = 1, 8
100             fname(n:n) = mapnam(n,mapno)
if (io.ge.2) then
  open (2,file=fname,status='unknown',access='direct',recl=1)
  write(2,rec=1601) mapnam(09,mapno)
  write(2,rec=1665) mapnam(10,mapno)
  write(2,rec=1666) mapnam(11,mapno)
  close(2)
  open (2,file=fname,status='unknown',access='direct',recl=4)
  write(2,rec=449)  nr
  write(2,rec=450)  nc
  write(2,rec=451)  nwide
  close(2)
else
  open (2,file=fname,status='old',access='direct',recl=1)
  read (2,rec=1601) mapnam(09,mapno)
  read (2,rec=1665) mapnam(10,mapno)
  read (2,rec=1666) mapnam(11,mapno)
  close(2)
endif
return
end

```

```

SUBROUTINE PROJEC(HORCOS,HORSIN,VERCOS,VER SIN,RANGE,
1 DIS,OLDX,OLDY,OLDZ,NEWX,NEWY)

```

C
C
C

```

SUBROUTINE PROJEC

```

```

IMPLICIT INTEGER*2 (A-Z)
INTEGER RANGE,DIS,OLDX,OLDY,OLDZ,NEWX,NEWY
REAL HORCOS,HORSIN,VERCOS,VER SIN,X,Y,Z,N
X = OLDX
Y = OLDY
Z = OLDZ
N = X
X = (X*HORCOS) - (Y*HORSIN)
Y = (N*HORSIN) + (Y*HORCOS)
N = Y
Y = (Y*VERCOS) - (Z*VER SIN)
Z = (N*VER SIN) + (Z*VERCOS) - DIS
NEWX = X / Z * DIS
NEWY = Y / Z * DIS
RETURN
END

```

```

SUBROUTINE SHOW (NCOLS,NROWS,IB,IA)

```

C
C
C

```

SUBROUTINE SHOW

```

```

IMPLICIT INTEGER*2 (A-Z)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 LETTER
REAL HORCOS, HORSIN, VERCOS, VER SIN, ANGLE
INTEGER NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB,
1 RANGE, DIS, SIZE, VERTIC, MIDX, MIDY, MIDZ,
2 TROW, TCOL, XSHIFT, YSHIFT, NEWX, NEWY,
3 OLDX, OLDY, OLDZ, NCOLS, NROWS
DIMENSION WORKIN(19),WORKOT(66),XY(4),COLOR(4),
1 IA(NCOLS,NROWS),IB(NCOLS,NROWS)
DATA WORKIN /1,1,1,3,1,1,1,0,0,0,1,68,73,83,80,76,65,89,32/
DATA COLOR /2,3,1,0/
WRITE(NP,105)
105 FORMAT(/3X,'*** This Routine requires GRAPHIC options ***'
1 /3X,'So cannot use at this time !!!')
RETURN
106 CONTINUE
NBYTES = NROWS * NCOLS * 2
SCALE = NWIDE
107 ROTATE = 15
TILT = 30
TROW = NROWS / 2
TCOL = NCOLS / 2
Distan = 100
VERTIC = 100
SIZE = 100
XSHIFT = 16384
YSHIFT = 11000

```

```

DO 300 NN = 1,999
  WORKIN(12) = 68
  WORKIN(13) = 73
  WORKIN(14) = 83
  WORKIN(15) = 80
  WORKIN(16) = 76
  WORKIN(17) = 65
  WORKIN(18) = 89
  WORKIN(19) = 32
  WRITE (NP,702) ROTATE,TILT,TROW,TCOL,DISTAN,VERTIC,SIZE
702  FORMAT (////,
1    ' CLOCKWISE ROTATION ANGLE:  A = ',I3,' DEGREES',
1  //'          TILT-BACK ANGLE:  T = ',I3,' DEGREES',
1  //'          TARGET ROW:      R = ',I3,
1  //'          TARGET COLUMN:   C = ',I3,
1  //'          VIEWING DISTANCE: D = ',I3,' GRID SPACES',
1  //'  VERTICAL SCALING FACTOR:  V = ',I3,' PERCENT',
1  //'          SIZE FACTOR:     S = ',I3,' PERCENT',
1  //' TO CHANGE ANY OF THESE, ENTER ITS LETTER',
1  //' FOLLOWED IMMEDIATELY BY A 3-DIGIT NUMBER AS IN R045.',
1  //' THEN TO BEGIN DRAWING, ENTER P FOR PRINTER OUTPUT'
1  //' OR JUST PRESS THE ENTER KEY ALONE FOR SCREEN OUTPUT.'
1  //' TO QUIT, ENTER Q.'//)
700  READ (NI,701) LETTER,N
701  FORMAT (A1,I3)
    IF (LETTER.NE.'A' .AND. LETTER.NE.'a') GO TO 721
    ROTATE = N
    GO TO 700
721  IF (LETTER.NE.'P' .AND. LETTER.NE.'p') GO TO 722
    WORKIN(12) = 80
    WORKIN(13) = 82
    WORKIN(14) = 73
    WORKIN(15) = 78
    WORKIN(16) = 84
    WORKIN(17) = 69
    WORKIN(18) = 82
    WORKIN(19) = 32
    GO TO 750
722  IF (LETTER.NE.'T' .AND. LETTER.NE.'t') GO TO 724
    TILT = N
    IF (TILT.LT.0 .OR. TILT.GE.90) TILT = 40
    GO TO 700
724  IF (LETTER.NE.'D' .AND. LETTER.NE.'d') GO TO 726
    DISTAN = N
    IF (DISTAN .LE. 0) DISTAN = 100
    GO TO 700
726  IF (LETTER.NE.'V' .AND. LETTER.NE.'v') GO TO 728
    VERTIC = N
    IF (VERTIC .LE. 0) VERTIC = 100
    GO TO 700
728  IF (LETTER.NE.'S' .AND. LETTER.NE.'s') GO TO 730
    SIZE = N
    IF (SIZE .LE. 0) SIZE = 100
    GO TO 700
730  IF (LETTER.NE.'C' .AND. LETTER.NE.'c') GO TO 732

```

```

TCOL = N
IF (TCOL.LE.0 .OR. TCOL.GT.NCOLS) TCOL = NCOLS/2
GO TO 700
732 IF (LETTER.NE.'R' .AND. LETTER.NE.'r') GO TO 734
TROW = N
IF (TROW.LE.0 .OR. TROW.GT.NROWS) TROW = NROWS/2
GO TO 700
734 IF (LETTER.NE.'Q' .AND. LETTER.NE.'q') GO TO 736
RETURN
736 IF (LETTER.NE.' ') GO TO 700
750 MIDX = TROW * SCALE
MIDY = TCOL * SCALE
MIDZ = IA(TCOL,TROW) * VERTIC / 100
ANGLE = (450 - ROTATE) * 0.01745329
HORCOS = COS(ANGLE)
HORSIN = SIN(ANGLE)
ANGLE = TILT * 0.01745329
VERCOS = COS(ANGLE)
VERSIN = SIN(ANGLE)
M = DISTAN * 4 / 5
RANGE = DISTAN * SCALE
DIS = M * SCALE
C OPEN AND CLEAR WORKSTATION
C STATUS = VOPNWK (WORKIN(1),DEVICE,WORKOT(1))
C IF (STATUS.NE. 0) THEN
C STATUS = VQERR()
C WRITE(NP,99999) STATUS
C99999 FORMAT(' VOPNWK ERROR NUMBER ',I5)
C ENDIF
C STATUS = VCLRWK(DEVICE)
C IF (STATUS.NE. 0) THEN
C STATUS = VQERR()
C WRITE(NP,99998) STATUS
C99998 FORMAT(' VCLRWK ERROR NUMBER ',I5)
C ENDIF
C DRAW BORDERS
TOP = 22000
BOTTOM = 0
LEFT = 0
RIGHT = 32767
XY(1) = LEFT
XY(2) = TOP
XY(3) = RIGHT
XY(4) = TOP
C STATUS = VPLINE(DEVICE,2,XY)
C IF (STATUS.NE. 0) THEN
C STATUS = VQERR()
C WRITE(NP,99997) STATUS
C99997 FORMAT(' VPLINE ERROR NUMBER',I5)
C ENDIF
XY(1) = RIGHT
XY(2) = BOTTOM
C STATUS = VPLINE(DEVICE,2,XY)
XY(3) = LEFT
XY(4) = BOTTOM

```

```

C      STATUS = VPLINE(DEVICE,2,XY)
C      XY(1)   =      LEFT
C      XY(2)   =      TOP
C      STATUS = VPLINE(DEVICE,2,XY)
C SET UP FOR TOP ROW
C      EDGE = -SCALE / 2
C      OLDX = EDGE - MIDX
C LOOP DOWN THROUGH ROWS
C      DO 200 ROW = 1,NROWS
C ESTABLISH X COORDINATE FOR THE WHOLE ROW
C      OLDX = OLDX + SCALE
C SET UP FOR LEFT COLUMN OF THE ROW
C      OLDY = EDGE - MIDY + SCALE
C      OLDZ = IA(1,ROW) * VERTIC / 100
C      CALL PROJEC(HORCOS,HORSIN,VERCOS,VER SIN,RANGE,
1          DIS,OLDX,OLDY,OLDZ,NEWX,NEWY)
C      XY(1) = ((NEWX * SIZE) / 100) + XSHIFT
C      XY(2) = ((NEWY * SIZE) / 100) + YSHIFT
C LOOP ACROSS REMAINING COLUMNS OF THE ROW
C      DO 200 COL = 2,NCOLS
C      OLDY = OLDY + SCALE
C      OLDZ = IA(COL,ROW) * VERTIC / 100
C REGARD LAST "FROM" POINT AS NEXT "TO" POINT
C      XY(3) = XY(1)
C      XY(4) = XY(2)
C DETERMINE SCREEN COORDINATES OF NEXT POINT
C      CALL PROJEC(HORCOS,HORSIN,VERCOS,VER SIN,RANGE,
1          DIS,OLDX,OLDY,OLDZ,NEWX,NEWY)
C      XY(1) = ((NEWX * SIZE) / 100) + XSHIFT
C      XY(2) = ((NEWY * SIZE) / 100) + YSHIFT
C DRAW LINE FROM A POINT TO ITS RIGHT NEIGHBOR
C      STATUS = VSLCOL(DEVICE,COLOR(IB(COL,ROW)+1))
C      IF(XY(1).GT.0.AND.XY(1).LT.32737.AND.
C      1      XY(2).GT.0.AND.XY(2).LT.22000.AND.
C      1      XY(3).GT.0.AND.XY(3).LT.32737.AND.
C      1      XY(4).GT.0.AND.XY(4).LT.22000)
C      1      STATUS = VPLINE(DEVICE,2,XY)
200      CONTINUE
C UPDATE AND CLOSE WORKSTATION
C      STATUS = VUPDWK(DEVICE)
C      STATUS = VENCUR(DEVICE)
C      STATUS = VCLSWK(DEVICE)
300      CONTINUE
C      RETURN
C      END

SUBROUTINE DRAIN (JCL, JC, JR, IV, IU, IA, IB)
C
C      S U B R O U T I N E   D R A I N
C
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON NEXTL,NEXTK
INTEGER*2 INDEX, SHARE, SPOUTS, DITEST, FRONT, REAR,
1      NEXTL(1000), NEXTK(1000), IA(JC,JR), IB(JC,JR),
1      DICODE(3,3), KSHIFT(8), LSHIFT(8), MAXK(8), MAXL(8)

```

```

      REAL      SLOPE, MAX, SLOP(8)
      DATA     DICODE /1,2,4,8,0,16,32,64,128/,
1      KSHIFT  /1,0,-1,1,-1,1,0,-1/,
1      LSHIFT  /1,1,1,0,0,-1,-1,-1/
C IA = ALTITUDE THEN ACCUMULATION; IB = DRAINAGE INDICATOR
C IN IB, RECORD DRAINAGE INFO DERIVED FROM ALTITUDE VALUES IN IA
      CALL FILE(JCL,IU,1,IA)
      DO 100 L = 1,JR
        DO 100 K = 1,JC
          IB(K,L) = 0
100 CONTINUE
C DETERMINE DIRECTION(S) OF EACH CELL'S DOWNSTREAM NEIGHBOR(S)
      DO 300 L = 1,JR
        DO 300 K = 1,JC
          LTOP = L - 1
          IF (LTOP .LT. 1)      LTOP = 1
          LBOT = L + 1
          IF (LBOT .GT. JR)     LBOT = JR
          KLEFT = K - 1
          IF (KLEFT .LT. 1)     KLEFT = 1
          KRIGHT = K + 1
          IF (KRIGHT .GT. JC)   KRIGHT = JC
          MAX = 0.0
          INDEX = 0
          DO 200 LL = LTOP, LBOT
            DO 200 KK = KLEFT, KRIGHT
              IF (IA(KK,LL) .GE. IA(K,L)) GO TO 200
              SLOPE = 100.0
              SLOPE = SLOPE*(IA(K,L) - IA(KK,LL)) / NWIDE
              IF (LL.NE.L .AND. KK.NE.K) SLOPE = SLOPE / 1.4142
              IF (SLOPE .LT. MAX) GO TO 200
              IF (SLOPE .GT. MAX) INDEX = 0
              INDEX = INDEX + 1
              MAXK(INDEX) = KK
              MAXL(INDEX) = LL
            MAX = SLOPE
          200 CONTINUE
C ENCODE DRAINAGE INFO FOR EACH CELL AS AN IB VALUE COMPUTED AS:
C # OF UPSTREAM NEIGHBORS * 2048 + DOWNSTREAM NEIGHBORS * 256
C + 128, 64, 32, 16, 8, 4, 2, AND/OR 1, RESPECTIVELY, FOR
C SE, S, SW, E, W, NE, N, AND NW DOWNSTREAM NEIGHBORS
          IF (INDEX .LE. 0) GO TO 300
          DO 250 N = 1, INDEX
            KK = MAXK(N)
            LL = MAXL(N)
            IB(KK,LL) = IB(KK,LL) + 2048
            IB(K,L) = IB(K,L) + 256 + DICODE(2+KK-K,2+LL-L)
          250 CONTINUE
        300 CONTINUE
C BEGIN THE DRAINAGE PROCESS
      CALL FILE (JCL,IV,1,IA)
C LOAD QUEUE WITH HEADWATER CELLS (IE HAVING NO UPSTREAM NEIGHBORS)
      INDEX = 0
      500 REAR = 0
      DO 600 L = 1,JR

```

```

DO 600 K = 1, JC
  IF (IB(K,L) .GE. 2048) GO TO 600
  REAR = REAR + 1
  NEXTK(REAR) = K
  NEXTL(REAR) = L
  IF (REAR .EQ. 500) GO TO 601
600 CONTINUE
C WHEN NO MORE HEADWATER CELLS, STOP EVERYTHING
  IF (REAR .EQ. 0) RETURN
C CYCLE THRU QUEUE, DELETING OLD HEADWATER CELLS FROM THE FRONT
C AS THEY ARE POURED DOWNSTREAM AND ADDING NEW ONES TO THE REAR
601 FRONT = 1
C GET THE COORDINATES OF THE NEXT HEADWATER CELL
700 K = NEXTK(FRONT)
  L = NEXTL(FRONT)
  INDEX = INDEX + 1
  IF (FRONT .EQ. 1) WRITE (NP,701) INDEX
701 FORMAT(1X, '(', I5, ' POINTS DRAINED SO FAR)')
C DETERMINE HOW MUCH GETS POURED TO EACH DOWNSTREAM NEIGHBOR
  SPOUTS = IB(K,L) / 256
  IF (SPOUTS .EQ. 0) GO TO 810
  SHARE = IA(K,L) / SPOUTS
  IF (SHARE .EQ. 0 .AND. IA(K,L) .GT. 0) SHARE = 1
C ITERATIVELY DECODE DRAINAGE INFO TO LOCATE DOWNSTREAM NEIGHBORS
  IB(K,L) = IB(K,L) - (256*SPOUTS)
  DITEST = 128
  DO 800 N = 1, 8
    IF (IB(K,L) .LT. DITEST) GO TO 800
C DOWNSTREAM NEIGHBOR FOUND; GET ITS COORDINATES
    IB(K,L) = IB(K,L) - DITEST
    KK = K + KSHIFT(N)
    LL = L + LSHIFT(N)
C ADD TO NEIGHBOR'S ACCUMULATION WITHOUT EXCEEDING 32000
    MAX = IA(KK,LL) + SHARE
    IA(KK,LL) = MAX
    IF (MAX .GT. 32767.0) IA(KK,LL) = 32767
C DECREMENT THIS NEIGHBOR'S UPSTREAM NEIGHBOR-COUNT
    IB(KK,LL) = IB(KK,LL) - 2048
    IF (IB(KK,LL) .GE. 2048) GO TO 790
C AND IF NEIGHBOR BECOMES A HEADWATER CELL,
C ADD IT TO REAR OF QUEUE UNLESS FULL
    REAR = REAR + 1
    IF (REAR .EQ. 1001) REAR = 1
    IF (REAR .EQ. FRONT) GO TO 790
    NEXTK(REAR) = KK
    NEXTL(REAR) = LL
C IF ALL DOWNSTREAMS ALREADY FOUND, DON'T LOOK FOR MORE
790 IF (SPOUTS .EQ. 1) GO TO 810
    SPOUTS = SPOUTS - 1
C IF MORE DOWNSTREAMS TO FIND, KEEP DECODING
    DITEST = DITEST / 2
800 CONTINUE
C MARK THIS HEADWATER CELL EMPTY
810 IB(K,L) = 32767
C WHENEVER QUEUE BECOMES EMPTY, TRY TO RELOAD IT

```

```

      IF (FRONT .EQ. REAR) GO TO 500
C OTHERWISE, INCREMENT FRONT OF QUEUE AND DO NEXT HEADWATER CELL
      FRONT = FRONT + 1
      IF (FRONT .EQ. 1001) FRONT = 1
      GO TO 700
END

```

```

SUBROUTINE SORT(ECHOED,POINTS,IA,IB)

```

```

C
C   S U B R O U T I N E   S O R T
C
      IMPLICIT   INTEGER*2   (A-Z)
      COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
      INTEGER    NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB,
1      ECHOED, POINTS
      INTEGER*2   IA(POINTS),IB(POINTS)
C SORTS THE VALUES OF A MAP OVERLAY
C DEFINE BEGINNING OF POINTER LIST
      OFFSET = (POINTS/2)
C POINT DUMMY LIST HEAD TO DUMMY LIST TAIL
      IB(OFFSET*2) = OFFSET - 1
C SET VALUE STORED IN DUMMY LIST TAIL TO INFINITY
      IB(OFFSET-1) = 32767
      VALUE      = 32767
C SET LIST TO BEGIN AT LOCATION 1
      NEWVAL = 0
C LOOP THRU POINTS BUILDING LINKED LIST OF SORTED VALUES
      DO 500 POINT = 1, POINTS
C IGNORE ANY POINT WHOSE VALUE IS SAME AS PREVIOUS POINT
      IF (IA(POINT) .EQ. VALUE) GO TO 500
C SET UP FOR COMPARISON TO VALUES ALREADY SORTED
      VALUE = IA(POINT)
C START AT DUMMY LIST HEAD
      OLDVAL = OFFSET
C LOOP THRU LINKED LIST
300      PREVAL = OLDVAL
          OLDVAL = IB(OLDVAL+OFFSET)
C IF NEW VALUE IS GREATER THAN OLD ONE, TRY AGAIN
      IF (VALUE .GT. IB(OLDVAL)) GO TO 300
C IF NEW VALUE IS SAME AS OLD ONE, FORGET IT
      IF(VALUE .EQ. IB(OLDVAL)) GO TO 500
C IF NEW VALUE IS LESS THAN OLD ONE, ADD IT
C BY DEFINING A NEW LIST LOCATION
      NEWVAL = NEWVAL + 1
      OLDVAL = NEWVAL
      IF (NEWVAL.LT.OFFSET-2 .AND. NEWVAL.LT.16380) GO TO 350
      WRITE(NP,340)
340      FORMAT(/3X,'NO, TOO MANY VALUES')
          STOP
C BY STORING THE VALUE THERE
350      IB(NEWVAL) = IA(POINT)
C BY LINKING THIS LIST ENTRY TO THE NEXT
      IB(NEWVAL+OFFSET) = IB(PREVAL+OFFSET)
C AND BY LINKING THE PREVIOUS ENTRY TO IT
      IB(PREVAL+OFFSET) = NEWVAL

```

```
C REPLACE POINT'S VALUE WITH IT ABSOLUTE LOCATION IN LIST
    IA(POINT) = OLDVAL
500 CONTINUE
C REPLACE SORTED LIST VALUES WITH ORDINAL LIST POSITION VALUES
    OLDVAL = OFFSET
    DO 600 N = 1,NEWVAL
        OLDVAL = IB(OLDVAL+OFFSET)
        IF (ECHOED .NE. 0) WRITE(NP,555) IB(OLDVAL), N
555    FORMAT(1X,'(',I6,' = ',I5,')')
        IB(OLDVAL) = N
600 CONTINUE
C ASSIGN ORDINAL POSITION VALUES TO POINTS
    DO 700 N = 1, POINTS
        OLDVAL = IA(N)
        IA(N) = IB(OLDVAL)
700 CONTINUE
    RETURN
    END
```



```

      fname(3:5) = '.IN'
      fname(6:12) = '
open(unit=ni,file=fname,status='old',err=10)
go to 20
010  ni = 0
020  CALL RECARD(W,WORD,1,ECHOED)
      DO 60 M = 1,120
          DO 60 N = 1,4
060      CATNAM(N,M) = DOTS
      GO TO 250
150  CALL RECARD(W,WORD,1,ECHOED)
250  NV = 0
      IV = 0
      IU = 0
      IW = 0
      IX = 0
      IY = 0
      IZ = 0
      IZA = 0
      IZB = 0
      IZC = 0
      IZD = 0
      IZE = 0
      IZF = 0
      DO 290 L = 1,8
290  NEWNAM(L) = THAT(L)
      CALL SMATCH(COMNO,COM,WORD(1,1))
      IF (COMNO.GT.0) GO TO 400
          ERRORS = 1
          WRITE(NP,300) (WORD(L,1),L=1,8)
300      FORMAT(/34H NO, THERE IS NO OPERATION NAMED ",8A1,'"')
          GO TO 99500
400  ERRORS = 0
      GO TO ( 1050, 410, 410, 410, 600, 410, 7150,
1          150, 9000, 410, 600, 7150, 13050, 14000,
1          7150, 7150, 17020, 7150, 410, 7150, 7150,
1          7150, 600, 24000, 600, 600, 27000, 410,
1          410, 410, 410, 32100, 410, 410, 410,
1          410, 410, 410, 410, 7150, 410, 600,
1          43000, 410, 410, 600, 600, 600, 600,
1          410, 51000, 410, 410, 54100, 410, 410,
1          410, 410, 59000, 60000, 61000, 150), COMNO
410  W = W + 1
      CALL REMAP(JMAPS,IV,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
600  W = W + 1
700  CALL RESUB(SUBNO,COMNO,ERRORS,W,WORD)
      W = W + 1
      GO TO ( 1000, 2000, 3000, 4000, 5000, 6000, 7000,
1          150, 9000, 10000, 11000, 12000, 13000, 14000,
1          15000, 16000, 17000, 18000, 19000, 20000, 21000,
1          22000, 23000, 24000, 25000, 26000, 27000, 28000,
1          29000, 30000, 31000, 32000, 33000, 34000, 35000,
1          36000, 37000, 38000, 39000, 40000, 41000, 42000,
1          43000, 44000, 45000, 46000, 47000, 48000, 49000,

```

```

1          50000, 51000, 52000, 53000, 54000, 4050, 56000,
1          57000, 58000, 59000, 60000, 61000), COMNO
C***** BAD / ASSI FOR THRU TO DONE ***** 1000
1000 GO TO (99500,96300,1100,96000,1300,1400,99200),SUBNO
1050 START = -999
      DO 1060 N = 1,NBUF
1060 BUF(N) = N - 1000
      GO TO 410
1100 CALL RENUM(IU,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      GO TO 600
1300 CALL RENUM(STOP,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      IF (STOP.GT.1000.CR.STOP.LT.-999) GO TO 13700
      IF(STOP.GE.START) GO TO 1320
          N = START
          START = STOP
          STOP = N
1320 CONTINUE
      DO 1350 N = START,STOP
1350 BUF(N+1000) = IU
      START = STOP + 1
      GO TO 600
1400 CALL RENUM(START,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      IF(START.LT.-999.OR.START.GT.1000) GO TO 13720
      BUF(START+1000) = IU
      START = START + 1
      GO TO 600
1900 DO 1920 N = 1,NCL
      C = IA(N) + 1000
      IF(C.LT.1.OR.C.GT.2000) GO TO 1910
      IB(N) = BUF(C)
      GO TO 1920
1910 IB(N) = IA(N)
1920 CONTINUE
      GO TO 99400
C***** DESCRIBE
2000 IF (ERRORS.GT.0) GO TO 99500
      GO TO 99250
2900 CALL TABULA(JCL,IV,IA,IB)
      GO TO 150
C***** BAD BELOW ABOVE FROM TO OVER SPECI DONE *** DISPLAY
3000 GO TO (99500,98300,98400,98500,98600,3100,95100,99250),SUBNO
3100 IZC = 1
      GO TO 700
3900 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL GREY(JC,JR,ECHOED,IV,IU,IY,IZ,IZA,IZB,IZC,IA,catnam)
      GO TO 150
C***** LABEL
4000 IF(MAPNAM(9,IV).NE.'P') GO TO 4050
      ERRORS = ERRORS + 1
4010 WRITE(NP,4020) (MAPNAM(L,IV),L=1,8)
4020 FORMAT(/6H NO, ",8A1,25H" IS A PROTECTED OVERLAY )
      GO TO 99500

```

```

4050  CALL IOCAT(IV,1,CATNAM)
      if (comno.eq.55) go to 55000
      IF(ECHOED.EQ.1) WRITE(NP,4060)
4060  FORMAT(/' (SPECIFY A POSITIVE 2-DIGIT VALUE, A SPACE,'
1      ' A 16-CHARACTER LABEL AND/' A CARRIAGE RETURN FOR EACH',
1      ' REGION TO BE LABELED. FINISH WITH A VALUE OF -1)')
4080  READ(NI,4100) N, (LABEL(M),M=1,4)
4100  FORMAT(I2,1X,4A4)
      IF(N.LT.0) GO TO 4900
      IF(ECHOED.EQ.1) WRITE(NP,4150) N, (LABEL(M),M=1,4)
4150  FORMAT(/1H ,I2,1X,4A4)
      N = N + 1
      DO 4200 M = 1,4
4200      CATNAM(M,N) = LABEL(M)
      GO TO 4080
4900  CALL IOCAT(IV,2,CATNAM)
      GO TO 99450
C***** BAD FROM THRU DONE ***** LIST
5000  GO TO (99500,98000,98100,5050),SUBNO
5050  WRITE(NP,5100) mname, nr, nc, nwide
5100  FORMAT(15x,' MAP NAME:',a2/
1      15x,' NUMBER OF ROWS:',I5/
1      15x,' NUMBER OF COLUMNS:',I5/
1      15x,' STANDARD SCALE:',I5//
1      19x,' ',4x,'OVERLAY',5x,'PROTECTION'/
1      15x,' TITLE STATUS SCALE'/)
      IF(IU.LE.0.OR.IU.GT.JMAPS) IU = 1
      IF(IW.LE.0.OR.IW.GT.JMAPS) IW = JMAPS
      IF (IU.LE.IW) GO TO 5150
      N = IU
      IU = IW
      IW = N
5150  CONTINUE
      DO 5800 M = IU,IW
      IF(MAPNAM(1,M).EQ.' ') GO TO 5800
      ISCALE = ICHAR(MAPNAM(10,M)) + (256*ichar(mapnam(11,m)))
      WRITE(NP,5700) (MAPNAM(L,M),L=1,9), ISCALE
5700  FORMAT(20X,3x,4X,8A1,4X,A1,8X,I8)
5800  CONTINUE
      GO TO 150
C***** ZAP
6000  IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      fname(9:10) = '.-'
      fname(11:12) = mname
      do 6100 n = 1, 8
6100      fname(n:n) = mapnam(n,iv)
      CALL RENAME(JMAPS,IV,ZAPPED,ERRORS)
      if (errors.ne.1) then
      open(1,file=fname,status='OLD',access='direct',recl=1)
      read(1,rec=1601) mapnam(9,iv)
      close(1,status='DELETE',err=99500)
      endif
      IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      GO TO 99450
C***** BAD FOR PLUS TIME / DONE ***** AVERAGE

```

```

7000  GO TO (99500,96000,7100,7200,96300,99200),SUBNO
7150  IU = 2
      BUF(2) = 1
      GO TO 410
7100  CALL REMAP(JMAPS,N,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      IU = IU + 2
      BUF(IU-1) = N
      BUF(IU) = 1
      GO TO 600
7200  CALL RENUM(N,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      BUF(IU) = N
      GO TO 600
7900  IF(IU.LT.3) GO TO 99480
      CALL OVRLAY(JCL,9,IU,ERRORS,IA,IB)
      GO TO 99420
C***** XXX
9000  go to 150
C***** BAD TO DONE ***** XXXX
10000 GO TO 150
C***** BAD AT DONE ***** PRINT
11000 GO TO (99500,98000,11900),SUBNO
11900 CALL PRINT(IU,ECHOED)
      GO TO 150
C***** BAD FOR WITH / DONE ***** COVER
12000 GO TO (99500,96000,7100,96300,99200), SUBNO
12900 IF(IU.LT.3) GO TO 99480
      CALL OVRLAY(JCL,8,IU,ERRORS,IA,IB)
      GO TO 99420
C***** BAD / ASSIG FOR THRU TO WITH DONE ** CROSS
13000 GO TO (99500,96300,13100,96000,13300,13400,97000,99200), SUBNO
13050 DO 13060 N = 1,NBUF
13060 BUF(N) = 0
      GO TO 410
13100 CALL RENUM(IW,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      GO TO 600
13300 CALL RENUM(STOP,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      IF (STOP.LT.0.OR.STOP.GT.99) GO TO 13700
      IF(STOP.GE.START) GO TO 13320
          N = START
          START = STOP
          STOP = N
13320 CONTINUE
      DO 13350 N = START,STOP
13350 MATRIX(IX+1,N+1) = IW
      START = STOP + 1
      GO TO 600
13400 CALL RENUM(IX,ERRORS,W,WORD)
      IF(ERRORS.GT.0) GO TO 99500
      IF (IX.LT.0.OR.IX.GT.19) GO TO 13710
      W = W + 1
      CALL RENUM(START,ERRORS,W,WORD)

```

```

IF(ERRORS.GT.0) GO TO 99500
IF (START.LT.0.OR.START.GT.99) GO TO 13720
MATRIX(IX+1,START+1) = IW
GO TO 600
13700 N = STOP
GO TO 13800
13710 N = IX
GO TO 13800
13720 N = START
13800 WRITE(NP,13850) N
13850 FORMAT(/6H NO, ",I6,35H" IS OUTSIDE OF THE PERMITTED RANGE,
1 16H OF INPUT VALUES)
ERRORS = ERRORS + 1
GO TO 99500
13900 IF(IU.EQ.0) GO TO 99480
CALL FILE(JCL,IU,1,IB)
DO 13920 N = 1,NCL
C = (IB(N) * 20) + IA(N)+1
IF(C.GE.1.AND.C.LE.2000) GO TO 13910
IB(N) = 0
GO TO 13920
13910 IB(N) = BUF(C)
13920 CONTINUE
GO TO 99400
C***** EXPLAIN
14000 continue
do 14910 m = 1, 8
if(word(m,2).ne.' ') then
fname(m:m) = word(m,2)
n = m
elseif (m.eq.1) then
write (np,14900)
format('// Explanation files are currently ',
1 'available on the following topics:'//)
open(1,file='ex-files.tmp',status='old')
do 14902 mm = 1, 4
14902 read (1,14903)
14903 format(8a1)
do 14904 mm = 1, 9999
read (1,14903,end=14906) (word(n,3),n=1,8)
14904 write(np,14905) (word(n,3),n=1,8)
14905 format(30x,8a1)
14906 write(np,14907)
14907 format('// For help, enter EXPLAIN followed by ',
1 'the name of one of the topics listed above.'//
1 20X,'For example, EXPLAIN ZAP '//)
close(1)
go to 150
endif
14910 continue
fname(n+1:n+4) = '.EX '
do 14920 m = n+5, 12
14920 fname(m:m) = ' '
open(unit=1,file=fname,status='old',err=14930)
do 14924 n = 1,9999

```

```

                read (1,14922,end=14926) ((word(m,mm),m=1,8),mm=3,12)
14922          format(80a1)
14924          write (np,14925) ((word(m,mm),m=1,8),mm=3,12)
14925          format(1x,t1,80a1)
14926  close(1)
        go to 150
14930  WRITE(NP,14940) (word(m,2),m=1,8)
14940  FORMAT(/' NO, THERE IS NO EXPLANATION FILE ON "',8a1,'"/)
        GO TO 99495
C***** BAD    VERS FOR    /    DONE ***** MAXIMIZE
15000  GO TO (99500,7100,96000,96300,99200), SUBNO
15900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,6,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    VERS FOR    /    DONE ***** MINIMIZE
16000  GO TO (99500,7100,96000,96300,99200), SUBNO
16900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,7,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    FROM INTO FOR THRU DONE ***** SLICE
17000  GO TO (99500,98000,98200,96000,98100,99200), SUBNO
17020  IU = 32767
        IW = -32767
        GO TO 410
17900  CALL SLICE(JCL,IU,IW,IX,IA,IB)
        GO TO 99400
C***** BAD    BY    FOR    /    DONE ***** DIVIDE
18000  GO TO (99500,7100,96000,96300,99200), SUBNO
18900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,3,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    FOR THRU OVER TO DOWN DONE ***** SPREAD
19000  GO TO (99500,96000,97100,97300,98200,95100,99200), SUBNO
19900  IF(IY.EQ.0.AND.IW.EQ.0) GO TO 19950
        CALL SPREAD(JC,JR,JCL,IY,IW,IX,IU,IA,IB)
        GO TO 99400
19950  CALL EXTEND(JC,JR,IX,IA,IB)
        GO TO 99400
C***** BAD    FOR TO    /    DONE ***** ADD
20000  GO TO (99500,96000,7100,96300,99200), SUBNO
20900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,2,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    FOR MINU /    DONE ***** SUBTRACT
21000  GO TO (99500,96000,7100,96300,99200), SUBNO
21900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,5,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    BY    FOR    /    DONE ***** MULTIPLY
22000  GO TO (99500,7100,96000,96300,99200), SUBNO
22900  IF(IU.LT.3) GO TO 99480
        CALL OVRLAY(JCL,1,IU,ERRORS,IA,IB)
        GO TO 99420
C***** BAD    FOR ONTO DONE ***** STRIP
23000  GO TO (99500,96000,96900,99200), SUBNO

```

```

23900  CALL INPUTS(JC,JR,IV,IA,ECHOED)
      GO TO 99420
C***** BAD    ON    DONE ***** READ
24000  continue
      do 24910 m = 1, 8
          if(word(m,2).ne.' ') then
              fname(m:m) = word(m,2)
              n = m
          elseif (m.eq.1) then
              if(ni.ge.10) ni = ni - 10
              IF(ECHOED.EQ.1) WRITE(NP,24905)
              go to 99450
          endif
24905  FORMAT(/' OK, INSTRUCTIONS NOW BEING READ FROM PREVIOUS SOURCE')
      continue
      fname(n+1:n+4) = '.IN '
      do 24920 m = n+5, 12
24920  fname(m:m) = ' '
      ni = ni + 10
      open(unit=ni,file=fname,status='old',err=24940)
      IF(ECHOED.EQ.1) WRITE(NP,24930) fname
24930  FORMAT(/' OK, INSTRUCTIONS NOW BEING READ FROM FILE "',A12,'"')
      GO TO 99450
24940  ni = ni - 10
      open(1,file='in-files.tmp',status='old')
      write(np,24950) fname
24950  format(/' NO, THERE IS NO INSTRUCTION FILE CALLED "',A12,'"'/
1      ' INSTRUCTION FILES CURRENTLY AVAILABLE INCLUDE:'//)
      do 24960 mm = 1, 4
24960  read (1,24970)
24970  format(8a1)
      do 24980 mm = 1, 9999
          read (1,24970,end=24999) (word(n,3),n=1,8)
24980  write(np,24990) (word(n,3),n=1,8)
24990  format(30x,8a1)
24999  close(1)
      go to 99495
C***** BAD    FOR    ONTO DONE ***** POINT
25000  GO TO (99500,96000,96900,99200), SUBNO
25900  CALL INPUTC(JCL,IV,IA,ECHOED)
      GO TO 99420
C***** BAD    FOR    ONTO DONE ***** GRID
26000  GO TO (99500,96000,96900,99200), SUBNO
26900  CALL INPUTR(JCL,IV,IA,ECHOED)
      GO TO 99420
C***** ***** STOP
27000  RETURN
C***** MED    MAJOR MINOR DIVER DEVIA PROP OVERL DONE ** SCORE
C***** BAD    FOR    BY    QUIET COMPL TOTAL AVER MAX MIN
28000  GO TO (99500,96000,97400,28100,28200,95100,95200,95300,95400,
1      95500,95600,95700,95800,95900,95910,95911,28800),SUBNO
28100  IZA = 1
      GO TO 700
28200  IZB = 1
      GO TO 700

```

```

28800  IF(IU.NE.0) GO TO 99200
        IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
        GO TO 99250
28900  IF(IZ.EQ.0) GO TO 99480
        CALL PSCORE(JCL,JMAPS,ECHOED,IV,IU,IZ,IZA,IZB,ERRORS,IA,IB,catnam)
        IF (ERRORS.GT.0) GO TO 99500
        IF(IU.EQ.0) GO TO 150
        GO TO 99400
C***** MIN    MED    MAJOR MINOR DIVER DEVIA PROP  DIAG  DONE ** SCAN
C***** BAD    FOR    AROUN WITHI BEYON TOTAL AVER  MAX
29000  GO TO (99500,96000,97100,98200,98300,95100,95200,95300,
1      95400,95500,95600,95700,95800,95900,95910,28100,99200),SUBNO
29900  CALL SCAN(JC,JR,JCL,IU,IW,IX,IY,IZA,IA,IB)
        GO TO 99400
C***** BAD    FOR    MAXI  DONE ***** DIFFEREN
30000  GO TO (99500,96000,95100,99200), SUBNO
30900  IW = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
        CALL SLOPE(JC,JR,IU,IW,IA,IB)
        GO TO 99400
C***** ABOVE /    SCREE DONE **** RADIATE
C***** BAD    FOR    OVER ONTO THRU TO    AT
31000  GO TO (99500,96000,97000,97100,97200,98300,98400,
1      98700,96300,98500,99200), SUBNO
31900  IZB = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
        CALL VIEW(JC,JR,JCL,NV,IV,IU,IW,IX,IY,IZ,IZA,IZB,
1      IZC,IA,IB)
        GO TO 99420
C***** BAD    FOR    PRECI DONE ***** ORIENT
32000  GO TO (99500,96000,95200,99200), SUBNO
32100  IU = 3
        GO TO 410
32900  IW = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
        CALL SLOPE(JC,JR,IU,IW,IA,IB)
        GO TO 99400
C***** W      S      AT    DONE ***** PROFILE
C***** BAD    FOR    INFLE NE    SE    NW    SW    N    E
33000  GO TO (99500,96000,33100,95100,95300,95700,95900,95400,95200,
1      95800,95600,98200,99200), SUBNO
33100  IW = 1
        GO TO 700
33900  iy = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
        CALL PROFIL(JC,JR,IU,IW,IX,IY,IA,IB)
        GO TO 99400
C***** BAD    FOR    DIAG AT  DONE ***** CLUMP
34000  GO TO (99500,96000,95100,98100,99200),SUBNO
34900  CALL CLUMP(JC,JR,IU,IW,ERRORS,IA,IB)
        GO TO 99400
C***** BAD    FOR    AROU TO    DONE ***** SPAN
35000  GO TO (99500,96000,97000,98100,99200), SUBNO
35900  CALL SPAN(JC,JR,JCL,IU,IW,IA,IB)
        GO TO 99400
C***** BAD    FOR    DONE ***** SURVEY
37000  IU = 1
36000  GO TO (99500,96000,99200), SUBNO
36900  CALL SURVEY(JC,JR,IU,IA,IB)

```

```

GO TO 99400
C***** BAD FOR DONE ***** SIZE
38000 GO TO (99500,96000,99200), SUBNO
38900 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
CALL SIZE(JCL,IA)
GO TO 99420
C***** TOSPA TOTOP TOBOT TOLEF TORI FOR DONE **** RESPACE
C***** BAD / ONTO FROMS FROMT FROMB FROML FROMR
39000 GO TO (99500,96300,97500,98000,98100,98300,98200,98400,
1 98600,98700,98710,98720,98730,96000,99200), SUBNO
39900 CALL REGRID(jc,jr,jcl,JC,JR,JCL,IU,IW,IX,IY,IZ,IZA,
1 IZB,IZC,IZE,IZD,IZF,IA,IB)
GO TO 99400
C***** BAD BY FOR / DONE ***** EXPONENT
40000 GO TO (99500,7100,96000,96300,99200), SUBNO
40900 IF(IU.LT.3) GO TO 99480
CALL OVRLAY(JCL,4,IU,ERRORS,IA,IB)
GO TO 99420
C***** BAD FOR OVER DONE ***** DRAIN
41000 GO TO (99500,96000,97000,99200), SUBNO
41900 IF(IU.EQ.0) GO TO 99480
CALL DRAIN (ECHOED,JCL,JC,JR,IV,IU,IA,IB)
GO TO 99420
C***** BAD FOR AT TARG DONE***** TRACE
42000 IW = 5
IX = 5
GO TO (99500,96000,98000,42400,99200), SUBNO
42400 CALL RENUM(IW,ERRORS,W,WORD)
IF(ERRORS.GT.0) GO TO 99500
W = W+1
CALL RENUM(IX,ERRORS,W,WORD)
GO TO 98800
42900 CALL TRACE(JC,JR,IV,ECHOED,IU,IW,IX,IA,IB)
GO TO 99420
C***** MAP
43000 if (word(1,2).eq.blank) stop 255
do 45050 n = 2,40
if(word(1,n).eq.blank) goto 45060
call remap (jmaps,iy,errors,n,word)
if (errors.gt.0) go to 99500
45050 continue
45060 write (np,43100)
43100 format(/' PLEASE IDENTIFY THE OVERLAY AREA TO BE',
1 ' COPIED TO ANOTHER MAP:'//' TOP ROW NUMBER ...'//)
read (ni,*) iv
write (np,43200)
43200 format(/' BOTTOM ROW NUMBER ...'//)
read (ni,*) iu
write (np,43300)
43300 format(/' LEFT COLUMN NUMBER ...'//)
read (ni,*) iw
write (np,43400)
43400 format(/' RIGHT COLUMN NUMBER ...'//)
read (ni,*) ix
open (1,file='copies.tmp',status='new')

```

```

write(1,*) jc, jr, jcl, iv, iu, iw, ix
do 45600 n = 2,40
  if(word(1,n).eq.blank) goto 45700
  do 45500 m = 1,8
    mm = m
    if(word(m,n).eq.blank) goto 45501
    fname(mm:mm) = word(m,n)
45500
    mm = 9
    fname (mm:mm+1) = '.-'
    fname (mm+2:mm+3) = mname
    do 45550 m = mm+4, 12
      fname(m:m) = blank
45550
45600
45700
      write (1,*) char(39), fname, char(39)
      write (1,*) char(39), char(46), char(39)

      close(1)
      stop 255
C***** PROTECT
44000 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      mapnam(9,iv) = prtect
      GO TO 45100
C***** EXPOSE
45000 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      mapnam(9,iv) = 'E'
45100 call iotit (iv,2)
      GO TO 99450
C***** XXX
46000 go to 150
C***** XXX
47000 GO TO 150
C***** XXX
48000 GO TO 150
C***** XXX
49000 GO TO 150
C***** XXX
50000 GO TO 150
C***** WRITE
51000 continue
      do 51910 m = 1, 8
        if(word(m,2).ne.' ') then
          fname(m:m) = word(m,2)
          n = m
        elseif (m.eq.1) then
          np = 0
          go to 99450
        endif
51910 continue
      fname(n+1:n+4) = '.OUT'
      do 51920 m = n+5, 12
        fname(m:m) = ' '
51920
      IF(ECHOED.EQ.1) WRITE(NP,51930) fname
51930 FORMAT(/' OK, OUTPUT NOW BEING WRITTEN TO FILE "',A12,'"')
      np = np + 11
      open(unit=np,file=fname,err=99495)
      GO TO 99450
C***** BAD AT DONE ***** SCALE

```

```

52000 GO TO (99500,98000,52900),SUBNO
52900 if (mapnam(9,iv).eq.'P')
      1 call newmap(jmaps,iv,mapnam(1,iv),errors)
      IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      n = iu/256
      MAPNAM(11,IV) = CHAR(n)
      n = iu - (n*256)
      MAPNAM(10,IV) = CHAR(n)
      call iotit (iv,2)
      GO TO 99450
C***** XXX
53000 go to 150
C***** BAD FOR DONE ***** ASSIGN
54000 GO TO (99500,96000,99200), SUBNO
54100 W = W + 1
      GO TO 98000
54900 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      DO 54950 N = 1,NCL
54950 IB(N) = IU
      GO TO 99400
C***** BAD FOR DONE ***** COPY
55000 GO TO (99500,96000,99200), SUBNO
55900 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
c CALL IOCAT(IV,1,CATNAM)
      MAPNAM(10,NV) = MAPNAM(10,IV)
      MAPNAM(11,NV) = MAPNAM(11,IV)
      CALL iotit (NV,2)
      CALL FILE (JCL,NV,2,IA)
      go to 99445
C***** CONTOUR
56000 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      GO TO 99250
56900 continue
c CALL CONTOR(echoed,JC,JR,IA,IB)
      GO TO 150
C***** BAD FOR DONE ***** SORT
57000 GO TO (99500,96000,99200), SUBNO
57900 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL SORT(ECHOED,JCL,IA,IB)
      GO TO 99420
C***** BAD ON DONE ***** SHOW
58000 GO TO (99500,97000,99250), SUBNO
58900 if(iu.ne.0) go to 58902
      do 58901 n = 1,jcl
58901 ib(n) = 0
      go to 58903
58902 call file(jcl,iu,1,ib)
58903 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL SHOW(echoed,JC,JR,IA,IB)
      GO TO 150
C*****
59000 IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      GO TO 99450
C***** ECHO
60000 ECHOED = 1

```

```

GO TO 61100
C***** QUIET
61000 ECHOED = 0
61100 CALL ECHO(NV,IV,IU,IW,IX,IY,IZ,IZA,IZB,IZC,IZD,
1 IZE,IZF,ETOTAL,COMNO,W,WORD,CATNAM)
GO TO 150
95100 IU = 1
GO TO 700
95200 IU = 2
GO TO 700
95300 IU = 3
GO TO 700
95400 IU = 4
GO TO 700
95500 IU = 5
GO TO 700
95600 IU = 6
GO TO 700
95700 IU = 7
GO TO 700
95800 IU = 8
GO TO 700
95900 IU = 9
GO TO 700
95910 IU = 10
GO TO 700
95911 IU = 11
GO TO 700
96000 DO 96100 L = 1,8
96100 NEWNAM(L) = WORD(L,W)
GO TO 600
96300 CALL RECARD(W,WORD,2,ECHOED)
GO TO 700
96900 CALL REMAP(JMAPS,IV,ERRORS,W,WORD)
GO TO 98800
97000 CALL REMAP(JMAPS,IU,ERRORS,W,WORD)
GO TO 98800
97100 CALL REMAP(JMAPS,IW,ERRORS,W,WORD)
GO TO 98800
97200 CALL REMAP(JMAPS,IX,ERRORS,W,WORD)
GO TO 98800
97300 CALL REMAP(JMAPS,IY,ERRORS,W,WORD)
GO TO 98800
97400 CALL REMAP(JMAPS,IZ,ERRORS,W,WORD)
GO TO 98800
97500 CALL REMAP(JMAPS,IZA,ERRORS,W,WORD)
GO TO 98800
98000 CALL RENUM(IU,ERRORS,W,WORD)
GO TO 98800
98100 CALL RENUM(IW,ERRORS,W,WORD)
GO TO 98800
98200 CALL RENUM(IX,ERRORS,W,WORD)
GO TO 98800
98300 CALL RENUM(IY,ERRORS,W,WORD)
GO TO 98800

```

```

98400  CALL RENUM(IZ,ERRORS,W,WORD)
      GO TO 98800
98500  CALL RENUM(IZA,ERRORS,W,WORD)
      GO TO 98800
98600  CALL RENUM(IZB,ERRORS,W,WORD)
      GO TO 98800
98700  CALL RENUM(IZC,ERRORS,W,WORD)
      GO TO 98800
98710  CALL RENUM(IZD,ERRORS,W,WORD)
      GO TO 98800
98720  CALL RENUM(IZE,ERRORS,W,WORD)
      GO TO 98800
98730  CALL RENUM(IZF,ERRORS,W,WORD)
98800  IF(ERRORS.GT.0) GO TO 99500
      GO TO 600
99200  IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL NEWMAP(JMAPS,NV,NEWNAM,ERRORS)
      IF (ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
99250  IF(IV.LE.0.OR.COMNO.EQ.41) GO TO 99300
      CALL FILE(JCL,IV,1,IA)
99300  GO TO ( 1900, 2900, 3900, 150, 150, 150, 7900,
1         150, 150, 150, 150, 12900, 13900, 150,
1         15900, 16900, 17900, 18900, 19900, 20900, 21900,
1         22900, 23900, 150, 25900, 26900, 150, 28900,
1         29900, 30900, 31900, 32900, 33900, 34900, 35900,
1         36900, 36900, 38900, 39900, 40900, 41900, 42900,
1         150, 150, 150, 150, 150, 150, 150, 150,
1         150, 150, 150, 150, 54900, 55900, 56900,
1         57900, 58900, 150, 150, 150), COMNO
99400  IF(ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL FILE(JCL,NV,2,IB)
      GO TO 99430
99420  IF(ERRORS.GT.0.OR.(ETOTAL.GT.0.AND.CHOKE.EQ.1)) GO TO 99500
      CALL FILE(JCL,NV,2,IA)
99430  DO 99440 M = 1,120
          DO 99440 N = 1,4
99440      CATNAM(N,M) = DOTS
99445  CALL IOCAT (NV,2,CATNAM)
99450  IF(ECHOED.EQ.0) GO TO 150
      GO TO 61100
99480  WRITE(NP,99490)
99490  FORMAT(/' NO, A SECOND INPUT OVERLAY MUST ALSO BE SPECIFIED')
99495  ERRORS = 0
99500  ETOTAL = ETOTAL + ERRORS
      WRITE(NP,99505) (WORD(L,W),L=1,8), char(7)
99505  FORMAT(/' (INSTRUCTION CANCELLED AT "',8A1,'"')',a1/
1      /' (FOR AN EXPLANATION, ENTER "EXPLAIN")'/)
      GO TO 150
      END

```

```

SUBROUTINE ECHO(NV,IV,IU,IW,IX,IY,IZ,IZA,IZB,IZC,IZD,IZE,IZF,
1          ETOTAL,COMNO,W,WORD,CATNAM)
common/ref/ mn, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER BUF(2000), MATRIX(20,100), ETOTAL, COMNO, W
CHARACTER*4 CATNAM(4,120),STAT(3,11), CAT(4,20)
character*2 mn
CHARACTER*1 MAPNAM(11,160),WORD(8,40)
EQUIVALENCE (BUF(1),MATRIX(1,1))
DATA STAT/4HTOTA,4HL ,4H ,4HAVER,4HAGE ,4H ,
1          4HMAXI,4HMUM ,4H ,4HMINI,4HMUM ,4H ,
1          4HMEDI,4HAN ,4H ,4HMAJO,4HRITY,4H ,
1          4HMINO,4HRITY,4H ,4HDIVE,4HRSIT,4HY ,
1          4HDEVI,4HATIO,4HN ,4HPROP,4HORTI,4HONS ,
1          4HOVER,4HLAP ,4H /
C      01 RENU 02 DES 03 DIS 04 LAB 05 LIS 06 ZAP 07 AVE
C      08 NOT 09 SKI 10 RENA 11 PRI 12 COV 13 CRO 14 EXPL
C      15 MAX 16 MIN 17 SLI 18 MUL 19 SPR 20 ADD 21 SUB
C      22 DIV 23 STRI 24 REA 25 POIN 26 GRID 27 STO 28 SCO
C      29 SCAN 30 DIF 31 RAD 32 ORI 33 PROF 34 CLU 35 SPA
C      36 SUR 37 EUL 38 SIZ 39 RES 40 XPON 41 DRAI 42 TRAC
C      43 INF 44 PROT 45 XPOS 46 INS 47 REME 48 PARD 49 FORG
C      50 SAV 51 WRI 52 SCAL 53 IDE 54 MAP 55 COP 56 CON
C      57 SOR 58 SHO 59 HAT 60 ECH 61 QUI
GO TO ( 1000, 99000, 99000, 4000, 99000, 6000, 7000,
1      99000, 99000, 10000, 99000, 12000, 13000, 99000,
1      15000, 16000, 17000, 18000, 19000, 20000, 21000,
1      22000, 23000, 99000, 25000, 26000, 99000, 28000,
1      29000, 30000, 31000, 32000, 33000, 34000, 35000,
1      36000, 37000, 38000, 39000, 40000, 41000, 42000,
1      99000, 44000, 45000, 46000, 47000, 48000, 49000,
1      99000, 99000, 52000, 53000, 54000, 55000, 56000,
1      57000, 99000, 99000, 60000, 61000 ), COMNO
1000 WRITE(NP,1910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
1910 FORMAT(6H OK, ",8A1,23H" CREATED BY RECODING ",8A1,1H")
CALL IOCAT(IV,1,CATNAM)
N = 0
1920 IF(N.EQ.NBUF) RETURN
N = N + 1
M = N - 1000
IF(BUF(N).EQ.M) GO TO 1920
IF(M.LT.0.OR.M.GT.99) GO TO 1925
WRITE(NP,1930) BUF(N),M,(CATNAM(L,M+1),L=1,4)
GO TO 1935
1925 WRITE(NP,1930) BUF(N), M
1930 FORMAT(16X,10HASSIGNING ,I5,' TO ',I5,2X,4A4)
1935 NN = N + 1
IF(NN.GT.NBUF) NN = NBUF
DO 1950 NEXT = NN,NBUF
IF(BUF(NEXT).NE.BUF(N)) GO TO 1960
1950 CONTINUE
NEXT = NBUF + 1
1960 IF(NEXT.EQ.NN) GO TO 1920
N = NEXT - 1

```

```

      M = N - 1000
      IF(BUF(N).EQ.M) M = M-1
      IF(M.LT.0.OR.M.GT.99) GO TO 1965
      WRITE(NP,1970) M,(CATNAM(L,M+1),L=1,4)
      GO TO 1920
1965   WRITE(NP,1970) M
1970   FORMAT(32X,5HTHRU ,I5,2X,4A4)
      GO TO 1920
4000   WRITE(NP,4100) (WORD(L,2),L=1,8)
4100   FORMAT(6H OK, ",8A1,9H" LABELED)
      RETURN
6000   WRITE(NP,6100) (WORD(L,2),L=1,8)
6100   FORMAT(6H OK, ",8A1,'" ZAPPED')
      RETURN
7000   WRITE(NP,7910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),BUF(2)
7910   FORMAT(6H OK, ",8A1,24H" CREATED BY AVERAGING ",8A1,7H" TIMES,
1       I6)
      DO 7920 N = 4,IU,2
      IV = BUF(N-1)
7920   WRITE(NP,7930) (MAPNAM(L,IV),L=1,8), BUF(N)
7930   FORMAT(1X,31X,6HPLUS ",8A1,7H" TIMES,I6)
      RETURN
10000  WRITE(NP,10400) (WORD(L,2),L=1,8), (MAPNAM(L,IV),L=1,8)
10400  FORMAT(6H OK, ",8A1,14H" RENAMED TO ",8A1,1H")
      RETURN
12000  WRITE(NP,12910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
12910  FORMAT(6H OK, ",8A1,23H" CREATED BY COVERING ",8A1,1H")
      DO 12920 N = 4,IU,2
      IV = BUF(N-1)
12920  WRITE(NP,12930) (MAPNAM(L,IV),L=1,8)
12930  FORMAT(1X,30X,6HWITH ",8A1,1H")
      RETURN
13000  WRITE(NP,13910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),
1       (MAPNAM(L,IU),L=1,8), (MAPNAM(L,IV),L=1,8),
1       (MAPNAM(L,IU),L=1,8), (MAPNAM(L,NV),L=1,8)
13910  FORMAT(6H OK, ",8A1,23H" CREATED BY CROSSING ",8A1,8H" WITH ",8A1,
1       1H"//47X,8A1,1X,8A1,1X,8A1/
1       47X,5HVALUE,4X,5HVALUE,4X,5HVALUE)
      CALL IOCAT(IV,1,CATNAM)
      DO 13920 M = 1,20
      DO 13920 L = 1,4
13920  CAT(L,M) = CATNAM(L,M)
      CALL IOCAT(IU,1,CATNAM)
      DO 13940 M = 1,20
      MM = M-1
      NEXT = 1
      DO 13940 N = 1,100
      IF(MATRIX(M,N).EQ.0) GO TO 13940
      IF(NEXT.EQ.1) WRITE(NP,13925) (CAT(L,M),L=1,4)
13925  FORMAT(13X,4A4)
      NEXT = 0
      NN = N-1
      WRITE(NP,13930) (CATNAM(L,N),L=1,4), MM, NN, MATRIX(M,N)
13930  FORMAT(29X,4A4,5X,I2,' BY ',I2,' = ',I5)
13940  CONTINUE

```

```

WRITE(NP,13950)
13950  FORMAT(/27X,'VALUE COMBINATIONS NOT REASSIGNED =      0')
      RETURN
15000  WRITE(NP,15910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
15910  FORMAT(6H OK, ",8A1,25H" CREATED BY MAXIMIZING ",8A1,1H")
      GO TO 16905
16000  WRITE(NP,16910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
16910  FORMAT(6H OK, ",8A1,25H" CREATED BY MINIMIZING ",8A1,1H")
16905  DO 16920 N = 4,IU,2
      IV = BUF(N-1)
16920  WRITE(NP,16930) (MAPNAM(L,IV),L=1,8)
16930  FORMAT(1X,30X,8HVERSUS ",8A1,1H")
      RETURN
17000  WRITE(NP,17910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8), IX,IU,IW
17910  FORMAT(6H OK, ",8A1,22H" CREATED BY SLICING ",8A1,1H"/15X,
1      5H INTO,I5,12H LEVELS FROM,I5,5H THRU,I5)
      RETURN
18000  WRITE(NP,18910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
18910  FORMAT(6H OK, ",8A1,29H"      CREATED BY MULTIPLYING ",8A1,1H")
      GO TO 40905
19000  WRITE(NP,19970) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8), IX
19970  FORMAT(6H OK, ",8A1,24H" CREATED BY SPREADING ",8A1,1H"/
1      16X,16HTO A DISTANCE OF,I4,12H GRID SPACES)
      IF(IY.EQ.0) GO TO 19985
      IF(IU.EQ.1) GO TO 19974
      WRITE(NP,19972)
19972  FORMAT(16X,22HUPHILL OR ACROSS SLOPE)
      GO TO 19978
19974  WRITE(NP,19976)
19976  FORMAT(16X,24HDOWNHILL OR ACROSS SLOPE)
19978  WRITE(NP,19980) (MAPNAM(L,IY),L=1,8)
19980  FORMAT(16X,21HOVER THE SURFACE OF ",8A1,1H")
19985  IF(IW.GT.0) WRITE(NP,19990) (MAPNAM(L,IW),L=1,8)
19990  FORMAT(16X,26HTHROUGH THE IMPEDANCE OF ",8A1,1H")
      RETURN
20000  WRITE(NP,20910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
20910  FORMAT(6H OK, ",8A1,21H" CREATED BY ADDING ",8A1,1H")
      DO 20920 N = 4,IU,2
      IV = BUF(N-1)
20920  WRITE(NP,20930) (MAPNAM(L,IV),L=1,8)
20930  FORMAT(1X,30X,4HTO ",8A1,1H")
      RETURN
21000  WRITE(NP,21910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
21910  FORMAT(6H OK, ",8A1,26H" CREATED BY SUBTRACTING ",8A1,1H")
      DO 21920 N = 4,IU,2
      IV = BUF(N-1)
21920  WRITE(NP,21930) (MAPNAM(L,IV),L=1,8)
21930  FORMAT(1X,32X,7HMINUS ",8A1,1H")
      RETURN
22000  WRITE(NP,22910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
22910  FORMAT(6H OK, ",8A1,29H"      CREATED BY DIVIDING ",8A1,1H")
      GO TO 40905
23000  WRITE(NP,23910) (MAPNAM(L,NV),L=1,8)
23910  FORMAT(6H OK, ",8A1,22H" CREATED BY STRIPPING)
      IF(IV.GT.0) WRITE(NP,23920) (MAPNAM(L,IV),L=1,8)

```

```

23920  FORMAT(16X,4HON ",8A1,1H")
      RETURN
25000  WRITE(NP,25910) (MAPNAM(L,NV),L=1,8)
25910  FORMAT(6H OK, ",8A1,21H" CREATED BY POINTING)
      IF(IV.GT.0) WRITE(NP,23920) (MAPNAM(L,IV),L=1,8)
      RETURN
26000  WRITE(NP,26910) (MAPNAM(L,NV),L=1,8)
26910  FORMAT(6H OK, ",8A1,21H" CREATED BY GRIDGING)
      IF(IV.GT.0) WRITE(NP,23920) (MAPNAM(L,IV),L=1,8)
      RETURN
28000  WRITE(NP,28910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),
1      (MAPNAM(L,IZ),L=1,8), (STAT(L,IU),L=1,3)
28910  1  FORMAT(6H OK, ",8A1,22H" CREATED BY SCORING ",8A1,6H" BY ",
1      8A1,1H",1X,3A4)
      RETURN
29000  WRITE(NP,29910) (MAPNAM(L,NV),L=1,8),
1      (MAPNAM(L,IV),L=1,8), (STAT(L,IU),L=1,3), IX, IY
29910  1  FORMAT(6H OK, ",8A1,23H" CREATED BY SCANNING ",8A1,2H" ,3A4/16X,
1      17HOVER AREAS WITHIN,I3,11H BUT BEYOND,I3,15H GRID SPACES IN,
1      9H DIAMETER)
      IF(IZA.EQ.1) WRITE(NP,29915)
29915  FORMAT(16X,19HEXTENDED DIAGONALLY)
      IF(IW.GT.0) WRITE(NP,29920) (MAPNAM(L,IW),L=1,8)
29920  FORMAT(16X,8HAROUND ",8A1,1H")
      RETURN
30000  N = IU + 2
      WRITE(NP,30910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),
1      (STAT(L,N),L=1,3)
30910  1  FORMAT(6H OK, ",8A1,30H" CREATED BY DIFFERENTIATING ",8A1,2H" ,
1      3A4)
      RETURN
31000  WRITE(NP,31910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),
1      IY, IZ, IZA, IZC
31910  1  FORMAT(6H OK, ",8A1,24H" CREATED BY RADIATING ",8A1,4H" TO,I4,
1      12H GRID SPACES/ 16X,14HAT A HEIGHT OF,I2,
1      16H, SCREENWIDTH OF,I4,20H, AND EXPOSURE ABOVE,I4)
      IF(IU.GT.0) WRITE(NP,31920) (MAPNAM(L,IU),L=1,8)
31920  FORMAT(16X,21HOVER THE SURFACE OF ",8A1,1H")
      IF(IW.GT.0) WRITE(NP,31930) (MAPNAM(L,IW),L=1,8)
31930  FORMAT(16X,21HONTO THE SURFACE OF ",8A1,1H")
      IF(IX.GT.0) WRITE(NP,31940) (MAPNAM(L,IX),L=1,8)
31940  FORMAT(16X,26HTHROUGH THE IMPEDANCE OF ",8A1,1H")
      RETURN
32000  WRITE(NP,32910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
32910  FORMAT(6H OK, ",8A1,24H" CREATED BY ORIENTING ",8A1,1H")
      IF(IU.EQ.2) WRITE(NP,32920)
32920  FORMAT(16X,31HPRECISELY, IN AZIMUTHAL DEGREES)
      RETURN
33000  WRITE(NP,33910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
33910  FORMAT(6H OK, ",8A1,24H" CREATED BY PROFILING ",8A1,1H")
      IF(IW.EQ.1) GO TO 33918
      WRITE(NP,33912) IX
33912  1  FORMAT (15X,36H AT A MAXIMUM HORIZONTALITY LIMIT OF,I4,
1      21H UNITS PER GRID SPACE)
      GO TO 33930

```

```

33918 WRITE(NP,33920)
33920 FORMAT(16X,30HINFLECTION, IN SKYWARD DEGREES)
33930 WRITE(NP,33940)
33940 FORMAT(15X,29H AS VIEWED LOOKING TOWARD THE)
      IF(IU.NE.1.AND.IU.NE.4.AND.IU.NE.7) GO TO 33950
          WRITE(NP,33945)
33945     FORMAT(15X,6H NORTH)
          GO TO 33960
33950     IF(IU.NE.3.AND.IU.NE.6.AND.IU.NE.9) GO TO 33960
          WRITE(NP,33955)
33955     FORMAT(15X,6H SOUTH)
33960     IF(IU.GT.3) GO TO 33970
          WRITE(NP,33965)
33965     FORMAT(15X,5H EAST)
          RETURN
33970     IF(IU.LT.7) RETURN
          WRITE(NP,33975)
33975     FORMAT(15X,5H WEST)
          RETURN
34000 WRITE(NP,34910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8), IW
34910     FORMAT(6H OK, ",8A1,23H" CREATED BY CLUMPING ",8A1,1H"/
1         16X,35HAT A MAXIMUM CONNECTION DISTANCE OF,I3,
1         12H GRID SPACES)
      IF(IU.EQ.1) WRITE(NP,34920)
34920     FORMAT(16X,19HEXTENDED DIAGONALLY)
          RETURN
35000 WRITE(NP,35910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8), IW
35910     FORMAT(6H OK, ",8A1,23H" CREATED BY SPANNING ",8A1,4H" TO,I3,
1         17H GRID SPACES WIDE)
      IF(IU.GT.0) WRITE(NP,35920) (MAPNAM(L,IU),L=1,8)
35920     FORMAT(16X,8HAROUND ",8A1,1H")
          RETURN
36000 WRITE(NP,36910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
36910     FORMAT(6H OK, ",8A1,24H" CREATED BY SURVEYING ",8A1,1H")
          RETURN
37000 WRITE(NP,37910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
37910     FORMAT(6H OK, ",8A1,23H" CREATED BY EULERING ",8A1,1H")
          RETURN
38000 WRITE(NP,38910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
38910     FORMAT(6H OK, ",8A1,21H" CREATED BY SIZING ",8A1,1H")
          RETURN
39000 WRITE(NP,39910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
39910     FORMAT(6H OK, ",8A1,24H" CREATED BY RESPACING ",8A1,1H")
      IF(IZA.GT.0) WRITE(NP,23920) (MAPNAM(L,IZA),L=1,8)
      WRITE(NP,39920) IU,IW,IY,IX,IZ,IZB,IZC,IZD,IZE,IZF
39920     FORMAT(16X,14HFROM SPACING =,I3,6H,TOP =,I4,10H, BOTTOM =,I4,
1         7H,LEFT =,I4,9H, RIGHT =,I4/16X,14HTO SPACING =,I3,
1         6H,TOP =,I4,10H, BOTTOM =,I4,7H,LEFT =,I4,9H, RIGHT =,I4)
          RETURN
40000 WRITE(NP,40902) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
40902     FORMAT(6H OK, ",8A1,29H" CREATED BY EXPONENTIATING ",8A1,1H")
40905     DO 40920 N = 4,IU,2
          IV = BUF(N-1)
40920     WRITE(NP,40930) (MAPNAM(L,IV),L=1,8)
40930     FORMAT(39X,4HBY ",8A1,1H")

```

```

RETURN
41000 WRITE(NP,41910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8),
1      (MAPNAM(L,IU),L=1,8)
41910 FORMAT(6H OK, ",8A1,' CREATED BY DRAINING ',8A1,'" OVER '",
1      8A1,1H")
RETURN
42000 WRITE(NP,42910) (MAPNAM(L,NV),L=1,8),IW,IX,IU
42910 FORMAT(6H OK, ",8A1,20H" CREATED BY TRACING/
1      15X,17HTARGETING POINT (,I5,1H,,I5,4H) AT,
1      I5,21H UNITS PER GRID SPACE)
RETURN
44000 WRITE(NP,44910) (MAPNAM(L,IV),L=1,8)
44910 FORMAT(6H OK, ",8A1,11H" PROTECTED)
RETURN
45000 WRITE(NP,45910) (MAPNAM(L,IV),L=1,8)
45910 FORMAT(6H OK, ",8A1,9H" EXPOSED)
RETURN
46000 WRITE(NP,46910)
46910 FORMAT(21H OK, END OF INSERTION)
RETURN
47000 WRITE(NP,47910) ETOTAL
47910 FORMAT(35H OK, OUTSTANDING ERRORS REMEMBERED;,
1      16H CURRENT TOTAL =,I3)
RETURN
48000 WRITE(NP,48910)
48910 FORMAT(32H OK, OUTSTANDING ERRORS PARDONED)
RETURN
49000 WRITE(NP,49910) ETOTAL
49910 FORMAT(34H OK, OUTSTANDING ERRORS FORGOTTEN;,
1      16H CURRENT TOTAL =,I3)
RETURN
52000 N = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
WRITE(NP,52910) (MAPNAM(L,IV),L=1,8), N
52910 FORMAT(6H OK, ",8A1,11H" SCALED AT,I5,21H UNITS PER GRID SPACE)
RETURN
53000 WRITE(NP,53910) IV, (MAPNAM(L,IV),L=1,8)
53910 FORMAT(22H OK, MAP FILE LOCATION,I4,16H IDENTIFIED AS ",
1      8A1,1H")
RETURN
54000 WRITE(NP,54910) (MAPNAM(L,NV),L=1,8), IU
54910 FORMAT(6H OK, ",8A1,'" CREATED BY ASSIGNING A CONSTANT VALUE OF',
1      I5)
RETURN
55000 WRITE(NP,55910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
55910 FORMAT(6H OK, ",8A1,22H" CREATED BY COPYING ",8A1,1H")
RETURN
56000 WRITE(NP,91000)
C      CONTOUR
RETURN
57000 WRITE(NP,57910) (MAPNAM(L,NV),L=1,8), (MAPNAM(L,IV),L=1,8)
57910 FORMAT(6H OK, ",8A1,'" CREATED BY SORTING '",8A1,'"')
RETURN
60000 WRITE(NP,60910)
60910 FORMAT(26H OK, MESSAGES BEING ECHOED)
RETURN

```

```
61000  WRITE(NP,61910)
61910  FORMAT(27H OK, MESSAGES BEING QUIETED)
      RETURN
91000  FORMAT(35H NO, OPERATION NOT YET IMPLEMENTED ,41(1H*))
99000  RETURN
      END
```

```

SUBROUTINE TABULA(JCL,IV,IA,IB)
common/ref/ mn, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER*2 IA(JCL), IB(JCL)
INTEGER BUF(1), C
CHARACTER*4 CATNAM(4,120)
character*2 mn
CHARACTER*1 MAPNAM(11,160)
CALL IOCAT(IV,1,CATNAM)
NBIGS = 0
MAX = NCL/2
MISSED = 0
MINIM = 0
MAXIM = 0
DO 100 N = 1,MAX
    ib(n+max) = 0
100 IB(N) = -32768
    DO 5 N = 1,NBUF
005    BUF(N) = 0
        N = ICHAR(MAPNAM(10,IV)) + (256*ichar(mapnam(11,iv)))
        WRITE(NP,110) (MAPNAM(L,IV),L=1,9), N
110    FORMAT(/14X,'OVERLAY TITLE: ',8A1/10X,'PROTECTION STATUS: ',
1        4X,A1/22X,'SCALE: ',I5/
1        //19X,'REGION ',3X,'VALUE POINTS ')
        K = -32767
        DO 200 N = 1,NCL
            IF(IA(N).EQ.K) GO TO 190
            IF(IA(N).GT.-1000.AND.IA(N).LE.1000) GO TO 180
        K = -32767
        IF(NMAPS.GT.-1) GO TO 160
        DO 150 M = 1,MAX
            IF (IB(M).NE.IA(N)) GOTO 140
130 IB(M+MAX) = IB(M+MAX) + 1
            GO TO 200
140 IF(IB(M).NE.-32767) GO TO 150
            IB(M) = IA(N)
            NBIGS = M
            GO TO 130
150 CONTINUE
160 MISSED = MISSED + 1
            IF(IA(N).LT.MINIM) MINIM = IA(N)
            IF(IA(N).GT.MAXIM) MAXIM = IA(N)
            GO TO 200
180        K = IA(N)
            C = K + 1000
190        BUF(C) = BUF(C) + 1
200    CONTINUE
        DO 400 N = 1,2000
            IF(BUF(N).EQ.0) GO TO 400
            NN = N-1000
            IF(NN.LT.0.OR.NN.GT.99) GO TO 300
            WRITE(NP,250) (CATNAM(M,NN+1),M=1,4),NN,BUF(N)
250        FORMAT(11X,4A4,2X,I6,I7)
            GO TO 400

```

```

300      WRITE(NP,350) NN, BUF(N)
350      FORMAT(29X,I6,I7)
400      CONTINUE
        IF(NMAPS.GT.-1) GO TO 650
        IF(NBIGS.EQ.0) RETURN
        WRITE(NP,500)
500      FORMAT(/19X,32H VALUES BELOW -999 OR ABOVE 1000/)
        DO 600 M = 1,NBIGS
600      WRITE(NP,350) IB(M),IB(M+MAX)
650      CONTINUE
        IF(MISSED.GT.0) WRITE(NP,700) MISSED,MINIM,MAXIM
700      FORMAT(/19X,1H(,I5,' POINTS WITH VALUES UNDER -999 OR OVER 1000'/
C20X,' MINIMUM = ',I6,' MAXIMUM = ',I6,1H))
        RETURN
        END

```

[illegible]

[illegible]

[illegible]

```
180  CONTINUE
    IF(BLANKS.EQ.7) GO TO 800
    BLANKS = 7-BLANKS
    DO 300 N = 1,7
        DO 200 M = 1,50
            IF(INPUT(N).EQ.FIGURE(M)) GO TO 300
200      CONTINUE
300      IN(N) = M.
        WRITE(NP,350)
350      FORMAT(/)
        DO 600 LAYER = 1,5
            M = 1
400          WRITE(NP,500)((STRIP(MM,LAYER,IN(N)),MM=1,2),N=1,BLANKS)
500          FORMAT(1X,7(2A4,2X))
            M = M+1
            IF(M.LE.IU) GO TO 400
600      CONTINUE
    GO TO 50
800  CONTINUE
    RETURN
    END
```

```

SUBROUTINE NEWMAP(JMAPS,MAPNO,NEWMAM,ERRORS)
common/ref/ mname, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 MAPNAM(11,160),NEWMAM(8),BLANK,THAT(8),
1      PRTECT,EXPOSE
character*2 mname
character*4 catnam(4,120)
INTEGER ERRORS, SPACE
DATA BLANK/1H /, THAT/1HX,1H ,1H ,1H ,1H ,1H ,1H ,1H /,
1      PRTECT/1HP/,EXPOSE/1HE/

SPACE = 0
C      IF NO TITLE HAS BEEN SPECIFIED, ASSUME THE TITLE "X"
      IF (NEWMAM(1).NE.BLANK) GO TO 50
      DO 40 L = 1,8
040      NEWMAM(L) = THAT(L)

C      LOOK FOR AN EXISTING OVERLAY BY THE SAME TITLE
050      DO 200 MAPNO = 1, NMAPS
          DO 100 L = 1,8
              IF(MAPNAM(L,MAPNO).NE.NEWMAM(L)) GO TO 150
100          CONTINUE
          GO TO 500
150      IF(SPACE.EQ.0.AND.MAPNAM(1,MAPNO).EQ.BLANK) SPACE = MAPNO
200      CONTINUE
          IF(SPACE.GT.0) GO TO 400

C      NO MORE ROOM
      WRITE(NP,300)
300      FORMAT(/'NO, TOO MANY OVERLAYS FOR ONE MAP')
350      ERRORS = 1
          RETURN

C      SET UP FOR THE NEW OVERLAY
400      MAPNO = SPACE
          do 450 n = 1, 8
450          mapnam(n,mapno) = newnam(n)
              n = nwide/256
              MAPNAM(11,MAPNO) = CHAR(n)
              n = nwide - (n*256)
              MAPNAM(10,MAPNO) = CHAR(n)
              mapnam(09,mapno) = 'E'
              call iotit (mapno,2)
          RETURN

C      THE TITLE OF THE NEW OVERLAY MATCHES AN EXISTING ONE
500      IF(MAPNAM(9,MAPNO).EQ.PRTECT) GO TO 600
          WRITE(NP,550) (NEWMAM(L),L=1,8)
550      FORMAT(/15H (OVERWRITING ",8A1,2H"))
          RETURN

C      BUT THE EXISTING OVERLAY IS PROTECTED
600      WRITE(NP,650) (NEWMAM(L),L=1,8)
650      FORMAT(/6H NO, ",8A1,36H" IS THE NAME OF AN EXISTING OVERLAY,
1          20H WHICH IS PROTECTED )

```

GO TO 350
END

```

SUBROUTINE SMATCH(N,OLD,NEW)
N= # OF TRIES TO GET MATCH, TRIES+1 IF ENDCARD FOUND, 0 IF NO MA
CHARACTER*1 OLD(1), NEW(8), BLANK, STAR
INTEGER DIFRNT, OL
DATA BLANK/1H /, STAR/1H*/
DIFRNT= 0
NL = 0
N = 1
DO 200 OL = 1,300
    NL = NL + 1
    IF (OLD(OL).EQ.BLANK) GO TO 100
    IF (OLD(OL).EQ.STAR) GO TO 300
    IF (OLD(OL).NE.NEW(NL)) DIFRNT= 1
    GO TO 200
100    IF (DIFRNT.EQ.0) RETURN
    DIFRNT = 0
    N = N + 1
    NL = 0
200    CONTINUE
300    IF (DIFRNT.EQ.0) RETURN
    IF(NEW(1).NE.BLANK) GO TO 400
    N = N + 1
    RETURN
400    N = 0
    RETURN
END

```

```

SUBROUTINE RESUB(SUBNO,COMNO,ERRORS,W,WORD)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 WORD(8,40), SUBLET(320),REV(11),DIS(12),AVE(8),
1      COV(6), CRO(13), MAXMIN(6), SLI(9),
1      MDIVEX(6), SPR(11), ADD(6), SUB(6), INPSTR(4),
1      SCO(38), SCA(41), SLO(4), RAD(21), ORI(4),
1      PRO(42), CLU(6), SPA(6), RES(56)
INTEGER CTOSUB(61),START,SUBNO,ERRORS,W,COMNO
EQUIVALENCE (REV(1),SUBLET( 1)),
1      (DIS(1),SUBLET( 12)), (AVE(1),SUBLET( 24)),
1      (COV(1),SUBLET( 32)), (CRO(1),SUBLET( 38)),
1      (MAXMIN(1),SUBLET( 51)), (SLI(1),SUBLET( 57)),
1      (MDIVEX(1),SUBLET( 66)), (SPR(1),SUBLET( 72)),
1      (ADD(1),SUBLET( 83)), (SUB(1),SUBLET( 89)),
1      (INPSTR(1),SUBLET( 95)), (CLU(1),SUBLET( 99)),
1      (SCO(1),SUBLET(105)), (SCA(1),SUBLET(143)),
1      (SLO(1),SUBLET(184)), (RAD(1),SUBLET(188)),
1      (ORI(1),SUBLET(209)), (SPA(1),SUBLET(213)),
1      (PRO(1),SUBLET(219)), (RES(1),SUBLET(265))
DATA REV /1H/,1H ,1HA,1H ,1HF,1H ,1HT,1HH,1H ,1HT,1H*/
DATA DIS /1HB,1H ,1HA,1H ,1HF,1H ,1HT,1H ,1HO,1H ,1HS,1H*/
DATA AVE /1HF,1H ,1HP,1H ,1HT,1H ,1H/,1H*/
DATA COV /1HF,1H ,1HW,1H ,1H/,1H*/

```

```

1 DATA CRO      /1H/,1H ,1HA,1H ,1HF,1H ,
                  1HT,1HH,1H ,1HT,1H ,1HW,1H*/
DATA MAXMIN     /1HV,1H ,1HF,1H ,1H/,1H*/
DATA SLI        /1HF,1HR,1H ,1HI,1H ,1HF,1H ,1HT,1H*/
DATA MDIVEX     /1HB,1H ,1HF,1H ,1H/,1H*/
DATA SPR        /1HF,1H ,1HT,1HH,1H ,1HO,1H ,1HT,1H ,1HD,1H*/
DATA ADD        /1HF,1H ,1HT,1H ,1H/,1H*/
DATA SUB        /1HF,1H ,1HM,1H ,1H/,1H*/
DATA INPSTR     /1HF,1H ,1HO,1H*/ , CLU/1HF,1H ,1HD,1H ,1HA,1H*/
DATA SCO        /1HF,1H ,1HB,1H ,1HQ,1H ,1HC,1H ,1HT,1H ,1HA,1H ,
1              1HM,1HA,1HX,1H ,1HM,1HI,1HN,1HI,1H ,1HM,1HE,1H ,1HM,1HA,
1              1H ,1HM,1H ,1HD,1HI,1H ,1HD,1H ,1HP,1H ,1HO,1H*/
DATA SCA        /1HF,1H ,1HA,1HR,1H ,1HW,1H ,1HB,1H ,
1              1HT,1H ,1HA,1H ,1HM,1HA,1HX,1H ,1HM,1HI,1HN,1HI,1H ,1HM,1HE,
1              1H ,1HM,1HA,1H ,1HM,1H ,1HD,1HI,1HV,1H ,1HD,1HE,1H ,1HP,
1              1H ,1HD,1H*/
DATA SLO        /1HF,1H ,1HM,1H*/
DATA RAD        /1HF,1H ,1HO,1HV,1H ,1HO,1H ,1HT,1HH,
1              1H ,1HT,1H ,1HA,1HT,1H ,1HA,1H ,1H/,1H ,1HS,1H*/
DATA ORI        /1HF,1H ,1HP,1H*/ , SPA/1HF,1H ,1HA,1H ,1HT,1H*/
DATA PRO        /1HF,1H ,1HI,1H ,1HN,1HO,1HR,1HT,1HH,1HE,1H ,
1              1HS,1HO,1HU,1HT,1HH,1HE,1H ,1HN,1HO,1HR,1HT,1HH,1HW,
1              1H ,1HS,1HO,1HU,1HT,1HH,1HW,1H ,1HN,1H ,1HE,
1              1H ,1HW,1H ,1HS,1H ,1HA,1H*/
DATA RES        /1H/,1H ,1HO,1H ,1HF,1HR,1HO,1HM,1HS,
1              1H ,1HF,1HR,1HO,1HM,1HT,1H ,1HF,1HR,1HO,1HM,1HB,1H ,
1              1HF,1HR,1HO,1HM,1HL,1H ,1HF,1HR,1HO,1HM,1HR,1H ,
1              1HT,1HO,1HS,1H ,1HT,1HO,1HT,1H ,1HT,1HO,1HB,1H ,
1              1HT,1HO,1HL,1H ,1HT,1HO,1HR,1H ,1HF,1H*/
DATA CTOSUB/
1      001,      320,      012,      320,      062,      320,      024,
1      320,      320,      64,      103,      32,      38,      320,
1      51,      51,      57,      66,      72,      83,      89,
1      66,      95,      97,      95,      95,      320,      105,
1      143,      184,      188,      209,      219,      99,      213,
1      319,      319,      319,      265,      66,      95,      213,
1      320,      320,      320,      320,      320,      320,      320,
1      320,      97,      103,      103,      319,      319,      12,
1      319,      97,      12,      12,      12/

```

```

01 RENU 02 DES 03 DIS 04 LAB 05 LIS 06 ZAP 07 AVE
08 NOT 09 SKI 10 RENA 11 PRI 12 COV 13 CRO 14 EXPL
15 MAX 16 MIN 17 SLI 18 MUL 19 SPR 20 ADD 21 SUB
22 DIV 23 STRI 24 REA 25 POIN 26 GRID 27 STO 28 SCO
29 SCAN 30 DIF 31 RAD 32 ORI 33 PROF 34 CLU 35 SPA
36 SUR 37 EUL 38 SIZ 39 RES 40 XPON 41 STRE 42 TRAC
43 INF 44 PROT 45 XPOS 46 INS 47 REME 48 REMI 49 OVER
50 FLO 51 WRI 52 SCAL 53 IDE 54 MAP 55 COP 56 CON
57 sor 58 sho 59 HAT 60 ECH 61 QUI

```

TO MAKE CHANGES IN THE SUBCOMMAND STRING:

1. MAKE ADDITIONS OR DELETIONS IN THE DATA STATEMENT
2. CHANGE THE LENGTH OF SUBLET() - AND WHATEVER SUBSTRINGS() WERE CHANGED - IN THE INTEGER*2 DECLARATIONS ABOVE
3. REALIGN ALL EQUIVALENCES WHICH COME AFTER THE CHANGE
4. REVISE DATA CTOSUB FOR ALL NUMBERS WHICH COME AFTER THE CHANGE
START = CTOSUB(COMNO)

```

CALL SMATCH(SUBNO,SUBLET(START),WORD(1,W))
SUBNO = SUBNO + 1
IF (SUBNO.NE.1) RETURN
ERRORS = 1
WRITE(NP,100) (WORD(L,W),L=1,8)
100 FORMAT(/6H NO, ",8A1,26H" IS NOT A VALID MODIFIER )
RETURN
END

SUBROUTINE REMAP(JMAPS,MAPNO,ERRORS,W,WORD)
common/ref/ mn, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 WORD(8,40),MAPNAM(11,160),BLANK
character*2 mn
INTEGER W,ERRORS
DATA BLANK/1H /
IF (WORD(1,W).EQ.BLANK) GO TO 400
DO 200 M = 1,NMAPS
    MAPNO = M
    DO 100 N = 1,8
        IF(MAPNAM(N,MAPNO).NE.WORD(N,W)) GO TO 200
100    CONTINUE
    RETURN
200 CONTINUE
400 WRITE(NP,450) (WORD(L,W),L=1,8)
450 FORMAT(/44H NO, THERE IS NO OVERLAY IN THE FILE NAMED ",
1      8A1,2H" )
    ERRORS = 1
    RETURN
END

SUBROUTINE RENUM(NUM,ERRORS,W,WORD)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 DIGIT(20),MINUS, BLANK, WORD(8,40)
INTEGER D,ERRORS,W,SIGN,START
DATA DIGIT/ 1H0,1H ,1H1,1H ,1H2,1H ,1H3,1H ,1H4,1H ,
1      1H5,1H ,1H6,1H ,1H7,1H ,1H8,1H ,1H9,1H*/
DATA MINUS/1H-/, BLANK/1H /
NUM = 0
IF (WORD(1,W).NE.MINUS) GO TO 100
    SIGN = -1
    START = 2
    GO TO 200
100 SIGN = 1
    START = 1
200 CONTINUE
    DO 400 D = START,8
        CALL SMATCH(N,DIGIT,WORD(D,W))
        IF (N.EQ.0.OR.N.GT.10) GO TO 500
        NUM = (NUM * 10) + N - 1
400 CONTINUE
500 IF (D.EQ.START) GO TO 700
    IF(NUM.LE.32000) GO TO 600

```

```

        NUM = 32000
        WRITE(NP,550) (WORD(L,W-1),L=1,8)
550      FORMAT(/28H (THE LARGE NUMBER SPECIFIED,
1        8H AFTER ",8A1,21H" IS BEING TRUNCATED))
600    NUM = NUM * SIGN
        RETURN
700    ERRORS = 1
        NUM = -1
        WRITE(NP,750) (WORD(L,W-1),L=1,8)
750    FORMAT(/33H NO, A VALID NUMBER MUST FOLLOW ",8A1,2H" )
        RETURN
        END

SUBROUTINE RECARD(W,WORD,P,ECHOED)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 INPUT(80),BLANK,WORD(8,40),PROMPT(2),ISTOP(4)
INTEGER C,W,P,ECHOED,FINALC
DATA BLANK/1H /, PROMPT/1H.,1H/,ISTOP/1HS,1HT,1HO,1HP/

050    WRITE(NP,51) PROMPT(P)
051    FORMAT(/1x,A1)

        DO 55 W = 1, 40
        DO 55 L = 1, 8
055      WORD(L,W) = BLANK
        W = 0
        L = 0
        FINALC = 0

        READ(NI,52,ERR=160,END=160) (INPUT(C),C=1,80)
052    FORMAT(80A1)

        DO 100 C = 1, 80
            IF (INPUT(C).EQ.BLANK) GO TO 95
                FINALC = C
                IF (L.GE.8) GO TO 100
                    L = L+1
                    IF (L.EQ.1) W = W + 1
                    if(input(c).ge.'a'.and.
1          input(c).le.'z') then
                        WORD(L,W) = char(ichar(input(c))-32)
                    else
                        WORD(L,W) = INPUT(C)
                    endif
                GO TO 100

095      L = 0
100    CONTINUE
        W = 1
c      IF(ECHOED.EQ.1) WRITE(NP,150) (INPUT(C),C=1,FINALC)
c 150    FORMAT(/1x,79A1)
        RETURN
160    WRITE(NP,170)
170    FORMAT(/40H (MISSING COMMAND INTERPRETED AS 'STOP'))
        WORD(1,1) = ISTOP(1)

```

```

WORD(2,1) = ISTOP(2)
WORD(3,1) = ISTOP(3)
WORD(4,1) = ISTOP(4)
W = 1
RETURN
END

```

```

SUBROUTINE RENAME(JMAPS,MAPNO,NEWMAM,ERRORS)
common/ref/ mn, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
CHARACTER*1 MAPNAM(11,160),NEWMAM(8),BLANK,PRTECT
character*2 mn
character*4 catnam(4,120)
INTEGER ERRORS
DATA BLANK/1H /, PRTECT/1HP/

```

```

c      catch protected overlays
      IF(MAPNAM(9,MAPNO).EQ.PRTECT) GO TO 600

c      catch zap requests
      IF(NEWMAM(1).EQ.BLANK) GO TO 250

c      catch existing overlay titles
      DO 200 M = 1, NMAPS
        DO 100 L = 1, 8
          IF(MAPNAM(L,M).NE.NEWMAM(L)) GO TO 200
100      CONTINUE
      GO TO 500
200      CONTINUE

c      record the new title
250      do 300 m = 1, 8
300      mapnam(m,mapno) = newnam(m)
      RETURN

c      choke on already-used titles
500      WRITE(NP,550) (NEWMAM(L),L=1,8)
550      FORMAT(/34H NO, THERE IS ALREADY AN OVERLAY ",8A1,2H" )
      GO TO 700

c      choke on protected overlays
600      WRITE(NP,650) (MAPNAM(L,MAPNO),L=1,8)
650      FORMAT(/6H NO, ",8A1,25H" IS A PROTECTED OVERLAY )
700      ERRORS = 1
      RETURN
      END

```

```

SUBROUTINE SETCOR(IBUF,NVALS,IVAL)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IBUF(1)
DO 10 N = 1,NVALS
010  IBUF(N) = IVAL
      RETURN

```

END

```

SUBROUTINE PSCORE(JCL,JMAPS,ECHOED,IAMAP,OPTION,IBMAP,NOLIST,
1      NOSKIP,ERRORS,IA,IB,catnam)
common/ref/ mn, mapnam
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2   IA(JCL),IB(NCL)
CHARACTER*1 MAPNAM(11,160)
character*2 mn
CHARACTER*4 CATNAM(4,120)
INTEGER     ECHOED,OPTION,XLIST,ERRORS,AMAX,BMAX
AMAX = -32767
BMAX = -32767
CALL FILE(JCL,IBMAP,1,IB)
DO 7000 N = 1,NCL
      IF(IA(N).GT.AMAX) AMAX = IA(N)
7000      IF(IB(N).GT.BMAX) BMAX = IB(N)
      MMA = AMAX + 2
      IF(MMA.LE.2) MMA = 20
      MMB = BMAX + 2
      IF(MMB.LE.2) MMB = 20
      N = 0
C7008      IF( (MMA*MMB) + (MMA*8) + (MMB*6) + 20.LE. NBUF) GO TO 7009
            IF(MMA.LE.020.AND.MMB.LE.020) GO TO 7009
      ERRORS = 1
      WRITE(NP,7001)
7001      FORMAT(51H NO, THOSE INPUT MAPS HAVE TOO MANY VALUES TO SCORE,
1          9H AT ONCE ,16(1H*))
      RETURN
7009      CONTINUE
C7009      JMAX = 1
C          JMIN = JMAX      +1+( (MMA+1) /2)
C          JMAJOR = JMIN    +1+( (MMA+1) /2)
C          JTYPES = JMAJOR  +1+( (MMA+1) /2)
C          JBNAME = JTYPES  +1+( (MMA+1) /2)
C          JBCELL = JBNAME   +1+( (MMB*4)
C          JBSCOR = JBCELL   +1+MMB
C          JANAME = JBSCOR   +1+MMB
C          JACELL = JANAME    +1+( MMA*4)
C          JASCOR = JACELL    +1+MMA
C          JMCELL = JASCOR    +1+MMA
C7004      CALL SCORE(JCL,JMAPS,OPTION,NOLIST,NOSKIP,MMA,MMB,ERRORS,IAMAP,
C          1      IBMAP,XX(JMAX),XX(JMIN),XX(JMAJOR),XX(JTYPES),
C          1      XX(JBNAME),XX(JBCELL),XX(JBSCOR),XX(JANAME),
C          1      XX(JACELL),XX(JASCOR),XX(JMCELL),IA,IB)
7004      CALL SCORE(JCL,JMAPS,OPTION,NOLIST,NOSKIP,MMA,MMB,ERRORS,IAMAP,
1      IBMAP,IA,IB,catnam)
      RETURN
      END

SUBROUTINE SCORE(JCL,JMAPS,OPTION,NOLIST,NOSKIP,MMA,MMB,ERRORS,
1      IAMAP,IBMAP,IA,IB,catnam)
common      major,max,min,types,acells,bcells,mcells,ascore,
1      bscore
common/ref/ mn, mapnam
COMMON/SAM/ NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB

```

```

      INTEGER*2    MAJOR(020),MAX(020),
1      TYPES(020),MIN(020),IA(JCL),IB(JCL)
      CHARACTER*1  MAPNAM(11,160)
      character*2  mn
      CHARACTER*4  ANAMES(020,4),BNAMES(020,4),TNAMES(4),
1      CATNAM(4,120),DOTS,BLANK
      INTEGER      BCELLS(020),MCELLS(020,020),ASCORE(020),BSCORE(020),
1      NOLIST,OPTION,XLIST,AMAX,BMAX,
1      ERRORS,ALLACE,ALLBCE,ALLASC,ALLBSC,X,Y,ACELLS(020)
      REAL          ACELLP,ASCORP,BCELLP,BSCORP,MCELPA,MCELPB
      DATA BLANK/,' ',/,DOTS/'____'/
      MA = MMA-1
      MB = MMB-1
      ALLACE = 0
      ALLBCE = 0
      ALLASC = 0
      ALLBSC = 0
      DO 600 X = 1,MMA
          ACELLS(X) = 0
          ASCORE(X) = 0
          TYPES(X) = 0
      DO 600 Y=1,MMB
600      MCELLS(X,Y) = 0
      DO 601 X = 1,MB
          BCELLS(X) = 0
          BSCORE(X) = 0
          BNAMES(X,1) = BLANK
601      CONTINUE
      DO 110 X = 1,NCL
          ITEMA = IA(X)+1
          ITEMB = IB(X)+1
          IF(ITEMB.LT.1.OR.ITEMA.LT.1) GO TO 110
          IF(ITEMA.GT.MA.OR.ITEMB.GT.MB) GO TO 110
          ACELLS(ITEMA) = ACELLS(ITEMA) + 1
          ALLACE = ALLACE+1
          BCELLS(ITEMB) = BCELLS(ITEMB)+1
          ALLBCE = ALLBCE+1
          MCELLS(ITEMA,ITEMB) = MCELLS(ITEMA,ITEMB)+1
          ASCORE(ITEMA) = ASCORE(ITEMA)+ITEMB-1
          ALLASC = ALLASC+ITEMB-1
          BSCORE(ITEMB) = BSCORE(ITEMB) + ITEMA-1
          ALLBSC = ALLBSC + ITEMA-1
110      CONTINUE
          IF(ALLASC.LE.0) ALLASC = 1
          IF(ALLBSC.LE.0) ALLBSC = 1
          IF(ALLACE.LE.0) ALLACE = 1
          IF(ALLBCE.LE.0) ALLBCE = 1
          IF(NOLIST.EQ.1) GO TO 63
          CALL IOCAT(IAMAP,1,CATNAM(1,1))
          DO 300 M = 1,MA
              DO 300 N = 1,4
300      ANAMES(M,N) = CATNAM(N,M)
          CALL IOCAT(IBMAP,1,CATNAM(1,1))
          DO 400 M = 1,MB
              DO 400 N = 1,4

```

```

400  BNames(M,N) = CATNAM(N,M)
403  WRITE(NP,200)
200  FORMAT(/,10X,42X,4H% OF,11X,4H% OF/ 10X,17X,3HMAP,15X,4H# OF,
1      2X,5HTOTAL,10X,5HTOTAL,2X,7HAVERAGE/10X,3HMAP,14X,
1      10HREGIONS      ,7X,5HCELLS,3X,4HAREA,3X,5HSCORE,2X,
1      5HSCORE,4X,5HSCORE/)
      DO 121 X = 1,MA
          IF(NOSKIP.EQ.0.AND.ACELLS(X).LT.1) GO TO 121
          ITEM = X-1
          Z1 = ASCORE(X)
          Z2 = ALLASC
          ASCORP = Z1 * 100 / Z2
          DENOMA = 0
          IF(ACELLS(X).LE.0)DENOMA = 1
          Z3 = ACELLS(X) + DENOMA
          Z = Z1 / Z3
          Z1 = ACELLS(X) * 100
          Z4 = ALLACE
          ACELLP = Z1 / Z4
          WRITE(NP,201) (MAPNAM(L,IAMAP),L=1,8),ITEM,
1              (ANAMES(X,N),N=1,4),ACELLS(X),ACELLP,ASCORE(X),ASCORP,Z
201  FORMAT(10X,8A1,3X,I5,1X,4A4,I6,2X,F5.1,I8,2X,F5.1,1X,F8.2)
121  CONTINUE
      AMAX = ITEM+1
      Z = Z2 / Z4
      WRITE(NP,202) ALLACE,ALLASC,Z
202  FORMAT(10X,17X,13H-----,4X,5H-----,10X,5H-----,11X,
1      5H-----/10X,13X,17HTOTAL PER OVERLAY,3X,I6,7X,I8,8X,F8.2)
      WRITE(NP,200)
      DO 131 Y = 1,MB
          IF(NOSKIP.EQ.0.AND.BCELLS(Y).LT.1) GO TO 131
          ITEM = Y-1
          Z1 = BSCORE(Y)
          Z5 = ALLBSC
          BSCORP = Z1 * 100 / Z5
          DENOMB = 0
          IF(BCELLS(Y).LE.0) DENOMB = 1
          Z6 = BCELLS(Y) + DENOMB
          Z = Z1 / Z6
          Z1 = BCELLS(Y) * 100
          Z7 = ALLBCE
          BCELLP = Z1 / Z7
          WRITE(NP,201) (MAPNAM(L,IBMAP),L=1,8),ITEM,
1              (BNames(Y,N),N=1,4),BCELLS(Y),BCELLP,BSCORE(Y),BSCORP,Z
131  CONTINUE
      BMAX = ITEM+1
      Z = Z5 / Z7
      WRITE(NP,202) ALLBCE,ALLBSC,Z
      WRITE(NP,204)
204  FORMAT(/56X,4H% OF,2X,4H% OF,2X,4H% OF/3X,13HMAJOR OVERLAY,12X,
1      13HMINOR OVERLAY,9X,4H# OF,1X,5HTOTAL,1X,5HMAJOR,1X,
1      5HMINOR,4X,'    %'/7X,'REGIONS      ',15X,'REGIONS      ',
1      7X,'POINTS',1X,'AREA',2X,'AREA',2X,'AREA OVERLAP'/)
      DO 60 X = 1,AMAX
          IF(ACELLS(X).LT.1) GO TO 60

```

```

ITEMA = X-1
DO 59 Y = 1,BMAX
IF(NOSKIP.EQ.0.AND.MCELLS(X,Y).LE.0) GO TO 59
  DENOMA=0
  DENOMB=0
  IF(ACELLS(X).EQ.0) DENOMA = 1
  IF(BCELLS(Y).EQ.0) DENOMB = 1
  ITEMB = Y-1
  Z3 = ACELLS(X) + DENOMA
  Z6 = BCELLS(Y) + DENOMB
  Z = MCELLS(X,Y) * 100.0
  MCELPA = Z / Z3
  MCELPB = Z / Z6
  Z = Z / Z4
  ZZ = (MCELPA + MCELPB) / 2.0
  WRITE(NP,203) ITEMA,(ANAMES(X,N),N=1,4),ITEMB,
    1      (BNAMES(Y,N),N=1,4),MCELLS(X,Y),Z,MCELPA,MCELPB,ZZ
203      FORMAT(1X,I5,1X,4A4,3X,I5,1X,4A4,I6,3F6.1,2X,F6.2)
59      CONTINUE
      WRITE(NP,203)
60      CONTINUE
C      WRITE(NP,500)
63      CONTINUE
      IF(OPTION.EQ.0) GO TO 999
      IF(OPTION.LT.2.OR.OPTION.GT.9) GO TO 899
      IF(OPTION.NE.5) GO TO 64
      N = 0
      DO 65 ITEMA = 1,MMA
      DO 66 ITEMB = 1,MMB
        N = N+MCELLS(ITEMA,ITEMB)
        IF(N.GE.ALLACE/2) GO TO 65
66      CONTINUE
65      MAJOR(ITEMA) = ITEMB-1
64      CONTINUE
      DO 800 ITEMA = 1,MA
        IF(ACELLS(ITEMA).LE.0) GO TO 800
        MOST = 0
        MIN(ITEMA) = 29999
      DO 880 ITEMB = 1,MB
        IF(MCELLS(ITEMA,ITEMB).LE.0) GO TO 880
        IF(MIN(ITEMA).NE.29999) GO TO 882
        MIN(ITEMA) = ITEMB-1
882      CONTINUE
        IF(OPTION.EQ.7) GO TO 883
        IF(MCELLS(ITEMA,ITEMB).LT.MOST) GO TO 881
        GO TO 884
883      IF(MCELLS(ITEMA,ITEMB).GT.MOST) GO TO 881
884      CONTINUE
        MAJOR(ITEMA) = ITEMB-1
        MOST = MCELLS(ITEMA,ITEMB)
881      CONTINUE
        MAX(ITEMA) = ITEMB-1
        TYPES(ITEMA) = TYPES(ITEMA)+1
880      CONTINUE
800      CONTINUE

```

```

899  CONTINUE
      DO 900 X = 1,NCL
        IF(IA(X).LT.0.OR.IA(X).GT.MA) GO TO 900
        ITEMA = IA(X)+1
        GO TO (901,902,903,904,905,906,906,908,909,910,911),OPTION
901  IB(X) = ASCORE(ITEMA)
        GO TO 900
902  IB(X) = ASCORE(ITEMA)/ACELLS(ITEMA) + 0.5
        GO TO 900
903  IB(X) = MAX(ITEMA)
        GO TO 900
904  IB(X) = MIN(ITEMA)
        GO TO 900
905  IB(X) = MAJOR(ITEMA)
        GO TO 900
906  IB(X) = MAJOR(ITEMA)
        GO TO 900
908  IB(X) = TYPES(ITEMA)
        GO TO 900
909  IB(X) = (((ASCORE(ITEMA)+0.)/ACELLS(ITEMA)) - IB(X)) + 0.5
        GO TO 900
910  DENOMA = 0
        IF(ACELLS(ITEMA).LE.0) DENOMA = 1
        ITEMB = IB(X)+1
        IB(X) = ((MCELLS(ITEMA,ITEMB)*100)/(ACELLS(ITEMA)+DENOMA))+ 0.5
        GO TO 900
911  DENOMA = 0
        IF(IB(X).LT.0) GO TO 900
        DENOMB = 0
        IF(ACELLS(ITEMA).LE.0) DENOMA = 1
        ITEMB = IB(X)+1
        IF(BCELLS(ITEMB).LE.0) DENOMB = 1
        IB(X) = (MCELLS(ITEMA,ITEMB) * 50 *
1          ((1/ (ACELLS(ITEMA) + DENOMA)) +
1          (1/(BCELLS(ITEMB) + DENOMB)))) + 0.5
900  CONTINUE
999  CONTINUE
      RETURN
      END
      SUBROUTINE OVLAY(JCL,K,IU,ERRORS,IA,IB)
      COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
      COMMON BUF
      INTEGER ERRORS, BUF(1)
      INTEGER*2 IA(JCL),IB(JCL)
005  TOTALW = BUF(2)
      DO 500 N = 4,IU,2
        IV = BUF(N-1)
        CALL FILE(JCL,IV,1,IB)
        GO TO (10,20,30,40,50,60,70,80,90), K
010  DO 15 I = 1,NCL
        IF(IA(I).EQ.0) GO TO 015
        IF (IB(I).EQ.0) GO TO 12
        IA(I) = IA(I) / IB(I)
        GO TO 15
012  IF(IA(I).LT.0) GO TO 14

```

```

IA(I) = 32767
GO TO 15
014   IA(I) = -32767
015   CONTINUE
      GO TO 500
020   DO 25 I = 1,NCL
025   IA(I) = IA(I) + IB(I)
      GO TO 500
030   DO 35 I = 1,NCL
035   IA(I) = IA(I) * IB(I)
      GO TO 500
040   DO 45 I = 1,NCL
      IF(IA(I).NE.0.AND.IB(I).GT.0) GO TO 42
          IA(I) = 0
          GO TO 45
042   IA(I) = IA(I) ** IB(I)
045   CONTINUE
      GO TO 500
050   DO 55 I = 1,NCL
055   IA(I) = IA(I) - IB(I)
      GO TO 500
060   DO 65 I = 1,NCL
      IF (IB(I).GT.IA(I)) IA(I) = IB(I)
065   CONTINUE
      GO TO 500
070   DO 75 I = 1,NCL
      IF (IB(I).LT.IA(I)) IA(I) = IB(I)
075   CONTINUE
      GO TO 500
080   DO 85 I = 1,NCL
      IF (IB(I).NE.0) IA(I) = IB(I)
085   CONTINUE
      GO TO 500
090   NEWW = BUF(N)
DO 95 I = 1,NCL
095   IA(I)=((IA(I)*TOTALW)+(IB(I)*NEWW))/(TOTALW+NEWW)
      TOTALW = TOTALW + NEWW
500   CONTINUE
      RETURN
      END

```

```

SUBROUTINE GREY(JC,JR,ECHOED,IV,SS, TOP,BOTTOM,LEFT,RIGHT,OVR,
1      IA,legend)
common/ref/ mn, mapnam
COMMON/SAM/ NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
REAL      PCT, TOTAL
INTEGER*2  symnum(104), IA(NC,NR), TCELLS, CELLS(103)
CHARACTER*4 LEGEND(4,120)
character*2 mn
CHARACTER*1 BAR,DOTS,BLANK,STRIP(4,068),
1      DIGIT(999,3), SYMBOL(4,104), mapnam(11,160)
INTEGER    COL,ROW,WIDTH,ECHOED,SS, TOP,BOTTOM,RIGHT,OVR,V,C
DATA       DOTS/1H-/ , BLANK/1H / , BAR/1H+/
data symnum / 045,061,240,035,254,236,248,249,250,
1      048,049,050,051,052,053,054,055,056,057,
1      065,097,066,098,067,099,068,100,069,101,
1      070,102,071,103,072,104,073,105,074,106,
1      075,107,076,108,077,109,078,110,079,111,
1      080,112,081,113,082,114,083,115,084,116,
1      085,117,086,118,087,119,088,120,089,121,
1      090,122,123,125,042,043,047,092,060,064,
1      124,126,224,227,233,058,247,094,237,229,
1      228,253,169,170,168,173,062,036,033,063,
1      155,032,174,175,034/
DO 9008 n = 1, 103
      DO 9008 m = 1, 4
9008      symbol(m,n) = char(symnum(n))
      n = 0
      nn = 0
      nnn = 0
      do 9009 m = 1, 999
        n = n + 1
        if (n.eq.10) then
          n = 0
          nn = nn + 1
          if (nn.eq.10) then
            nn = 0
            nnn = nnn + 1
          endif
        endif
        digit(m,3) = char(48+n)
        digit(m,2) = char(48+nn)
9009      digit(m,1) = char(48+nnn)
c
c      INITIALIZE
DO 1001 N = 1,103
1001      CELLS(N) = 0
DO 1003 N = 1,068
DO 1003 LAYER = 1,4
1003      STRIP(LAYER,N) = BLANK
      TCELLS = 0
C      READ LEGEND NAMES
CALL IOCAT(IV,1,LEGEND)
DO 002 N = 1,4
002      LEGEND(N,101) = LEGEND(N,1)
DO 004 M = 1,99

```

```

DO 004 N = 1,4
004  LEGEND(N,M) = LEGEND(N,M+1)
C      ESTABLISH SYMBOLISM TO BE USED
C      DO 8 N = 1,103
C      DO 8 LAYER = 1,4
C 008  SYMBOL(LAYER,N) = SYMSET(LAYER,N)
      IF(SS.EQ.0) GO TO 40
023  IF(ECHOED.EQ.1) WRITE(NP,024)
024  FORMAT(53H (SPECIFY A 2-DIGIT VALUE FOLLOWED BY A SPACE, UP TO ,
1      '4 OVERPRINT CHARACTERS AND'/'A CARRIAGE RETURN FOR ',
1      53HEACH RESYMBOLIZED REGION. FINISH WITH A VALUE OF -1)/
1      8H ++ ++++)
025  READ(NI,030) N, (SYMBOL(LAYER,104),LAYER=1,4)
030  FORMAT(I2,1X,4A1)
      IF(N.LT.0) GO TO 40
      IF(ECHOED.EQ.1)WRITE(NP,35)N,(SYMBOL(LAYER,104),LAYER=1,4)
035  FORMAT(1H ,I2,1X,4A1)
      IF(N.EQ.0) N = 101
      DO 38 LAYER = 1,4
038  SYMBOL(LAYER,N) = SYMBOL(LAYER,104)
      GO TO 25
C      ESTABLISH DISPLAY BOUNDARIES
040  IF(LEFT .LE.0.OR. LEFT.GT.NC) LEFT = 1
      IF(RIGHT.LE.0.OR.RIGHT.GT.NC) RIGHT = NC
      MRIGHT = RIGHT
041  MORE = 0
042  IF(RIGHT.LE.LEFT+059) GO TO 45
      MORE = 1
043  RIGHT = RIGHT - 060
      IF(RIGHT.GT.LEFT+059) GO TO 043
      GO TO 042
045  IF(TOP.LE.0.OR.TOP.GT.NR) TOP = 1
      IF(BOTTOM.LE.0.OR.BOTTOM.GT.NR) BOTTOM = NR
C      WRITE COLUMN NUMBERS ABOVE MAP
      WRITE(NP,130) CHAR(27),CHAR(51),CHAR(21)
      WRITE(NP,130)
130  FORMAT(76x,3A1)
      DO 100 LAYER = 1,3
100  WRITE(NP,120) (BAR,N=1,3),BLANK,
1      (DIGIT(COL,LAYER),COL=LEFT,RIGHT),BLANK, (BAR,N=1,3)
120  FORMAT(68A1,8x,3a1)
      WRITE(NP,130)
C      PRINT MAP, ROW BY ROW
      WIDTH = (RIGHT-LEFT) + 9
      IROW = 1
      DO 500 ROW = TOP,BOTTOM
C      LOAD ROW NUMBERS INTO PRINT VECTOR MARGINS
      C = 1
      DO 150 LAYER = 1,2
      DO 140 N = 1,3
      STRIP(1,C) = DIGIT(ROW,N)
140  C = C+1
150  C = (RIGHT-LEFT) + 7
C      LOAD APPROPRIATE SYMBOLS INTO THE FOUR PRINT VECTORS
      LASTV = -32768

```

```

C = 4
DO 400 COL = LEFT,RIGHT
      IF CELL MATCHES NEIGHBOR TO LEFT, SAVE SOME WORK
190  IF(IA(COL,ROW).EQ.LASTV) GO TO 300
      LASTV = IA(COL,ROW)
C      FLAG CELLS OF VALUE < 0, = 0 OR > 100
      IF(IA(COL,ROW).LE.100) GO TO 200
      V = 103
      GO TO 300
200  IF(IA(COL,ROW).GE.0) GO TO 250
      V = 102
      GO TO 300
250  IF(IA(COL,ROW).NE.0) GO TO 280
      V = 101
      GO TO 300
280  V = IA(COL,ROW)
300  C = C+1
      DO 350 LAYER = 1,4
350  STRIP(LAYER,C) = SYMBOL(LAYER,V)
      CELLS(V) = CELLS(V) + 1
400  TCELLS = TCELLS + 1
      do 450 n = width+1, 68
      do 450 layer = 1, 4
450      strip(layer,n) = blank
C      WRITE OUT PRINT VECTORS
      IF (OVR.LE.0) GO TO 496
      M = WIDTH - 4
      WRITE(NP,120) (STRIP(1,N),N=1,68), CHAR(27),CHAR(51),CHAR(1)
      WRITE(NP,495) (STRIP(2,N),N=5,68), CHAR(27),CHAR(51),CHAR(1)
      WRITE(NP,495) (STRIP(3,N),N=5,68), CHAR(27),CHAR(51),CHAR(1)
490  IF(IROW.NE.2.AND.IROW.NE.4.AND.IROW.NE.6.AND.IROW.NE.8)GO TO 491
      WRITE(NP,495) (STRIP(4,N),N=5,68), CHAR(27),CHAR(51),CHAR(19)
      GO TO 500
491  IF(IROW.NE.10) GO TO 492
      WRITE(NP,495) (STRIP(4,N),N=5,68), CHAR(27),CHAR(51),CHAR(20)
      IROW = 0
      GO TO 500
492  WRITE(NP,495) (STRIP(4,N),N=5,68), CHAR(27),CHAR(51),CHAR(18)
495  FORMAT(4X,064A1,8x,3a1)
      GO TO 500
496  IF(IROW.NE.2.AND.IROW.NE.4.AND.IROW.NE.6.AND.IROW.NE.8)GO TO 497
      WRITE(NP,120)(STRIP(1,N),N=1,68), CHAR(27),CHAR(51),CHAR(22)
      GO TO 500
497  IF(IROW.NE.10) GO TO 498
      WRITE(NP,120)(STRIP(1,N),N=1,68), CHAR(27),CHAR(51),CHAR(23)
      IROW = 0
      GO TO 500
498  WRITE(NP,120)(STRIP(1,N),N=1,68), CHAR(27),CHAR(51),CHAR(21)
500  IROW = IROW + 1
C      WRITE COLUMN NUMBERS BELOW MAP
      WRITE(NP,130) CHAR(27),CHAR(51),CHAR(21)
      DO 600 LAYER = 1,3
600  WRITE(NP,120) (BAR,N=1,3),BLANK,
1      (DIGIT(COL,LAYER),COL=LEFT,RIGHT),BLANK, (BAR,N=1,3)
      IF(MORE.EQ.0) GO TO 640

```

```

WRITE(NP,130)
LEFT = RIGHT + 1
RIGHT = MRIGHT
STRIP(1,WIDTH-3) = BLANK
GO TO 41

```

WRITE OUT MAP TITLE AND LEGEND

```

640 WRITE(NP,650) CHAR(14), (mapnam(n,iv),n=1,8)
650 FORMAT(////A1,11X,8A1////)
TOTAL = TCELLS - (CELLS(102)+CELLS(103))
IF(CELLS(101).EQ.0) GO TO 750

```

LEGEND ENTRY FOR VALUE 0

```

PCT = ((CELLS(101)*100)/ TOTAL)
V = 0
IF(OVR.EQ.0) GO TO 680

```

```

WRITE(NP,130) CHAR(27),CHAR(51),CHAR(01)
DO 660 LAYER = 2,4
660 WRITE(NP,670) (SYMBOL(LAYER,101),N=1,10)
670 FORMAT(4X,10A1)
680 WRITE(NP,700) CHAR(27),CHAR(51),CHAR(36), (SYMBOL(1,101),N=1,10),
1 V, (LEGEND(N,101),N=1,4), CELLS(101), PCT
700 FORMAT(3A1,4X,10A1 ,I5,2X,4A4,I7,' POINTS',2X,F5.1,1H%)

```

LEGEND ENTRIES FOR VALUES FROM 1 THROUGH 100

```

750 CONTINUE
DO 800 V = 1,100
IF(CELLS(V).EQ.0) GO TO 800
PCT = ((CELLS(V)*100)/ TOTAL)
WRITE(NP,130)
IF(OVR.EQ.0) GO TO 780

```

```

WRITE(NP,130) CHAR(27),CHAR(51),CHAR(01)
DO 760 LAYER = 2,4
760 WRITE(NP,670) (SYMBOL(LAYER,V),N=1,10)
780 WRITE(NP,700) CHAR(27),CHAR(51),CHAR(36), (SYMBOL(1,V),N=1,10),
1 V, (LEGEND(N,V),N=1,4), CELLS(V), PCT

```

```

800 CONTINUE
LEGEND ENTRY FOR VALUES LESS THAN 0 AND GREATER THAN 100

```

```

IF(CELLS(102).EQ.0) GO TO 830
WRITE(NP,820) (SYMBOL(1,102),N=1,10), CELLS(102)
820 FORMAT(/4X,10A1,' <0 (BELOW 0) ',I7,' POINTS')
830 IF(CELLS(103).EQ.0) GO TO 900
WRITE(NP,870) (SYMBOL(1,103),N=1,10), CELLS(103)
870 FORMAT(/4X,10A1,' >100 (ABOVE 100) ',I7,' POINTS')
900 CONTINUE
RETURN
END

```

```

SUBROUTINE SCAN(JC,JR,JCL,OPTION,SMAP,OUTER,INNER,SQUARE,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON ROLL
INTEGER ROLL(1),INNER,RADIUS,
1 OFFSET, SQUARE,OPTION,SMAP,OUTER,ROW,COLUMN,PEGS,
1 TYPES,X,Y,TOP,BOTTOM,RIGHT,CELL,DIFER,SHIFT
INTEGER*2 IA(JC,JR), IB(JC,JR)
REAL RO,CO,SUM,HALO
IF(OPTION.EQ.0.OR.OPTION.GT.10) OPTION = 2
IF(SMAP.GT.0) GO TO 105

```

```

        CALL SETCOR(IB(1,1),NCL,1)
        GO TO 106
105    CALL FILE(JCL,SMAP,1,IB)
106    IF(INNER.LT.0) INNER = -INNER
        OFFSET = 0
        IF(OUTER.LE.0) OUTER = 3
        RADIUS = OUTER/2
        IF(RADIUS.EQ.0) RADIUS = 1
        IF(RADIUS*2.EQ.OUTER) OFFSET = 1
        IF(RADIUS.GT.19) RADIUS = 1
        IF(INNER.GT.RADIUS*2) INNER = 0
        MAX = 2000
        IHALF = NR/2
        IF(OPTION.GT.10) OPTION = 2
C.....LOOP THROUGH BUFFERS
        DO 801 ROW = 1,NR
        IF(ROW.EQ.IHALF) WRITE(NP,150)
150    FORMAT(27H (SCAN OPERATION HALF DONE))
        DO 801 COLUMN = 1,NC
            IF(IB(COLUMN,ROW).LE.0) GO TO 801
            CO = COLUMN+(0.5*OFFSET)
            RO = ROW+(0.5*OFFSET)
            TOP = (RO-RADIUS) + 0.5
            IF(TOP.LT.1) TOP = 1
            BOTTOM = RO + RADIUS
            IF(BOTTOM.GT. NR) BOTTOM = NR
            DO 750 N = 1,MAX
750        ROLL(N) = 0
            SUM = 0.0
            PEGS = 0
            MIN = 29999
            MAX = 0
            TYPES = 0
            MOST = 0
            IF(OPTION.EQ.7) MOST = 29999
            MAJOR = 0
C.....LOOP THROUGH SCAN FIELD AROUND EACH BUFFER CELL
            DO 800 X = TOP,BOTTOM
                SHIFT = RADIUS
                IF(SQUARE.NE.1) SHIFT=SQRT((RADIUS**2)-((RO-X)**2)+0.1)
200        LEFT = CO-SHIFT
                IF(LEFT.LT.1) LEFT = 1
                RIGHT = CO+SHIFT
                IF(RIGHT.GT.NC) RIGHT = NC
                DO 800 Y = LEFT,RIGHT
                    CELL = IA(Y,X) + 1
                    IF(CELL.LE.0.OR.CELL.GT.2000) GO TO 800
                    PEGS = PEGS + 1
                    IF(CELL.LT.MIN) MIN = CELL
                    IF(CELL.GT.MAX) MAX = CELL
                    GO TO (771,771,800,800,773,773,773,773,770,773),OPTION
C.....VARIABLE SCAN OPTION PROGRAM GOES HERE                                OPTION
                    GO TO 800
770        DIFFER =IA(Y,X)-IA(COLUMN,ROW)
                IF(DIFFER.LT.0) DIFFER = -DIFFER

```

```

SUM = SUM + DIFFER
GO TO 800
771 SUM = SUM+CELL-1
GO TO 800
773 IF(ROLL(CELL).EQ.0) TYPES = TYPES+1
ROLL(CELL) = ROLL(CELL)+1
IF(OPTION.EQ.10) GO TO 800
IF(OPTION.EQ.7) GO TO 622
IF(ROLL(CELL).EQ.MOST.AND.CELL.GE.MAJOR) MAJOR = CELL
IF(ROLL(CELL).LE.MOST) GO TO 800
GO TO 621
622 IF(ROLL(CELL).EQ.MOST.AND.CELL.GE.MAJOR) MAJOR = CELL
IF(ROLL(CELL).GE.MOST) GO TO 800
621 MOST = ROLL(CELL)
MAJOR = CELL
800 CONTINUE
C.....ASSIGN SCAN VALUE
IF(PEGS.LE.0) GO TO 799
IF(OPTION.NE.5) GO TO 502
N = 0
DO 500 CELL = 1,MAX
N = N+ROLL(CELL)
IF(N.GE.PEGS/2) GO TO 501
500 CONTINUE
501 MAJOR = CELL
502 CONTINUE
GO TO (781,782,783,784,785,786,786,788,782,790),OPTION
C.....VARIABLE SCAN OPTION PROGRAM GOES HERE OPTION 21
781 IF(SQUARE.EQ.1) GO TO 881
IB(COLUMN,ROW) = (SUM*22*((RADIUS+(0.5*(1-OFFSET)))**2)
1 - ((INNER / 2.0)**2)))/(7*PEGS)
GO TO 801
881 IB(COLUMN,ROW) = (SUM* (((2*RADIUS)+1-OFFSET)**2)
1 - (INNER**2))) / PEGS
GO TO 801
782 IB(COLUMN,ROW) =(SUM/PEGS) + 0.5
GO TO 801
783 IB(COLUMN,ROW) = MAX-1
GO TO 801
784 IB(COLUMN,ROW) = MIN-1
GO TO 801
785 IB(COLUMN,ROW) = MAJOR-1
GO TO 801
786 IB(COLUMN,ROW) = MAJOR-1
GO TO 801
788 IB(COLUMN,ROW) = TYPES
GO TO 801
790 CELL = IA(COLUMN,ROW) +1
IB(COLUMN,ROW) = (ROLL(CELL)*100.) / PEGS
GO TO 801
799 IB(COLUMN,ROW) = 0
801 CONTINUE
RETURN
END

```

```

SUBROUTINE VIEW (JC,JR,JCL,OMAP,VPMAP,ELMAP,VHMAP,SHMAP,EXTENT,
1          ILEVEL,SWIDTH,CWIDTH,XANGLE,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
common shadow
INTEGER*2 IA(JC,JR), IB(JC,JR), SHADOW(360)
INTEGER OMAP,VPMAP,ELMAP,VHMAP,SHMAP,EXTENT,SWIDTH,CWIDTH,
1      XANGLE,C,R,UPDOWN,RIGHT,ROW,COL,START,STOP,ANGLE,AZMUTH
REAL HEIGHT,BASE,WIDTH,RADIAN
CALL FILE(JCL,VPMAP,1,IB)
IF(VHMAP.EQ.0) GO TO 005
      CALL FILE(JCL,VHMAP,1,IA)
      GO TO 006
005      CALL SETCOR(IA(1,1),NCL,0)
006      CONTINUE
C.....SET VIEWPOINTS TO -29999 ON VISIBLE HEIGHT MAP IN IA
      DO 110 C = 1,NC
      DO 110 R = 1,NR
            IF(IA(C,R).LT.0) IA(C,R) = 0
110          IF(IB(C,R).GT.0) IA(C,R) = -29999
            IF(ELMAP.EQ.0) GO TO 012
            CALL FILE(JCL,ELMAP,1,IB)
            GO TO 013
012      CALL SETCOR(IB(1,1),NCL,0)
013      IF(ILEVEL.LT.0) ILEVEL = 0
C.....SET VIEWPOINTS TO ELEVATION PLUS EYELEVEL AND NON-VIEWPOINTS TO
C.....(ELEVATION + VISIBLE HEIGHT + 2) * -1 IN IA, THEN SIDELINE IA
      DO 120 C = 1,NC
      DO 120 R = 1,NR
            IF(IB(C,R).LT.0) IB(C,R) = 0
            IF(IA(C,R).EQ.-29999) GO TO 115
            IA(C,R) = -(IB(C,R) + IA(C,R) + 2)
            GO TO 120
115          IA(C,R) = IB(C,R) + ILEVEL
120      CONTINUE
      CALL FILE(JCL,OMAP,2,IA)
C.....READ SCREENHEIGHT MAP INTO IA THEN ADD IT TO ELEVATION IN IB
      IF(SHMAP.EQ.0) GO TO 022
      CALL FILE(JCL,SHMAP,1,IA)
      GO TO 023
022      CALL SETCOR(IA(1,1),NCL,0)
023      CONTINUE
      DO 130 C = 1,NC
      DO 130 R = 1,NR
            IF(IA(C,R).LT.0) IA(C,R) = 0
130          IB(C,R) = IB(C,R) + IA(C,R)
C.....RESTORE THE SIDELINED IA, THEN SET REMAINING PARAMETERS
      CALL FILE(JCL,OMAP,1,IA)
      IF(EXTENT.LE.0) EXTENT = 4
      IF(CWIDTH.LE.0) CWIDTH = NWIDE
      WIDTH = SWIDTH * 0.5
      IF(SWIDTH.LE.0) WIDTH = (CWIDTH*0.7071) + 0.5
      XANGLE = XANGLE * 10
      IF(XANGLE.LE.0) XANGLE = 0
C.....LOOP THROUGH VIEWPOINT CELLS
      DO 801 ROW = 1,NR

```

```

DO 801 COL = 1,NC
  IF(IA(COL,ROW).LT.0) GO TO 801
  WRITE(NP,150) ROW, COL
150  FORMAT(' (RADIATING FROM ROW ',I4,', COLUMN ',I4,')')
      RADIUS = 1
      CALL SETCOR(SHADOW(1),360,-29999)
C.....LOOP AROUND VIEWPOINT CELL
701  LEFT = COL-RADIUS
      IF(LEFT.LT.1) LEFT = 1
      RIGHT = COL+RADIUS
      IF(RIGHT.GT.NC) RIGHT = NC
      UPDOWN = -1
      DO 800 C = LEFT,RIGHT
700  UPDOWN = -UPDOWN
      R = ROW - (UPDOWN * (RADIUS-IABS(COL-C)) )
      IF(R.LT.1.OR.R.GT.NR) GO TO 731
C.....COMPUTE AZIMUTHAL BEARING OF SCREENING CELL
      HEIGHT = COL - C + 0.0
      BASE = R - ROW + 0.0
      RADIAN = ATAN2(HEIGHT,BASE)
      AZMUTH = (RADIAN*57.296) + 180.5
      BASE = SQRT( BASE**2 + HEIGHT**2 ) * CWIDTH
      IF(BASE.LE.EXTENT*CWIDTH) GO TO 702
      UPDOWN = -1
      GO TO 800
C.....COMPUTE AZIMUTHAL RANGE OF SCREENING EFFECT
702  RADIAN = ATAN2(WIDTH,BASE)
      LATERL = RADIAN * 57.296
      START = AZMUTH - LATERL
      ISTOP = AZMUTH + LATERL
      IF(IA(C,R).GE.-1) GO TO 703
      HEIGHT = -(IA(C,R)+IA(COL,ROW)+2.0)
      RADIAN = ATAN2(HEIGHT,BASE)
      ANGLE = RADIAN * 572.96
      IF(ANGLE-XANGLE.GE.SHADOW(AZMUTH)) IA(C,R) = -1
703  HEIGHT = IB(C,R) - IA(COL,ROW) + 0.0
      RADIAN = ATAN2(HEIGHT,BASE)
      ANGLE = RADIAN * 572.96
      IF(START.GT.0) GO TO 710
      START = 360 + START
      DO 705 N = START,360
      IF(ANGLE.GT.SHADOW(N)) SHADOW(N) = ANGLE
705  CONTINUE
      START = 1
      GO TO 720
710  IF(ISTOP.LE.360) GO TO 720
      ISTOP = ISTOP-360
      DO 715 N = 1,ISTOP
      IF(ANGLE.GT.SHADOW(N)) SHADOW(N) = ANGLE
715  CONTINUE
      ISTOP = 360
720  CONTINUE
      DO 725 N = START,ISTOP
      IF(ANGLE.GT.SHADOW(N)) SHADOW(N) = ANGLE
725  CONTINUE

```

```

731         IF(UPDOWN.EQ.1.AND.R.NE.ROW) GO TO 700
           UPDOWN = -1
800         CONTINUE
           IF(RADIUS*10.GT.EXTENT*15) GO TO 801
           RADIUS = RADIUS+1
           GO TO 701
801         CONTINUE
           DO 900 ROW = 1, NR
           DO 900 COL = 1, NC
               IF(IA(COL,ROW)+1) 897,898,899
897             IA(COL,ROW) = 0
               GO TO 900
898             IA(COL,ROW) = 1
               GO TO 900
899             IA(COL,ROW) = 1
900         CONTINUE
999         CONTINUE
           RETURN
           END

SUBROUTINE CLUMP(JC,JR,DIAGON,REACH,ERRORS,IA,IB)
COMMON OLCLAN, ROLL, CLAN, DAM
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER HIBLOB,HICLAN,OLCLAN(500),CLAN(500),ROLL(500),DAM(500),
1         DIAGON,REACH,ROW,COLUMN, TOP,BOTTOM,SHIFT,LINKS,CELL,
1         X,Y,RIGHT,OTHERS,SISTER,COUSIN,XCLAN,D,ERRORS
INTEGER*2 IA(JC,JR), IB(JC,JR)
nomore = jr + 1
IF(REACH.LE.0) REACH = 1
IF(REACH.GT.10) REACH = 10
DO 720 N = 1,500
    OLCLAN(N) = 0
    CLAN(N) = 0
    ROLL(N) = 0
720    DAM(N) = 0
HIBLOB = 1
CLAN(1) = 1
DO 710 ROW = 1, NR
DO 710 COLUMN = 1, NC
    IB(COLUMN,ROW) = IA(COLUMN,ROW)
    IF(IA(COLUMN,ROW).GT.0) IA(COLUMN,ROW) = 1
710    CONTINUE
DO 801 ROW = 1, NR
    TOP = ROW - REACH
    IF(TOP.LT.1) TOP = 1
DO 801 COLUMN = 1, NC
    IF(IA(COLUMN,ROW).LT.1) GO TO 801
    SHIFT = REACH
    LINKS = 0
DO 707 N = 1,HIBLOB
707    ROLL(N) = 0
DO 800 X = TOP,ROW
    IF(DIAGON.EQ.1) GO TO 200
    SHIFT = SQRT((REACH**2)-((ROW-X)**2)+0.1)
200    LEFT = COLUMN-SHIFT

```

```

      RIGHT = COLUMN+SHIFT
      IF(LEFT.LT.1) LEFT = 1
      IF(RIGHT.GT.NC) RIGHT = NC
DO 800 Y = LEFT,RIGHT
      IF(IB(Y,X).NE.IB(COLUMN,ROW)) GO TO 800
      IF(IA(Y,X).LE.1) GO TO 800
700      SISTER = IA(Y,X)
      LOGAL = SISTER
      IF(ROLL(SISTER).GE.1) GO TO 802
      LINKS = LINKS+1
802      ROLL(SISTER) = 1 + ROLL(SISTER)
      IF(ROLL(SISTER).NE.1.OR.LINKS.LT.2) GO TO 800
      DO 827 D = 1,HIBLOB
827      OLCLAN(D) = CLAN(D)
      DO 803 OTHERS = 2,HIBLOB
          IF(ROLL(OTHERS).LT.1) GO TO 803
          IF(CLAN(OTHERS).LT.CLAN(LOGAL))CLAN(LOGAL)=CLAN(OTHERS)
          CLAN(OTHERS) = CLAN(LOGAL)
803      CONTINUE
      DO 804 MEMBER =2,HIBLOB
          IF(ROLL(MEMBER).LT.1) GO TO 804
      DO 808 COUSIN=1,HIBLOB
          IF(CLAN(COUSIN).EQ.OLCLAN(MEMBER))CLAN(COUSIN)=CLAN(LOGAL)
808      CONTINUE
804      CONTINUE
800      CONTINUE
      IF(LINKS.GT.0) GO TO 701
      IA(COLUMN,ROW) = HIBLOB+1
      HIBLOB = HIBLOB + 1
      CLAN(HIBLOB) = HIBLOB
      IF(HIBLOB.GE.500) GO TO 8802
      GO TO 801
701      IA(COLUMN,ROW) = CLAN(LOGAL)
801      CONTINUE
      GO TO 900

8802      nomore = row
      WRITE(NP,8803) nomore
8803      FORMAT(' (TOO MANY CLUMPS; POINTS BELOW ROW ',I4,
1          ' WILL BE SET TO -1)')

900      CONTINUE
      HICLAN = 0
      Number = 1
      DO 8808 ROW = 1,NR
          if (row.ge.nomore) then
              do 8044 column = 1, nc
                  ib(column,row) = -1
8044              go to 8808
              endif
      DO 807 COLUMN = 1,NC
          IF(IA(COLUMN,ROW).LT.1) GO TO 817
          CELL = IA(COLUMN,ROW)
          XCLAN = CLAN(CELL)
          IF(DAM(XCLAN).GT.0) GO TO 805

```

```

        IB(COLUMN,ROW) = Number
        DAM(XCLAN) = Number
        Number = Number + 1
        GO TO 806
805      IB(COLUMN,ROW) = DAM(XCLAN)
        GO TO 807
817      IF(IA(COLUMN,ROW).EQ.0) IA(COLUMN,ROW) = 10
        GO TO 807
806      IF(DAM(XCLAN).GT.HICLAN) HICLAN = DAM(XCLAN)
807      CONTINUE
8808     continue
        WRITE(NP,002) HICLAN,HIBLOB
002      FORMAT(2H (,13,8H CLUMPS,,15,20H UPWARD CUSPS FOUND))
        RETURN
        END

1      SUBROUTINE REGRID(ic,ir,icl,JC,JR,JCL,ISIZE,ITOP,ILEFT,IBOTTM,
           IRIGHT,IBAK,JSIZE,JTOP,JLEFT,JBOTTM,JRIGHT,IA,IB)
        COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
        INTEGER*2 IA(ic,ir),IB(JC,JR)
        IF(IBAK.GT.0) GO TO 55
           CALL SETCOR(IB(1,1),NCL,0)
           GO TO 57
055      CALL FILE(JCL,IBAK,1,IB)
057      CONTINUE
        IF(ISIZE.LE.0) ISIZE = 1
        IF(ITOP.LE.0.OR.ITOP.GT.ir) ITOP = (ISIZE+1)/2
        IF(IBOTTM.LE.0.OR.IBOTTM.GT.ir) IBOTTM = ir
        IF(ILEFT.LE.0.OR.ILEFT.GT.ic) ILEFT = (ISIZE+1)/2
        IF(IRIGHT.LE.0.OR.IRIGHT.GT.ic) IRIGHT = ic
        IF(JSIZE.LE.0) JSIZE = 1
        IF(JTOP.LE.0.OR.JTOP.GT.NR) JTOP = (JSIZE+1)/2
        IF(JBOTTM.LE.0.OR.JBOTTM.GT.NR) JBOTTM = NR
        IF(JLEFT.LE.0.OR.JLEFT.GT.NC) JLEFT = (JSIZE+1)/2
        IF(JRIGHT.LE.0.OR.JRIGHT.GT.NC) JRIGHT = NC
        JROW = JTOP-JSIZE
        DO 150 IROW = ITOP,IBOTTM,ISIZE
           JROW = JROW + JSIZE
           IF(JROW.GT.JBOTTM) RETURN
           JCOL = JLEFT-JSIZE
           DO 100 ICOL = ILEFT,IRIGHT,ISIZE
              JCOL = JCOL + JSIZE
              IF(JCOL.GT.JRIGHT) GO TO 150
100          IB(JCOL,JROW) = IA(ICOL,IROW)
150      CONTINUE
        RETURN
        END

        SUBROUTINE INPUTS(JC,JR,IV,IA,ECHOED)
        COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
        INTEGER*2 IA(NC,NR), INPUT(25), R, C, COL, ROW
        INTEGER ECHOED,VALUE
        IF(IV.EQ.0) CALL SETCOR(IA(1,1),NCL,0)
        R = 1
        C = 1

```

```

        ICARD = 0
        IF(ECHOED.EQ.1) WRITE(NP,70)
070    FORMAT(54H (SPECIFY A 3-DIGIT ROW NUMBER FOLLOWED BY CONSECUTIVE,
1        17H PAIRS OF 3-DIGIT/35H "VALUE-THRU-COLUMN" NUMBERS IN THE,
1        33H 2ND THROUGH 25TH 3-DIGIT FIELDS./16H FINISH WITH A ,
1        17HROW NUMBER OF -1)/4H +++,12(6H---++))
080    READ(NI,100) (INPUT(N),N=1,25)
100    FORMAT(25I3)
        ICARD = ICARD + 1
        DO 105 N = 3,25,2
        IF(INPUT(N).NE.0) GO TO 106
105    CONTINUE
        GO TO 800
106    IF(ECHOED.EQ.1) WRITE(NP,075) (INPUT(N),N=1,25)
075    FORMAT(1H ,I3,1X,12(2I3))
        N = 1
        IF(INPUT(1).GT.NR) GO TO 600
        IF(INPUT(1).LE.0) GO TO 200
        R = INPUT(1)
C    USE THE FOLLOWING FOR ROW #1 AT BOTTOM OF MAP RATHER THAN AT TOP
C
        R = NR+1 - INPUT(1)
        COL = 0
        C = 0
        ICARD = 1
200    N = N+1
        IF(N.GE.25.OR.COL.GE.NC) GO TO 80
        VALUE = INPUT(N)
        N = N+1
        IF(INPUT(N).LE.0) GO TO 80
        IF(INPUT(N).GT.NC) INPUT(N) = NC
        IF(INPUT(N).LE.C) GO TO 600
        C = INPUT(N)
        IF(VALUE.EQ.998) GO TO 499
        IF(VALUE.NE.999) GO TO 400
300    COL = C
        GO TO 200
400    COL = COL + 1
        IA(COL,R) = VALUE
        IF(COL.GE.C) GO TO 200
        GO TO 400
499    IF(R.LE.1) GO TO 300
500    COL = COL+1
        IA(COL,R) = IA(COL,R-1)
        IF(COL.GE.C) GO TO 200
        GO TO 500
600    WRITE(NP,601) N,INPUT(N),ICARD,R
601    FORMAT(9H (ENTRY #,I3,1H:,I4,8H ON CARD,I2,7H OF ROW,I4,
1        36H IS OUT OF RANGE OR OUT OF SEQUENCE))
        GO TO 80
300    CONTINUE
        RETURN
        END

```

SUBROUTINE INSERT(ERRORS)

COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB

```

INTEGER ERRORS
WRITE(NP,1)
00001 FORMAT(42H NO, AN INSERT SUBROUTINE MUST BE PROVIDED,34(1H*))
ERRORS = 1
RETURN
END

SUBROUTINE INPUTR(JCL,IV,IA,ECHOED)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER*2 IA(JCL)
INTEGER BUF(1),ECHOED,C,R
CHARACTER*1 FMAT(80)
IF (IV.EQ.0) CALL SETCOR(IA(1),NCL,0)
IF(ECHOED.EQ.1) WRITE(NP,100)
100  FORMAT(' (SPECIFY FORTRAN INPUT FORMAT;  FOR EXAMPLE: ',
1      '(I3,2X,75I1)  )')
READ(NI,200) FMAT
200  FORMAT(80A1)
IF(ECHOED.EQ.1) WRITE(NP,300) FMAT
300  FORMAT(1H ,80A1)
IF(ECHOED.EQ.1) WRITE(NP,350)
350  FORMAT(53H (SPECIFY A ROW NUMBER FOLLOWED BY THE LEFT-TO-RIGHT ,
1      22HVALUES ACROSS EACH ROW/25H USING THE ABOVE FORMAT.,
1      26H FINISH WITH BLANK LINES))
400  READ(NI,FMAT) R, (BUF(C),C=1,NC)
IF(R.LE.0) RETURN
IF(R.GT.NR) GO TO 600
IF(ECHOED.EQ.1) WRITE(NP,450) R, (BUF(C),C=1,NC)
450  FORMAT(1H ,3HROW,I5/(5X,15I5))
N = (R-1) * NC
DO 500 C = 1,NC
N = N+1
500  IA(N) = BUF(C)
GO TO 400
600  WRITE(NP,650) R
650  FORMAT(24H (THERE IS NO ROW NUMBER,I4,1H))
GO TO 400
END

SUBROUTINE INPUTC(JCL,IV,IA,ECHOED)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IA(JCL)
CHARACTER*1 FMAT(80)
INTEGER ECHOED,C,R,NN
IF (IV.EQ.0) CALL SETCOR(IA(1),NCL,0)
IF(ECHOED.EQ.1) WRITE(NP,100)
100  FORMAT(53H (SPECIFY FORTRAN INPUT FORMAT;  FOR EXAMPLE: (3I3) ))
READ(NI,200) FMAT
200  FORMAT(80A1)
IF(ECHOED.EQ.1) WRITE(NP,300) FMAT
300  FORMAT(1H ,80A1)
IF(ECHOED.EQ.1) WRITE(NP,350)
350  FORMAT(51H (SPECIFY A ROW NUMBER FOLLOWED BY A COLUMN NUMBER,,
1      19H VALUE AND CARRIAGE/30H RETURN FOR EACH POINT USING ,

```

```

1          43H THE ABOVE FORMAT.  FINISH WITH BLANK LINES))
600  READ(NI,FMAT) R, C, N
      IF(R.LT.0) RETURN
      IF(R.LE.0.OR.R.GT.NR.OR.C.LE.0.OR.C.GT.NC) GO TO 700
      NN = ((R-1) * NC) + C
      IA(NN) = N
      IF(ECHOED.EQ.1) WRITE(NP,610) R, C, IA(NN)
610  FORMAT(1H+, ' ROW', I4, ', ' COLUMN', I4, ' = ', I5/)
      GO TO 600
700  WRITE(NP,750) R, C
750  FORMAT(26H (THERE IS NO POINT AT ROW, I4, 8H, COLUMN, I4, 1H))
      GO TO 600
      END

      SUBROUTINE TRACE(JC, JR, IV, ECHOED, WIDTH, XSTART, YSTART, IA, IB)
      COMMON/SAM/NP, NI, NCL, NC, NR, NMAPS, NBPM, NBUF, NWIDE, NCPB
      INTEGER*2 CSHIFT, COL, ROW, STOPCO, STOPRO, ENDCOL, IA(JC, JR),
1      IB(JC, JR)
      INTEGER ECHOED, XSTART, WIDTH, YSTART, A, B, C, D, CC, DD
      CHARACTER*1 FMAT(80)
      IF(WIDTH.LE.0) WIDTH = 10
      IF(IV.EQ.0) CALL SETCOR(IA(1,1), NCL, 0)
      NEW = 1
      LONE = 0
      CSHIFT = -1
      XSHIFT = 0
      IF(ECHOED.EQ.1) WRITE(NP,11)
011  FORMAT(46H (SPECIFY FORTRAN INPUT FORMAT;  FOR EXAMPLE: ,
1      9H(3I3)  ))
      READ(NI,112) FMAT
112  FORMAT(80A1)
      IF(ECHOED.EQ.1) WRITE(NP,113) FMAT
113  FORMAT(1H ,80A1)
      IF(ECHOED.EQ.1) WRITE(NP,001)
001  FORMAT(' (USING THE ABOVE FORMAT',
1      ' SPECIFY AN X (ROW) COORDINATE FOLLOWED BY A',
1      ' Y (COLUMN)'/ COORDINATE, VALUE, AND A CARRIAGE RETURN',
1      ' FOR EACH POINT TO BE TRACED.  USE '/
1      ' BLANK LINES TO INTERRUPT TRACING.  FINISH',
1      ' WITH AN X COORDINATE OF -1.)')
005  nval = newval
      READ(NI,FMAT) IC, ID, NEWVAL
      IF(ECHOED.EQ.1) WRITE(NP,006) IC, ID, NEWVAL
006  FORMAT(1X, 3I10)
      C = ID
      D = (NR*WIDTH) - IC
      IF(IC.NE.0.OR.ID.NE.0) GO TO 20
CCCCC 0,0 INPUT INDICATING END OF A CONTINUOUS SEQUENCE
      NEW = 1
      IF(LONE.EQ.0) GO TO 5
      LCOL = (A/WIDTH) + 1
      LROW = NR - (B/WIDTH)
      IF(ECHOED.EQ.1) WRITE(NP,19) LROW, LCOL, nval
019  FORMAT(' (ROW ', I4, ', COLUMN ', I4, ' = ', I5, ')')
      IF(LCOL.GT.0.AND.LCOL.LE.NC.AND.LROW.GT.0.AND.LROW.LE.NR)

```

```

1          IA(LCOL,LROW) = nval
          GO TO 5
CCCCCC NEGATIVE X COORDINATE INDICATING END OF TRACE LIST INPUT
020  IF(IC.LT.0) RETURN
CCCCCC OFFSET COORDINATES WITH RESPECT TO SPECIFIED ORIGIN
      C = C - YSTART
      D = D + XSTART
      IF(NEW.EQ.0) GO TO 30
CCCCCC BEGIN A NEW SEQUENCE OF POINTS
      LONE = 1
          NEW = 0
          A = C
          B = D
          GO TO 05
CCCCCC CONTINUE THE CURRENT SEQUENCE; REMEMBER THIS NEW POINT AS (CC,DD)
030  CC = C
      DD = D
      LONE = 0
      IF(D.GT.B) GO TO 40
CCCCCC POINT DOWNWARD LINES UPWARD BY SWITCHING (A,B) WITH (C,D)
      N = D
      D = B
      B = N
      N = A
      A = C
      C = N
CCCCCC COMPUTE THE SLOPE OF THIS SEGMENT, CATCHING VERTICAL SLOPES
040  IF(C.NE.A) GO TO 50
          SLOPE = 9999.9
          GO TO 60
050  SLOPE = (D-B+0.0) / (C-A+0.0)
CCCCCC COMPUTE CHANGE IN DIGITIZED X PER VERTICAL CHANGE OF 1 GRIDSPACE
060  IF(SLOPE.EQ.0.0) GO TO 65
      XSHIFT = (WIDTH/SLOPE)
065  IF(C.GT.A) GO TO 70
          CSHIFT = -1
          GO TO 80
070  CSHIFT = 1
CCCCCC (COL,ROW) = GRID CELL COORDINATES OF DIGITIZED POINT (A,B)
CCCCCC EACH CELL INCLUDING POINTS ON ITS TWO EDGES NEAREST THE ORIGIN
080  COL = (A/WIDTH) + 1
      ROW = NR - (B/WIDTH)
CCCCCC (STOPCO,STOPRO) = GRID CELL COORDINATES OF DIGITIZED POINT (C,D)
      STOPCO = (C/WIDTH) + 1
      STOPRO = NR - (D/WIDTH)
      IF(A.LT.0) COL = COL - 1
      IF(B.LT.0) ROW = ROW + 1
      IF(C.LT.0) STOPCO = STOPCO - 1
      IF(D.LT.0) STOPRO = STOPRO + 1
      IF(ECHOED.EQ.1) WRITE(NP,91) newval,ROW,COL,STOPRO,STOPCO
091  FORMAT(' (',i5,18H FROM ROW-COLUMN (,i3,1H,,i3,6H) TO (,i3,
1      1H,,i3,2H)))
CCCCCC INCREMENT THE VALUE OF ALL CELLS CROSSED BY LINE (A,B) TO (C,D)
CCCCCC BEGINNING WITH LOWERMOST CELL (COL,ROW). THAT LINE CROSSES THE
CCCCCC UPPER EDGE OF THIS ROW AT DIGITIZER COORDINATES (FRONTX,NORTHY)

```

```

NORTHY = (NR+1-ROW) * WIDTH
IF (SLOPE.NE.0.0) GO TO 100
    ENDCOL = STOPCO
    GO TO 250
100  FRONTX = ((NORTHY-B)/SLOPE) + A
CCCCC (ENDCOL,ROW) = END OF A HORIZONTAL SEQUENCE OF CELLS
200  IFX = FRONTX
    ENDCOL = (IFX/WIDTH) + 1
    IF(IFX.LT.0) ENDCOL = ENDCOL - 1
250  IF(ROW.LT.1) GO TO 400
CCCCC ASSIGN VALUES TO CELLS FROM (COL,ROW) TO (ENDCOL,ROW)
300  IF(COL.GT.0.AND.COL.LE.NC.AND.ROW.LE.NR)
    1      IA(COL,ROW) = NEWVAL
    IF((CSHIFT.EQ.-1.AND.COL.LE.STOPCO.AND.ROW.LE.STOPRO).OR.
    1      (CSHIFT.EQ. 1.AND.COL.GE.STOPCO.AND.ROW.LE.STOPRO)) GO TO 400
    IF(COL.EQ.ENDCOL) GO TO 350
    COL = COL + CSHIFT
    GO TO 300
CCCCC START ANOTHER HORIZONTAL SEQUENCE OF CELLS JUST ABOVE THIS ONE
350  FRONTX = FRONTX + XSHIFT
    ROW = ROW - 1
    GO TO 200
400  A = CC
    B = DD
    GO TO 05
END

```

```

SUBROUTINE SPREAD(JC,JR,JCL,ELMAP,FMAP,EXTENT,IU,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 NORTH,EAST,SOUTH,WEST,IA(JC,JR),IB(JC,JR)
INTEGER EXTENT,ELMAP,FMAP,TOP,BOTTOM,SLOPA,CANCEL,C,R,WES,
1      EAS,SOU,XTENT,EMPTY,TILT,ROW,COLUMN,HALO,SLOPE,X,Y,
1      RIGHT,UPDOWN
IF(EXTENT.LE.1.OR.EXTENT.GT.327) EXTENT= 4
IF(IU.EQ.1) GO TO 400
      UPDOWN = 1
      GO TO 450
400  UPDOWN = -1
450  CONTINUE
      IF(FMAP.GT.0) GO TO 698
      CANCEL = 0
      GO TO 699
698  CALL FILE(JCL,FMAP,1,IB)
C.....INTERPRET NEGATIVE SCALE FACTORS AS 1/100THS
      DO 697 C = 1,NC
      DO 697 R = 1,NR
      IF(IB(C,R).GT.0) IB(C,R) = IB(C,R) * (-100)
      IB(C,R) = IB(C,R) + 100
      IB(C,R) = -IB(C,R)
      IF(IB(C,R).LT.0) IB(C,R) = 0
697  CONTINUE
      CANCEL = 1
699  WEST = NC
      EAST = 1
      NORTH = NR
      SOUTH = 1
      DO 700 C = 1,NC
      IF(C.GT.EAST) EAST = C
      IF(C.LT.WEST) WEST = C
      DO 700 R = 1,NR
      IF(IA(C,R).GT.0) GO TO 500
      IB(C,R) = - ((IB(C,R)*CANCEL) +1)
      GO TO 700
500  IB(C,R) = IA(C,R)-1
      IF(R.GT.SOUTH) SOUTH = R
      IF(R.LT.NORTH) NORTH = R
700  CONTINUE
      CALL SETCOR(IA(1,1),NCL,0)
      IF(ELMAP.GT.0) CALL FILE(JCL,ELMAP,1,IA)
      RADIUS = 1
      LEFT = 0
      RIGHT = 0
      TOP = 0
      BOTTOM = 0
      WES = WEST
      EAS = EAST
      NOR = NORTH
      SOU = SOUTH
      EMPTY = 0
      XTENT = (EXTENT*2.5) - 1
      IHALF = XTENT/2
C.....INCREMENT EACH INCREASING PROXIMITY HALO

```

```

DO 802 HALO = 1,XTENT
IF(HALO.EQ.IHALF) WRITE(NP,150)
150  FORMAT(29H (SPREAD OPERATION HALF DONE))
      EMPTY = EMPTY+1
C.....LOOP THROUGH INCREASINGLY LARGE AREA AROUND SEARCH CORE
C.....STOPPING ONLY ON MOST RECENT HALO
      DO 801 TILT = 1,2
      DO 801 ROW = NORTH,SOUTH
      DO 801 COLUMN = WEST,EAST
        IF(IB(COLUMN,ROW).LT.(HALO-1)*40) GO TO 801
        IF(IB(COLUMN,ROW).GE.(HALO )*40) GO TO 801
        EMPTY = 0
        TOP = ROW - RADIUS
        IF(TOP.LT.1) TOP = 1
        BOTTOM = ROW + RADIUS
        IF(BOTTOM.GT. NR) BOTTOM = NR
        LEFT = COLUMN-RADIUS
        IF(LEFT.LT.1) LEFT = 1
        RIGHT = COLUMN + RADIUS
        IF(RIGHT.GT.NC) RIGHT = NC
C.....LOOP AROUND EACH CELL OF THE MOST RECENT HALO
C.....TO SET OUT NEW HALO
        DO 800 X = TOP,BOTTOM
        DO 800 Y = LEFT,RIGHT
          IF(TILT.EQ.1.AND.X.NE.ROW.AND.Y.NE.COLUMN) GO TO 800
          IF(IB(Y,X).GE.0) GO TO 800
          IF(IA(Y,X)*UPDOWN.LT. IA(COLUMN,ROW)*UPDOWN)GO TO 800
          STEP = 141
          IF(Y.EQ.COLUMN.OR.X.EQ.ROW) STEP = 100
C.....CONSIDER SLOPE RESTRICTION
C      IF(SLOPE.GE.0) GO TO 630
C      IF((IA(Y,X)-IA(COLUMN,ROW))*UPDOWN*100/STEP.GT.SLOPE) GO TO 800
C      GC TO 600
C 630 IF((IA(Y,X)-IA(COLUMN,ROW))*UPDOWN*100/STEP.LT.SLOPE) GO TO 800
C 600 CONTINUE
C.....ASSIGN EFFECTIVE PROXIMITY VALUE
      IB(Y,X) = IB(COLUMN,ROW)+STEP+((STEP*(-(IB(Y,X)+1)))/100)
      IF(Y.LT.WES) WES = Y
      IF(Y.GT.EAS) EAS = Y
      IF(X.LT.NOR) NOR = X
      IF(X.GT.SOU) SOU = X
800    CONTINUE
801    CONTINUE
      WEST = WES
      EAST = EAS
      NORTH = NOR
      SOUTH = SOU
802    CONTINUE
803    CONTINUE
      DO 809 Y=1,NC
      DO 809 X = 1,NR
        IF(IB(Y,X).LT.0.OR.IB(Y,X).GT.EXTENT*100) IB(Y,X)=EXTENT * 100
        IB(Y,X) = IB(Y,X)/100
809    CONTINUE
      RETURN

```

END

```

SUBROUTINE EXTEND(JC,JR,EXTENT,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IA(JC,JR),IB(JC,JR),NORTH,SOUTH,EAST,WEST,XSTEP,YSTEP
INTEGER EXTENT,C,R,RADIUS,XTENT,WES,EAS,SOU,CYCLE,X,Y,TOP,
1      BOTTOM,RIGHT,XBACK,YBACK,COLUMN,ROW
IF(EXTENT.LE.1.OR.EXTENT.GT.999) EXTENT = 4
WEST = NC
EAST = 1
NORTH = NR
SOUTH = 1
DO 700 C = 1,NC
DO 700 R = 1,NR
      IF(IA(C,R).GT.0) GO TO 500
      IB(C,R) = EXTENT**2
      IA(C,R) = -29999
      GO TO 700
500      IB(C,R) = 0
      IA(C,R) = 1
      IF(C.GT.EAST) EAST = C
      IF(C.LT.WEST) WEST = C
      IF(R.GT.SOUTH) SOUTH = R
      IF(R.LT.NORTH) NORTH = R
700      CONTINUE
      RADIUS = 1
      XTENT = EXTENT + 1
      NOR = NORTH
      SOU = SOUTH
      EAS = EAST
      WES = WEST
      IHALF = XTENT/2
      DO 802 CYCLE = 1,XTENT
      IF(CYCLE.EQ.IHALF) WRITE(NP,150)
150      FORMAT(29H (SPREAD OPERATION HALF DONE))
      DO 801 ROW = NORTH,SOUTH
      DO 801 COLUMN = WEST,EAST
      IF(IB(COLUMN,ROW).GE.EXTENT**2) GO TO 801
      IF(IA(COLUMN,ROW).NE.CYCLE) GO TO 801
      TOP = ROW - RADIUS
      IF(TOP.LT.1) TOP = 1
      BOTTOM = ROW + RADIUS
      IF(BOTTOM.GT. NR) BOTTOM = NR
      LEFT = COLUMN-RADIUS
      IF(LEFT.LT.1) LEFT = 1
      RIGHT = COLUMN + RADIUS
      IF(RIGHT.GT.NC) RIGHT = NC
      DO 800 X = TOP,BOTTOM
      DO 800 Y = LEFT,RIGHT
      IF(IB(Y,X).LE.IB(COLUMN,ROW)) GO TO 800
      YSTEP = 0
      XSTEP = 0
      IF(COLUMN.EQ.Y) GO TO 660
      YBACK = (2*COLUMN)-Y

```

```

        IF(YBACK.LT.1.OR.YBACK.GT.NC) GO TO 650
        YSTEP = IB(COLUMN,ROW) - IB(YBACK,ROW)
        IF(ystep.LE.0) GO TO 650
        YSTEP = YSTEP+2
        GO TO 660
650      IF(CYCLE.NE.1) GO TO 800
        YSTEP = 1
660      IF(X.EQ.ROW) GO TO 680
        XBACK = (2*ROW)-X
        IF(XBACK.LT.1.OR.XBACK.GT.NR) GO TO 670
        XSTEP = IB(COLUMN,ROW)
        XSTEP = XSTEP - IB(COLUMN,XBACK)
        IF(XSTEP.LE.0) GO TO 670
        XSTEP = XSTEP+2
        GO TO 680
670      IF(CYCLE.NE.1) GO TO 800
        XSTEP = 1
680      NEWVAL = XSTEP+YSTEP+IB(COLUMN,ROW)
        IF(NEWVAL.GE.IB(Y,X)) GO TO 800
            IB(Y,X) = NEWVAL
            IA(Y,X) = CYCLE+1
            IF(Y.LT.WES) WES = Y
            IF(Y.GT.EAS) EAS = Y
            IF(X.LT.NOR) NOR = X
            IF(X.GT.SOU) SOU = X
800      CONTINUE
801      CONTINUE
        WEST = WES
        NORTH = NOR
        EAST = EAS
        SOUTH = SOU
802      CONTINUE
        DO 809 Y=1,NC
        DO 809 X = 1,NR
809      IB(Y,X) = SQRT(IB(Y,X)+0.1)
        RETURN
        END

```

```

SUBROUTINE SLOPE (JC,JR,OPTION,CWIDTH,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IA(JC,JR), IB(JC,JR)
INTEGER EL(9), OPTION, CWIDTH
REAL SLOP(4)
IF(OPTION.LT.0.OR.OPTION.GT.3) OPTION = 0
IF(CWIDTH.LE.0) CWIDTH = NWIDE
DO 100 L = 1,NR
DO 100 K = 1,NC
    LTOP = L-1
    LBOT = L+1
    KLEFT = K-1
    KRIGHT = K+1
    N = 0
    DO 200 LL = LTOP,LBOT
    DO 200 KK = KLEFT,KRIGHT

```

```

      N = N+1
      IF(LL.LE.0.OR.LL.GT.NR.OR.KK.LE.0.OR.KK.GT.NC) GO TO 210
      EL(N) = IA(KK,LL)
      GO TO 200
210    EL(N) = -99
200    CONTINUE
      DO 220 N = 1,9
      IF(EL(N).NE.-99) GO TO 220
      NN = 10-N
      IF(EL(NN).EQ.-99) EL(NN) = EL(5)
      EL(N) = (EL(5)*2) - EL(NN)
220    CONTINUE
      IF(OPTION.NE.1) GO TO 290
      NN = -1
      SLOP(1) = 0.0
      SLOP(2) = 1.0
      DO 240 N = 1,9
      NN = -NN
      SLOP(2) = SLOP(2) + (0.4142*NN)
      SLOP(3) = (EL(5)-EL(N)) / (CWIDTH*SLOP(2))
      SLOP(4) = 1.0
      IF(SLOP(3).LT.0) SLOP(4) = -1.0
240    IF(SLOP(3)*SLOP(4).GT.SLOP(1)) SLOP(1)=SLOP(3)*SLOP(4)
      IB(K,L) = (SLOP(1)*100.0)
      GO TO 100
290    SOUTH = 0.0
      WEST = 0.0
      DO 300 N = 1,4
      NN = 10-N
      IF( (EL(N)-EL(5)) * (EL(5)-EL(NN)) .GE.0) GO TO 310
      SLOP(N) = 0.0
      EL(N) = 0
      GO TO 300
310    SLOP(N) = (EL(N)-EL(NN)+0.0) / (CWIDTH*2.0)
      EL(N) = 1
300    CONTINUE
      IF(EL(1)+EL(2)+EL(3).EQ.0.OR.EL(1)+EL(4)+EL(3).EQ.0)GO TO 400
      SOUTH = (SLOP(1)+SLOP(2)+SLOP(3)) / (EL(1)+EL(2)+EL(3))
      WEST = (SLOP(1)+SLOP(4)-SLOP(3)) / (EL(1)+EL(4)+EL(3))
      IF(OPTION.GT.1) GO TO 500
      IB(K,L) = (100.0*SQRT(SOUTH**2+WEST**2)) + 0.5
      GO TO 100
400    IF(OPTION.GT.0) GO TO 500
      IB(K,L) = 0
      GO TO 100
500    IF(SOUTH.NE.0.0.OR.WEST.NE.0.0) GO TO 600
      IB(K,L) = 361
      GO TO 700
      IB(K,L) = 0
      GO TO 100
600    KK = 2
      RADIANT = ATAN2(WEST,SOUTH)
      IF(RADIANT.GE.0.0) KK = 0
      IB(K,L) = 180.0-((RADIANT+(KK*3.141593)) * 57.295773)
      IF(IB(K,L).LT.0) IB(K,L) = IB(K,L)+360

```

```

700      IF(OPTION.EQ.2) GO TO 100
          IF(IB(K,L).GT.337.AND.IB(K,L).LE.360) IB(K,L) = 0
          IB(K,L) = ((IB(K,L)+22.5) / 45.0) + 1
100  CONTINUE
      RETURN
      END

```

```

SUBROUTINE PROFIL(JC,JR,ORIENT,OPTION,LIMIT,CWIDTH,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IA(JC,JR),IB(JC,JR)
INTEGER ORIENT,OPTION,LIMIT,CWIDTH,
1      ROW,COL,X,XX,YY,Y,BACK,FRONT
REAL BRISE,FRISE,WIDTH
IF(ORIENT.EQ.5.OR.ORIENT.GT.9.OR.ORIENT.LT.1) ORIENT = 1
IF(CWIDTH.LE.0) CWIDTH = NWIDE
WIDTH = CWIDTH * 1.0
IF( (ORIENT/2)*2 .GE. ORIENT ) GO TO 4050
      LIMIT = LIMIT*14142 / 10000
      WIDTH = WIDTH * 1.4142
4050 CONTINUE
DO 800 ROW = 1,NR
X = ROW-1+((2*ORIENT)/7)
XX = (2*ROW) - X
DO 800 COL = 1,NC
      IF(IA(COL,ROW).NE.-99) GO TO 80
          IB(COL,ROW) = -99
          GO TO 800
080  Y = COL-2+ORIENT-(((2*ORIENT)/7)*3)
      YY = (2*COL) - Y
      BACK = -29999
      IF(X.LT.1.OR.X.GT.NR.OR.Y.LT.1.OR.Y.GT.NC) GO TO 100
      IF(IA(Y,X).NE.-99) BACK = IA(Y,X)
100  FRONT = -29999
      IF(XX.LT.1.OR.XX.GT.NR.OR.YY.LT.1.OR.YY.GT.NC) GO TO 200
      IF(IA(YY,XX).NE.-99) FRONT = IA(YY,XX)
200  IF(BACK.NE.-29999) GO TO 400
      IF(FRONT.NE.-29999) GO TO 300
      BACK = IA(COL,ROW)
      FRONT = BACK
      GO TO 500
300  BACK = ( IA(COL,ROW)*2 ) - FRONT
      GO TO 500
400  IF(FRONT.NE.-29999) GO TO 500
      FRONT = ( IA(COL,ROW)*2 ) - BACK
500  IF(OPTION.NE.0) GO TO 600
      B = 1
      IF(IA(COL,ROW)-BACK.GT.LIMIT) B = 0
      IF(IA(COL,ROW)-BACK.LT.-LIMIT) B = 2
      F = 2
      IF(IA(COL,ROW)-FRONT.GT.LIMIT) F = 3
      IF(IA(COL,ROW)-FRONT.LT.-LIMIT) F = 1
      IB(COL,ROW) = (B*3) + F
      GO TO 800
600  BRISE = BACK - IA(COL,ROW) + 0.0

```

```

      FRISE = FRONT - IA(COL,ROW) + 0.0
      IB(COL,ROW) = (ATAN2(WIDTH,BRISE)*57.296)+
1             (ATAN2(WIDTH,FRISE)*57.296)
800 1 CONTINUE
      RETURN
      END

SUBROUTINE SURVEY(JC,JR,OPTION,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON AREA(500), POINTS(500), NOOKS(500), EDGE(500)
INTEGER*2 IA(JC,JR), IB(JC,JR)
INTEGER AREA,POINTS,NOOKS,
1     EDGE, SQUARE(9),OPTION,ROW,COLUMN,RIGHT, BLOB(9),
1     EARTH,PEGS,SW,SE,TOP,BOTTOM,X,Y
WRITE(NP,100)
100 1 FORMAT(19X,' VALUE',5X,'POINTS',4X,'EDGES',3X,
1     'CONVEXITY',3X,'EULER')
DO 400 N = 1,500
      AREA(N) = 0
      POINTS(N) = 0
      NOOKS(N) = 0
      EDGE(N) = 0
400 CONTINUE
DO 666 ROW = 1,NR
DO 666 COLUMN = 1,NC
      TOP = ROW-1
      IF(TOP.LT.1)TOP = 1
      BOTTOM = ROW+1
      IF(BOTTOM.GT.NR) BOTTOM = NR
      LEFT = COLUMN-1
      IF(LEFT.LT.1) LEFT = 1
      RIGHT = COLUMN+1
      IF(RIGHT.GT.NC) RIGHT = NC
      EARTH = IA(COLUMN,ROW)+1
      IF(EARTH.GT.0.AND.EARTH.LE.500) GO TO 450
      IA(COLUMN,ROW) = -99
      IB(COLUMN,ROW) = -99
      GO TO 666
450 AREA(EARTH) = AREA(EARTH)+1
DO 554 N = 1,9
      BLOB(N) = -99
554 SQUARE(N) = 1
      PEGS = 0
      NW = 0
      NE = 0
      SW = 0
      SE = 0
      DO 555 X = TOP,BOTTOM
      DO 555 Y = LEFT,RIGHT
      N = ((Y-COLUMN)+2) + (((X-ROW)+1)*3)
      SQUARE(N) = 0
      BLOB(N) = -(1000+N)
      IF(IA(Y,X).EQ.IA(COLUMN,ROW)) GO TO 553
      SQUARE(N) = 1

```

```

                                BLOB(N) = IA(Y,X)+1
553      CONTINUE
                                EDGE(EARTH) = EDGE(EARTH)+SQUARE(N)
                                PEGS = PEGS+1
555      CONTINUE
                                EDGE(EARTH) = EDGE(EARTH)+9-PEGS
C.....FOR TOPOLOGY, CALCULATE # OF INTERIOR AND EXTERIOR CORNE
C      IF(COLUMN.GE.43.AND.COLUMN.LE.47.AND.ROW.LE.8)
C      1  WRITE(NP,66666) BLOB, SQUARE,COLUMN,ROW
C      66666  FORMAT(1X,9I6)
                                MOON = BLOB(2)
                                IF(MOON.LT.1) GO TO 557
                                    IF(BLOB(2).EQ.BLOB(4)) NOOKS(MOON) = NOOKS(MOON)+1
                                    IF(BLOB(2).EQ.BLOB(6)) NOOKS(MOON) = NOOKS(MOON)+1
C      IF(MOON.EQ.2) WRITE(NP,33333) COLUMN,ROW,NOOKS(MOON)
557      MOON = BLOB(8)
                                IF(MOON.LT.1) GO TO 556
                                    IF(BLOB(6).EQ.BLOB(8)) NOOKS(MOON) = NOOKS(MOON)+1
                                    IF(BLOB(4).EQ.BLOB(8)) NOOKS(MOON) = NOOKS(MOON)+1
556      CONTINUE
C      IF(MOON.EQ.2) WRITE(NP,33333) COLUMN,ROW,NOOKS(MOON)
                                NW = SQUARE(1)*SQUARE(2)*SQUARE(4)
                                NE = SQUARE(2)*SQUARE(3)*SQUARE(6)
                                SW = SQUARE(4)*SQUARE(7)*SQUARE(8)
                                SE = SQUARE(6)*SQUARE(8)*SQUARE(9)
                                POINTS(EARTH) = POINTS(EARTH)+NW+NE+SW+SE
C      IF(EARTH.EQ.2) WRITE(NP,44444) COLUMN,ROW,POINTS(EARTH)
666      CONTINUE
                                IF(OPTION.EQ.1) GO TO 668
                                    DO 667 ROW = 1,NR
                                        DO 667 COLUMN = 1,NC
                                            IF(IA(COLUMN,ROW).EQ.-99) GO TO 667
                                                EARTH = IA(COLUMN,ROW)+1
                                                IB(COLUMN,ROW) = ((SQRT(AREA(EARTH)+0.0))*700)
                                                    / EDGE(EARTH)
1
667      CONTINUE
                                GO TO 670
668      CONTINUE
C..... ASSIGN TOPOLOGY VALUE
                                DO 669 ROW = 1,NR
                                    DO 669 COLUMN = 1,NC
                                        IF(IA(COLUMN,ROW).EQ.-99) GO TO 669
                                            EARTH = IA(COLUMN,ROW)+1
                                            IB(COLUMN,ROW) = (NOOKS(EARTH)-(POINTS(EARTH)-4))/ 4
669      CONTINUE
670      CONTINUE
                                DO 700 EARTH = 1,500
                                    IF(AREA(EARTH).LE.0) GO TO 700
                                        N = EARTH-1
                                        TOP = ((SQRT(AREA(EARTH)+0.0))*700) / EDGE(EARTH)
                                        RIGHT = (NOOKS(EARTH)-(POINTS(EARTH)-4))/ 4
                                        WRITE(NP,701) N,AREA(EARTH),EDGE(EARTH),TOP,RIGHT
701      FORMAT(5H      ,10X,5I10)
700      CONTINUE
                                RETURN

```

END

```

SUBROUTINE SPAN(JC,JR,JCL,SMAP,EXTENT,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
INTEGER*2 IA(JC,JR),IB(JC,JR)
INTEGER SMAP,EXTENT,RADIUS,ROW,COLUMN,RIGHT,X,Y,T,V,A,B,SLIDE
REAL DIAMON,EDGE,SHORT,SHIFT
IF(SMAP.EQ.0) GO TO 600
      CALL FILE(JCL,SMAP,1,IB)
      GO TO 650
600  CALL SETCOR(IB(1,1),NCL,1)
650  IF(EXTENT.LE.0.OR.EXTENT.GT.19) EXTENT = 4
      RADIUS = 1
      DO 802 ROW = 1,NR
      WRITE(NP,150) ROW
150  FORMAT(21H (SPAN PROCESSING ROW,I4,1H))
      DO 801 COLUMN = 1,NC
      MMM = IB(COLUMN,ROW)
      IB(COLUMN,ROW) = 0
      IF(MMM.EQ.0) GO TO 801
      SHORT = EXTENT
706  LEFT = COLUMN-RADIUS
      IF(LEFT.LT.1) LEFT = 1
      RIGHT = COLUMN + RADIUS
      IF(RIGHT.GT.NC) RIGHT = NC
      T = -1
      DO 800 Y = LEFT,RIGHT
700  T = -T
      X = ROW+(T*(RADIUS-IABS(COLUMN-Y)))
      IF(X.LT.1.OR.X.GT.NR) GO TO 709
      IF(IA(Y,X).EQ.IA(COLUMN,ROW))GO TO 703
709  V = -1
      IF(ROW.GT.X) V=1
      IF(ROW.EQ.X) V=0
      H = -1
      IF(COLUMN.GT.Y)H=1
      IF(COLUMN.EQ.Y)H=0
      A = ROW
      Z = -1
704  Z = Z+1
      IF(ROW.EQ.X) GO TO 707
      A = A+V
708  SHIFT = ((A-ROW)*(COLUMN-Y)*1.0)/ ((ROW-X)*1.0)
      SLIDE = ABS(SHIFT)+0.5
      B = COLUMN+(H*SLIDE)
      GO TO 701
707  B = ((2*COLUMN) - Y) + (Z*H)
701  IF(Z.GT.EXTENT) GO TO 703
      IF(A.LE.0.OR.A.GT.NR.OR.B.LE.0.OR.B.GT.NC) GO TO 702
      EDGE = 1.0
      IF(IA(B,A).NE.IA(COLUMN,ROW)) GO TO 603
      GO TO 704
702  CONTINUE
      EDGE = 2.0

```

```

603      CONTINUE
        SHIFT = (((A-X) -V) **2) + (((B-Y) -H) **2)
        IF(SHIFT.LE.0.0) SHIFT = 0.5
        IF(EDGE*SQRT(SHIFT).GE.SHORT) GO TO 680
        SHORT = EDGE * SQRT(SHIFT)
        IF(A.LE.0.OR.A.GT.NR.OR.B.LE.0.OR.B.GT.NC.OR.
1          X.LE.0.OR.X.GT.NR.OR.Y.LE.0.OR.Y.GT.NC) GO TO 680
        IF(IA(B,A).EQ.IA(Y,X)) POLES = 1
680      CONTINUE
703      CONTINUE
        IF(T.EQ.1.AND.Y.NE.LEFT.AND.Y.NE.RIGHT) GO TO 700
        T = -1
800      CONTINUE
        DIAMON = RADIUS-1.0
        IF(SHORT.LE.((SQRT(2.0))*(DIAMON))) GO TO 889
        RADIUS = RADIUS + 1
        GO TO 706
889      CONTINUE
        IB(COLUMN,ROW) = SHORT
801      RADIUS = 1
802      CONTINUE
        RETURN
        END

```

```

SUBROUTINE SIZE(JCL,IA)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER*2 IA(JCL)
INTEGER BUF(1), VALUE
DO 100 N = 1,NBUF
100  BUF(N) = 0
DO 200 N = 1,NCL
    IF(IA(N).LT.1.OR.IA(N).GT.2000) GO TO 200
    VALUE = IA(N)
    BUF(VALUE) = BUF(VALUE) + 1
200  CONTINUE
DO 300 N = 1,NCL
    IF(IA(N).GE.1.AND.IA(N).LE.2000) GO TO 250
    IA(N) = 0
    GO TO 300
250  VALUE = IA(N)
    IA(N) = BUF(VALUE)
300  CONTINUE
    RETURN
    END

```

```

SUBROUTINE SLICE(JCL,MIN,MAX,LEVELS,IA,IB)
COMMON/SAM/NP,NI,NCL,NC,NR,NMAPS,NBPM,NBUF,NWIDE,NCPB
COMMON BUF
INTEGER*2 IA(JCL),IB(JCL)
INTEGER BUF(1),C,START,STOP,VALUE
REAL NEXCUT, WIDTH
N = NBUF - 2

```

```

IF(LEVELS.GT.N) LEVELS = N
IF(LEVELS.LE.0) LEVELS = 4
DO 50 N = 1,NBUF
050 BUF(N) = 0
IF(MAX.GT.-32767) GO TO 200
    DO 100 C = 1,NCL
100     IF(IA(C).GT.MAX) MAX = IA(C)
200 IF(MIN.LT.32767) GO TO 275
    DO 250 C = 1,NCL
250     IF(IA(C).LT.MIN) MIN = IA(C)
IF(MAX.GE.MIN) GO TO 275
    N = MAX
    MAX = MIN
    MIN = N
275 WIDTH = (MAX-MIN) / LEVELS
IF(WIDTH.LE.0.0) WIDTH = 1.0
ITOP = -32767
IBOTTM = 32767
DO 300 C = 1,NCL
    IB(C) = (((IA(C)-MIN)-1)/ WIDTH) + 1
    IF(IA(C).LT.IBOTTM) IBOTTM = IA(C)
    IF(IA(C).GT.ITOP) ITOP = IA(C)
    IF(IB(C).LT.1) IB(C) = 0
    IF(IB(C).GT.LEVELS) IB(C) = LEVELS + 1
    N = IB(C) + 1
300 BUF(N) = BUF(N) + 1
NEXCUT = MIN
START = IBOTTM
L = LEVELS + 2
DO 600 LEVNUM = 1,L
    IF(BUF(LEVNUM).EQ.0) GO TO 550
    IF(LEVNUM.EQ.L) NEXCUT = ITOP
    N = LEVNUM - 1
480 WRITE(NP,500) N, BUF(LEVNUM), START, NEXCUT
500 FORMAT(5X,I6,12H ASSIGNED TO,I6,21H CELLS WITH VALUES OF,
1      I6,5H THRU,F6.0)
550 START = NEXCUT + 1
600 NEXCUT = NEXCUT + WIDTH

RETURN
END

```

การ INPUT ข้อมูลโดยใช้โปรแกรม GISENTRY

1. เริ่มใช้พิมพ์ GISENTRY
2. ตั้งชื่อ FILE ลุ่มน้ำที่ต้องการเก็บข้อมูล โดยมี FILE TYPE เป็น ".IN" เช่น HUAIYANG.IN
3. INPUT ข้อมูลครั้งแรกพิมพ์ 1
4. INPUT จำนวน ROWS และจำนวน COLUMNS ของลุ่มน้ำที่ต้องการเก็บข้อมูล เช่น มี 35 ROWS x 20 COLUMNS ให้พิมพ์ 35, 20
5. ตั้งชื่อแผนที่ (MAP LAYER) ที่ต้องการเก็บข้อมูลในลุ่มน้ำ เช่น LANDUSE
6. พิมพ์ 1 สำหรับข้อมูลที่แทน GRID CELL และพิมพ์ 2 สำหรับข้อมูลที่แทนจุด (POINT) เช่น ถนน ลำน้ำ เป็นต้น
7. INPUT ข้อมูลในแต่ละ ROW ให้พิมพ์เลขที่ของ ROW ลงไปสำหรับ ROW ที่ต้องการ INPUT
8. INPUT ข้อมูล COLUMN เริ่มต้น, COLUMN ที่สิ้นสุดและค่าของข้อมูลที่อยู่ในช่วงนี้
9. ทำข้อ 8 ซ้ำกระทั่ง INPUT ข้อมูลหมด ROW ให้พิมพ์ 0,0,0 และเริ่ม INPUT ข้อมูลใน ROW อื่น ๆ ต่อไป
10. ทำซ้ำจาก 7) ถึง 9) กระทั่ง INPUT ข้อมูลหมดทุก ROW เรียบร้อย ให้พิมพ์ -1 เป็นการ INPUT ข้อมูลแผนที่ (MAP LAYER) เสร็จ 1 แผ่น
11. พิมพ์ 1 เมื่อต้องการ INPUT ข้อมูลแผนที่ (MAP LAYER) อื่น ๆ ต่อไป และให้ดำเนินการตามข้อ 1 - 10
12. พิมพ์ 2 เมื่อต้องการสิ้นสุดการ INPUT ข้อมูล

GISENTRY

GISENTRY - Make data base creation file for PC MAP
Remote Sensing Center, Asian Institute of Technology
Version 2.0 June 1986

NAME OF OUPUT FILE TO BE CREATED?
HUIIYANG.IN

ENTER RECORD NUMBER TO START WITH. IF THIS IS THE
FIRST TIME YOU HAVE ENTERED DATA FOR THIS STUDY AREA,
ENTER RECORD NUMBER 1 (INTEGER):

1

HOW MANY ROWS AND COLUMNS IN YOUR DATA SET
(INTEGER,INTEGER)?
35,20

ENTER NAME FOR NEXT MAP LAYER TO BE CODED.
(MAXIMUM 32 CHARACTERS):
LANDUSE

ENTER 1 TO CODE GRID DATA, 2 TO CODE POINT DATA
(USE POINT DATA FOR ROADS, ETC.) (INTEGER):

1

ENTER ROW NUMBER (INTEGER)
(ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

1

ROW 1: ENTER STARTING AND ENDING COLUMN NUMBERS,
AND VALUE TO ASSIGN TO THIS RANGE
(ENTER 0,0,0 TO CHOOSE NEW ROW)
(INTEGER, INTEGER, INTEGER):
5,13,1

ROW 1: ENTER STARTING AND ENDING COLUMN NUMBERS,
AND VALUE TO ASSIGN TO THIS RANGE
(ENTER 0,0,0 TO CHOOSE NEW ROW)
(INTEGER, INTEGER, INTEGER):
14,17,2

ROW 1: ENTER STARTING AND ENDING COLUMN NUMBERS,
AND VALUE TO ASSIGN TO THIS RANGE
(ENTER 0,0,0 TO CHOOSE NEW ROW)
(INTEGER, INTEGER, INTEGER):
0,0,0

ENTER ROW NUMBER (INTEGER)
(ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

ENTER ROW NUMBER (INTEGER)
 (ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
 FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

2

ROW 2: ENTER STARTING AND ENDING COLUMN NUMBERS,
 AND VALUE TO ASSIGN TO THIS RANGE
 (ENTER 0,0,0 TO CHOOSE NEW ROW)
 (INTEGER, INTEGER, INTEGER):

4,11,1

ROW 2: ENTER STARTING AND ENDING COLUMN NUMBERS,
 AND VALUE TO ASSIGN TO THIS RANGE
 (ENTER 0,0,0 TO CHOOSE NEW ROW)
 (INTEGER, INTEGER, INTEGER):

12,18,2

ROW 2: ENTER STARTING AND ENDING COLUMN NUMBERS,
 AND VALUE TO ASSIGN TO THIS RANGE
 (ENTER 0,0,0 TO CHOOSE NEW ROW)
 (INTEGER, INTEGER, INTEGER):

0,0,0

ENTER ROW NUMBER (INTEGER)
 (ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
 FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

3

ROW 3: ENTER STARTING AND ENDING COLUMN NUMBERS,
 AND VALUE TO ASSIGN TO THIS RANGE
 (ENTER 0,0,0 TO CHOOSE NEW ROW)
 (INTEGER, INTEGER, INTEGER):

ENTER ROW NUMBER (INTEGER)
 (ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
 FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

35

ROW 35: ENTER STARTING AND ENDING COLUMN NUMBERS,
 AND VALUE TO ASSIGN TO THIS RANGE
 (ENTER 0,0,0 TO CHOOSE NEW ROW)
 (INTEGER, INTEGER, INTEGER):

5,7,1

ROW 35: ENTER STARTING AND ENDING COLUMN NUMBERS,
AND VALUE TO ASSIGN TO THIS RANGE
(ENTER 0,0,0 TO CHOOSE NEW ROW)
(INTEGER, INTEGER, INTEGER):

0,0,0

ENTER ROW NUMBER (INTEGER)
(ENTER 0 TO STOP FOR A WHILE AND RESUME DATA ENTRY LATER
FOR THE CURRENT LAYER, -1 TO FINISH THIS LAYER)

-1

ENTER 1 TO CODE NEW ENTITY, 2 TO STOP (INTEGER):

2

PLEASE REMEMBER THE LAST RECORD NUMBER
IF YOU PLAN TO CONTINUE DATA ENTRY LATER.

*** THE LAST RECORD NUMBER WAS: 20

D>

วิธีการใช้โปรแกรม The Map Analysis Package

1. เริ่มใช้พิมพ์ MAP
2. ตั้งชื่อ File ลุ่มน้ำที่ต้องการ RUN (ไม่เกิน 2 ตัวอักษร)
3. INPUT จำนวนแถว (ROWS) ที่ใช้ในลุ่มน้ำ
4. INPUT จำนวนหลัก (COLUMNS) ที่ใช้ในลุ่มน้ำ
5. INPUT ขนาดความกว้างของ grid cell (scale)
6. ใช้คำสั่ง READ อ่านข้อมูลที่สร้างขึ้นโดยโปรแกรม GISENTRY (ไม่ต้องพิมพ์ file type)
7. การประมวลผลข้อมูลที่ใช้คำสั่ง (operation) ตาม FLOWCHART ในรูปที่ 4.2 ซึ่งรายละเอียดของคำสั่งต่าง ๆ ได้อธิบายไว้ในโปรแกรมในที่นี้จะอธิบายเฉพาะคำสั่งที่ใช้เป็นพื้นฐาน พอสรุปได้ดังนี้
 - คำสั่ง LIST จะแสดงรายชื่อแผนที่ต่าง ๆ (MAP LAYER) ที่มีอยู่ใน File ที่กำลังประมวลผล

วิธีใช้ พิมพ์ LIST เช่น LIST
 - คำสั่ง DISPLAY จะแสดงรายละเอียดของแผนที่ (MAP LAYER) ที่ต้องการดูรายละเอียด

วิธีใช้ พิมพ์ DISPLAY [ชื่อ MAP LAYER]

เช่น DISPLAY LANDUSE
 - คำสั่ง EXPLAIN จะอธิบายความหมายและรายละเอียดของแต่ละคำสั่ง (operation)

วิธีใช้ พิมพ์ EXPLAIN [ชื่อคำสั่ง (operation)]

เช่น EXPLAIN ADD
8. วิธีใช้และรายละเอียดเพิ่มเติม ดู Tomlin, C.D. (1984)

MAP

D>ECHO OFF
(LOADING MAP EXPLANATION FILES)
File not found
(LOADING MAP INSTRUCTION FILES)
(LOADING MAP DATA FILES)
File not found
(LOADING MAP PROGRAM)

AVAILABLE DATA: MAP NAME # OF OVERLAYS

PLEASE ENTER MAP NAME:

HY

"HY" MUST BE NEW MAP

HOW MANY (horizontal) ROWS SHOULD IT HAVE?

35

• HOW MANY (vertical) COLUMNS? (no more than 932)

20

HOW WIDE (IN GEOGRAPHIC UNITS SUCH AS FEET OR METERS) IS EACH ROW OR COLUMN?

250

The Map Analysis Package

(C) Copyright 1986
by The President and Fellows of Harvard College
All Rights Reserved

OK, MAP ANALYSIS PACKAGE READY

READ HUIYANG

OK, INSTRUCTIONS NOW BEING READ FROM FILE "HUAIYANG.IN "

```
(SPECIFY FORTRAN INPUT FORMAT;  FOR EXAMPLE: (I3,2X,75I1)  )
(16I15/16I5/16I5/16I5/16I5/16I5/16I5/16I5/16I5/5I5)
(SPECIFY A ROW NUMBER FOLLOWED BY THE LEFT-TO-RIGHT VALUES ACROSS EACH ROW
USING THE ABOVE FORMAT.  FINISH WITH BLANK LINES)
```

[illegible]

ROW	35													
	0	0	0	0	1	1	1	0	0	0	0	0	0	0
	0	0	0	0	0									

OK, "LANDUSE " CREATED BY GRIDDING

220

NO, THERE IS NO INSTRUCTION FILE CALLED "ON.IN
INSTRUCTION FILES CURRENTLY AVAILABLE INCLUDE:

HUAIYANG
NKO

(INSTRUCTION CANCELLED AT "READ ")

(FOR AN EXPLANATION, ENTER "EXPLAIN")

(MISSING COMMAND INTERPRETED AS 'STOP')

OK, MAP ANALYSIS PACKAGE STOPPED

Returning from MAP to DOS

D>

MAP

D>ECHO OFF
(LOADING MAP EXPLANATION FILES)
File not found
(LOADING MAP INSTRUCTION FILES)
(LOADING MAP DATA FILES)
(LOADING MAP PROGRAM)

AVAILABLE DATA: MAP NAME # OF OVERLAYS

 HY 1

PLEASE ENTER MAP NAME:

HY

(LOADING HY OVERLAYS)

Number of Rows: 35
Number of Columns: 20
Standard Scale: 250

Overlay Title	Scale	Protection Status
LANDUSE	250	E

The Map Analysis Package

(C) Copyright 1986
by The President and Fellows of Harvard College
All Rights Reserved

OK, MAP ANALYSIS PACKAGE READY

LIST

MAP NAME: HY
NUMBER OF ROWS: 35
NUMBER OF COLUMNS: 20
STANDARD SCALE: 250

OVERLAY TITLE	PROTECTION STATUS	SCALE
LANDUSE	E	250

EXPLAIN

223

Explanation files are currently available on the following topics:

STRIP
BASICS
ADD
ASSIGN
AVERAGE
CLUMP
COPY
COVER
CROSS
DESCRIBE
DIFFEREN
DISPLAY
DIVIDE
DRAIN
ECHO
EULER
EXPLAIN
EXPONENT
EXPOSE
GRID
LABEL
LIST
MAP
MAXIMIZE
MINIMIZE
MULTIPLY
NOTE
ORIENT
POINT
PRINT
PROFILE
PROTECT
QUIET
RADIATE
READ
RECODE
RESPACE
SCALE
SCAN
SCORE
SHOW
SIZE
SLICE
SORT
SPAN
SPREAD
STOP
SUBTRACT
SURVEY
TRACE
WRITE
ZAP

5

For help, enter EXPLAIN followed by the name of one of the topics listed above.
For example, EXPLAIN ZAP

ตัวอย่างการคำนวณ HYDROGRAPH เมื่อทราบ Excess Rainfall
และ Isochrone

SCORE TC6HR BY EXCE11 TOTAL

MAP	MAP REGIONS	# OF CELLS	% OF TOTAL AREA	SCORE	% OF TOTAL SCORE	AVERAGE SCORE
TC6HR	0	1797	55.3	2	.4	.00
TC6HR	1	27	.8	6	1.3	.22
TC6HR	2	41	1.3	8	1.7	.20
TC6HR	3	36	1.1	10	2.1	.28
TC6HR	4	68	2.1	79	16.9	1.16
TC6HR	5	71	2.2	8	1.7	.11
TC6HR	6	54	1.7	23	4.9	.43
TC6HR	7	62	1.9	18	3.9	.29
TC6HR	8	82	2.5	13	2.8	.16
TC6HR	9	129	4.0	43	9.2	.33
TC6HR	10	107	3.3	43	9.2	.40
TC6HR	11	124	3.8	42	9.0	.34
TC6HR	12	119	3.7	47	10.1	.39
TC6HR	13	202	6.2	63	14.6	.34
TC6HR	14	134	4.1	17	3.6	.13
TC6HR	15	108	3.3	32	6.9	.30
TC6HR	16	56	1.7	6	1.3	.11
TC6HR	17	18	.6	2	.4	.11
TC6HR	18	15	.5	0	.0	.00
TOTAL PER OVERLAY		3250		467		.14

MAP	MAP REGIONS	# OF CELLS	% OF TOTAL AREA	SCORE	% OF TOTAL SCORE	AVERAGE SCORE
EXCE11	0	3090	95.1	13368	89.9	4.33
EXCE11	2	139	4.3	1316	8.8	9.47
EXCE11	9	21	.6	191	1.3	9.10
TOTAL PER OVERLAY		3250		14875		4.58

MAJOR OVERLAY REGIONS	MINOR OVERLAY REGIONS	# OF TOTAL POINTS	% OF AREA	% OF MAJOR AREA	% OF MINOR AREA	% OF OVERLAP	
0 _____	0 _____	1796	55.3	99.9	58.1	79.03	
0 _____	2 : 0.2	1	0.0	0.0	.7	.39	$0.2 \times 1000 \times 2.5 = 500$
1 _____	0 _____	24	7.7	88.9	1.8	44.83	
1 _____	2 : 0.2	3	0.0	11.1	2.2	6.63	$0.2 \times 3000 \times 2.5 = 3000$
2 _____	0 _____	37	11.1	90.2	1.2	45.72	
2 _____	2 : 0.2	4	1.1	9.8	2.9	6.32	$0.2 \times 4000 \times 2.5 = 5000$
3 _____	0 _____	31	1.0	86.1	1.0	43.56	
3 _____	2 : 0.2	5	1.2	13.9	3.6	8.74	$0.2 \times 5000 \times 2.5 = 6250$
4 _____	0 _____	53	1.6	77.9	1.7	39.83	
4 _____	2 : 0.2	8	.2	11.8	5.8	8.76	$0.2 \times 8000 \times 2.5 = 10000$
4 _____	9 : 0.7	7	.2	10.3	33.3	21.81	$0.9 \times 7000 \times 2.5 = 37375$
5 _____	0 _____	67	2.1	94.4	2.2	48.27	
5 _____	2 : 0.2	4	1.1	5.6	2.9	4.26	$0.2 \times 4000 \times 2.5 = 5000$
6 _____	0 _____	46	1.4	85.2	1.5	43.34	
6 _____	2 : 0.2	7	.2	13.0	5.0	9.00	$0.2 \times 7000 \times 2.5 = 8750$
6 _____	9 : 0.7	1	0.0	1.9	4.8	3.31	$0.9 \times 10000 \times 2.5 = 86250$
7 _____	0 _____	53	1.6	85.5	1.7	43.60	
7 _____	2 : 0.2	9	1.3	14.5	6.5	10.50	$0.2 \times 9000 \times 2.5 = 11250$
8 _____	0 _____	79	2.4	96.3	2.6	49.45	
8 _____	2 : 0.2	2	0.0	2.4	1.4	1.94	$0.2 \times 2000 \times 2.5 = 2500$
8 _____	9 : 0.7	1	0.0	1.2	4.8	2.99	$0.7 \times 10000 \times 2.5 = 86250$
9 _____	0 _____	111	3.4	86.0	3.6	44.82	
9 _____	2 : 0.2	17	.5	13.2	12.2	12.70	$0.2 \times 17000 \times 2.5 = 21250$
9 _____	9 : 0.7	1	0.0	.8	4.8	2.77	$0.9 \times 10000 \times 2.5 = 86250$

10	0	89	2.7	85.2	2.9	43.03	$0.28 \times 1122.5 = 212.3$	212.3 m^3
10	2 ± 0.2	17	1.5	15.9	12.2	14.06	$0.28 \times 1122.5 = 212.3$	
10	9 ± 0.9	1	1.0	1.9	4.8	2.85	$0.28 \times 1122.5 = 212.3$	
11	0	110	3.4	88.7	3.6	46.13	$0.28 \times 1122.5 = 212.3$	212.3 m^3
11	2 ± 0.2	12	1.4	9.7	8.6	9.16	$0.28 \times 1122.5 = 212.3$	
11	9 ± 0.9	2	1.0	1.6	9.5	5.57	$0.28 \times 1122.5 = 212.3$	
12	0	99	3.0	83.2	3.2	43.20	$0.28 \times 1122.5 = 212.3$	212.3 m^3
12	2 ± 0.2	19	1.6	16.0	13.7	14.82	$0.28 \times 1122.5 = 212.3$	
12	9 ± 0.9	1	1.0	1.8	4.8	2.80	$0.28 \times 1122.5 = 212.3$	
13	0	182	5.6	90.1	5.9	47.99	$0.28 \times 1122.5 = 212.3$	212.3 m^3
13	2 ± 0.2	16	1.5	7.9	11.5	9.72	$0.28 \times 1122.5 = 212.3$	
13	9 ± 0.9	4	1.1	2.0	19.0	10.51	$0.28 \times 1122.5 = 212.3$	
14	0	129	4.0	96.3	4.2	50.22	$0.28 \times 1122.5 = 212.3$	212.3 m^3
14	2 ± 0.2	4	1.1	3.0	2.9	2.93	$0.28 \times 1122.5 = 212.3$	
14	9 ± 0.9	1	1.0	1.7	4.8	2.75	$0.28 \times 1122.5 = 212.3$	
15	0	99	3.0	91.7	3.2	47.44	$0.28 \times 1122.5 = 212.3$	212.3 m^3
15	2 ± 0.2	7	1.2	6.5	5.0	5.76	$0.28 \times 1122.5 = 212.3$	
15	9 ± 0.9	2	1.0	1.9	9.5	5.69	$0.28 \times 1122.5 = 212.3$	
16	0	53	1.6	94.6	1.7	48.18	$0.28 \times 1122.5 = 212.3$	212.3 m^3
16	2 ± 0.2	3	1.0	5.4	2.2	3.76	$0.28 \times 1122.5 = 212.3$	
17	0	17	1.5	94.4	1.6	47.50	$0.28 \times 1122.5 = 212.3$	212.3 m^3
17	2 ± 0.2	1	1.0	5.6	1.7	3.14	$0.28 \times 1122.5 = 212.3$	
18	0	15	1.5	100.0	1.5	50.24	0	

OK, "X" " CREATED BY SCORING "TCSHR" BY "EXCEL11" TOTAL

EXCEL

HUAI TAKO WATERSHED
HYDROGRAPH
RAINFALL YEAR 1982
28 - 31 AUG 1982

	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	SUM	M ³ /DAY	CMS
0																		50.00		
6	50.00																	1400.00		
12	50.00	1350.00																1337.50		
18	62.50	1275.00	0.00															2087.50	4875.00	0.056
24	493.75	1593.75	0.00	0.00														5256.25		
30	50.00	5206.25	0.00	0.00	0.00													2131.25		
36	143.75	1275.00	262.50	0.00	0.00	450.00												3312.50		
42	112.50	2750.00	0.00	0.00	0.00	450.00	0.00											3725.00	14425.00	0.167
48	81.25	2868.75	37.50	0.00	175.00	562.50	0.00	0.00										3525.00		
54	268.75	1306.25	0.00	0.00	0.00	1950.00	0.00	0.00	0.00									6868.75		
60	268.75	6087.50	37.50	0.00	25.00	450.00	0.00	0.00	0.00	0.00								7475.00		
66	262.50	6237.50	37.50	0.00	0.00	937.50	0.00	0.00	0.00	0.00	0.00							6231.25	24100.00	0.279
72	293.75	4862.50	37.50	0.00	25.00	1012.50	0.00	0.00	0.00	0.00	0.00	0.00						7475.00		
78	425.00	6575.00	75.00	0.00	25.00	375.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00					10481.25		
84	106.25	7025.00	37.50	0.00	25.00	2062.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1225.00			5331.25		
90	200.00	1643.75	150.00	0.00	50.00	2062.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1225.00	0.00		6306.25	29593.75	0.343
96	37.50	2968.75	37.50	0.00	25.00	1650.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	56.25	1531.25	0.00	0.00	8375.00		
102	12.50	956.25	75.00	0.00	100.00	2287.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4943.75	0.00	0.00	3987.50		
108	0.00	318.75	0.00	0.00	25.00	2400.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.75	1225.00	0.00	0.00	3150.00		
114		0.00	0.00	0.00	50.00	600.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2500.00	0.00	0.00	3862.50	19375.00	0.224
120			0.00	0.00	0.00	1087.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.75	2756.25	0.00	0.00	1044.00		
126				0.00	0.00	337.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.75	687.75	0.00	0.00	5693.75		
132					0.00	112.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.75	5562.50	0.00	0.00	5600.00		
138						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	37.50	5562.50	0.00	0.00	4406.25	16744.00	0.194
144							0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.75	4387.50	0.00	0.00	3743.75		
150								0.00	0.00	0.00	0.00	0.00	0.00	75.00	3668.75	0.00	0.00	6343.75		
156									0.00	0.00	0.00	0.00	0.00	18.75	6325.00	0.00	0.00	1618.75		
162										0.00	0.00	0.00	0.00	37.50	1581.25	0.00	0.00	2856.25	14562.50	0.169
168											0.00	0.00	0.00	0.00	2856.25	0.00	0.00	918.75		
174												0.00	0.00	0.00	918.75	0.00	0.00	306.25		
180													0.00	0.00	306.25	0.00	0.00	0.00	1225.00	0.014
186															0.00	0.00	0.00	0.00		
192																0.00	0.00	0.00		
198																	0.00	0.00		

HUAI TAO WATERSHED
HYDROGRAPH
RAINFALL YEAR 1982
1 - 4 SEPT 1982

0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	SUM	M ³ /DAY	CMS
6	675.00																675.00		
12	675.00	7381.25															8056.25		
16	843.75	7225.00	525.00														8593.75		
24	2837.50	7625.00	525.00	0.00													10987.50	28312.50	0.328
30	675.00	21362.50	656.25	0.00	856.25												23550.00		
36	1393.75	11162.50	2231.25	0.00	850.00	14250.00											29887.50		
42	1518.75	13318.75	525.00	175.00	1062.50	14987.50	650.00										32237.50		
48	550.00	15225.00	1087.50	0.00	3543.75	14100.00	650.00	100.00									35256.25	120931.25	1.400
54	3081.25	14818.75	1181.25	25.00	850.00	34931.25	812.50	100.00	350.00								56150.00		
60	3081.25	34693.75	431.25	0.00	1762.50	26668.75	2787.50	125.00	350.00	6493.75							76393.75		
66	2450.00	31043.75	2400.00	25.00	1912.50	24375.00	650.00	681.25	437.50	6375.00	225.00						70575.00		
72	3418.75	31737.50	2400.00	25.00	712.50	29337.50	1350.00	100.00	1618.75	6375.00	225.00	0.00					77300.00	280418.75	3.246
78	3550.00	30775.00	1912.50	25.00	3900.00	37268.75	1462.50	243.75	350.00	17256.25	281.25	0.00	2362.50				99387.50		
84	887.50	30993.75	2662.50	50.00	3912.50	67900.00	537.50	225.00	743.75	10837.50	1150.00	0.00	2275.00	31875.00			154050.00		
90	1606.25	6431.25	2775.00	25.00	3100.00	57068.75	2975.00	118.75	787.50	11337.50	225.00	0.00	2562.50	37687.50	1343.75		128043.75		
96	506.25	9800.00	693.75	100.00	4312.50	66737.50	2975.00	493.75	306.25	13175.00	493.75	0.00	7712.50	34562.50	1275.00	225.00	143368.75	524850.00	6.075
102	168.75	2968.75	1256.25	25.00	4487.50	57168.75	2375.00	493.75	1618.25	15187.50	506.25	0.00	3062.50	76187.50	1593.75	225.00	167324.50		
108	0.00	962.50	393.75	50.00	1112.50	63600.00	3300.00	437.50	1618.75	30912.50	212.50	0.00	4475.00	67312.50	5200.00	281.25	179868.75		
114		750.00	131.25	0.00	2012.50	17787.50	3450.00	543.75	1312.50	26812.50	1056.25	0.00	4912.50	56125.00	1275.00	1281.25	117450.00		
120			0.00	0.00	637.50	18725.00	862.50	675.00	1793.75	29975.00	1056.25	0.00	3850.00	67312.50	2737.50	225.00	127850.00	592493.25	6.858
126				0.00	212.50	5893.75	1562.50	168.75	1925.00	26831.25	875.00	0.00	10937.50	87562.50	2868.75	512.50	139350.00		
132					0.00	1593.75	487.50	312.50	481.25	27918.75	1168.75	0.00	10337.50	149750.00	1281.25	506.25	193837.50		
138						2487.50	162.50	75.00	875.00	5775.00	1300.00	0.00	9625.00	124312.50	6062.50	231.25	150906.25		
144							0.00	25.00	262.50	7612.50	325.00	0.00	10587.50	145750.00	6200.00	1075.00	171837.50	655931.25	7.592
150								0.00	87.50	2081.25	593.75	0.00	11112.50	129812.50	4837.50	1075.00	149600.00		
156									0.00	637.50	168.75	0.00	2450.00	170187.50	6562.50	912.50	180918.75		
162										850.00	56.25	0.00	3975.00	75625.00	6987.50	1187.50	88681.25		
168											0.00	0.00	1225.00	64375.00	1643.75	1375.00	68618.75	487818.75	5.646
174												0.00	400.00	28562.50	2968.75	343.75	32275.00		
180													150.00	8312.50	956.25	631.25	10050.00		
186														8937.50	318.75	168.75	9425.00		
192															0.00	56.25	56.25	51806.25	0.600
198																0.00	0.00		

HUAI TAKO WATERSHED
HYDROGRAPH
RAINFALL YEAR 1982
5 - 8 SEPT 1982

0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	SUM	M ³ /DAY	CMS
6	16362.50																16362.50		
12	17543.75	96187.50															113731.25		
18	16431.25	127187.50	12456.25														156075.00		
24	39800.00	114375.00	12850.00	2225.00													169250.00	455418.75	5.27
30	31368.75	229875.00	12162.50	2150.00	0.00												275556.25		
36	28062.50	225187.50	30751.25	2406.25	0.00	906.25											287343.75		
42	33818.75	176750.00	22787.50	7231.25	0.00	900.00	0.00										241487.50		
48	43306.25	208000.00	21262.50	2937.50	175.00	1125.00	0.00	0.00									276806.25	1081193.75	12.51
54	101262.50	274250.00	25537.50	4312.50	0.00	3731.25	0.00	0.00	2137.50								411231.25		
60	65093.75	445750.00	32175.00	4631.25	25.00	900.00	87.50	0.00	2087.50	21306.25							572056.25		
66	76260.75	368187.50	59500.00	3706.25	0.00	1862.50	0.00	0.00	2281.25	23856.25	1456.25						537118.75		
72	65681.25	427937.50	50175.00	10325.00	25.00	2025.00	12.50	0.00	6762.50	22131.25	1412.50	400.00					586887.50	2107293.75	24.39
78	75262.50	397562.50	58575.00	9687.50	25.00	750.00	0.00	0.00	3006.25	51281.25	1625.00	400.00	0.00				598175.00		
84	23350.00	603125.00	49962.50	9087.50	25.00	4125.00	12.50	0.00	3975.00	42843.75	5062.50	500.00	0.00	975.00			743043.75		
90	22237.50	345937.50	54025.00	9843.75	50.00	4475.00	12.50	0.00	4456.25	36900.00	1806.25	1806.25	0.00	975.00	0.00		482525.00		
96	7956.25	280625.00	13467.50	10218.75	25.00	3275.00	12.50	0.00	3850.00	44431.25	2812.50	400.00	131.25	406.25	0.00	0.00	367631.25	2191375.00	25.36
102	2156.25	139375.00	14312.50	2231.25	100.00	4562.50	25.00	0.00	9993.75	57593.75	3075.00	843.75	0.00	1525.00	0.00	0.00	235793.75		
108	3112.50	42625.00	4337.50	3643.75	25.00	4737.50	12.50	0.00	9156.25	100693.75	2156.25	900.00	18.75	975.00	87.50	0.00	172481.25		
114		38750.00	1175.00	1125.00	50.00	1175.00	50.00	0.00	8887.50	83718.75	6731.25	343.75	0.00	693.75	0.00	0.00	142700.00		
120			2000.00	368.75	0.00	2125.00	12.50	0.00	9156.25	98068.75	6406.25	1843.75	18.75	731.25	12.50	0.00	120743.75	671718.75	7.77
126				150.00	0.00	675.00	25.00	0.00	9268.75	85243.75	5725.00	1843.75	18.75	287.50	0.00	0.00	103237.50		
132					0.00	225.00	0.00	0.00	1975.00	103531.25	6506.25	1487.50	18.75	1506.25	12.50	0.00	115262.50		
138						0.00	0.00	0.00	3212.50	38806.25	6643.75	2043.75	37.50	1506.25	12.50	0.00	52262.50		
144							0.00	0.00	987.50	34512.50	1468.75	2175.00	18.75	1225.00	12.50	0.00	40400.00	311162.50	3.60
150								0.00	325.00	14056.25	2556.25	543.75	75.00	1668.75	25.00	0.00	19250.00		
156									175.00	3931.25	806.25	987.50	18.75	1800.00	12.50	0.00	7731.25		
162										4875.00	268.75	300.00	37.50	450.00	50.00	0.00	5981.25		
168											75.00	100.00	0.00	818.75	12.50	0.00	1006.25	33968.75	0.39
174												0.00	0.00	243.75	25.00	0.00	268.75		
180													0.00	81.25	0.00	0.00	81.25		
186														0.00	0.00	0.00	0.00		
192															0.00	0.00	0.00	350.00	0.00
198																0.00	0.00		

HUAI TAKO WATERSHED
HYDROGRAPH
RAINFALL YEAR 1982
9 SEPT 1982

0	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	SUM	MM3/DAY	CM3
																	981.25		
6	981.25																23775.00		
12	950.00	22825.00															30250.00		
18	1187.50	28281.25	781.25														30562.50	85568.75	0.99
24	3943.75	25643.75	775.00	200.00													56193.75		
30	950.00	54075.00	968.75	200.00	0.00												56162.50		
36	2012.50	50637.50	3262.50	250.00	0.00	0.00											44606.25		
42	2137.50	40637.50	775.00	1056.25	0.00	0.00	0.00										51431.25	208393.75	2.41
48	887.50	48731.25	1612.50	200.00	0.00	0.00	0.00	0.00									71012.50		
54	4450.00	64375.00	1743.75	443.75	0.00	0.00	0.00	0.00	0.00								112131.25		
60	4512.50	106512.50	656.25	450.00	0.00	0.00	0.00	0.00	0.00	0.00							95231.25		
66	3550.00	87925.00	3562.50	193.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00						112781.25	391156.25	4.53
72	5012.50	103250.00	3575.00	943.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00					102487.50		
78	5137.50	93568.75	2837.50	943.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				137087.50		
84	1237.50	131125.00	3937.50	787.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			72731.25		
90	2237.50	65337.50	4112.50	1043.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		56318.75	368625.00	4.27
96	712.50	53412.50	1018.75	1175.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	27368.75		
102	237.50	24993.75	1843.75	293.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8368.75		
108	0.00	7250.00	581.25	537.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7687.50		
114		7343.75	193.75	150.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50.00	43475.00	0.50
120			0.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
126				0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
132					0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
138						0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
144							0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
150								0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
156									0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
162										0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		
168											0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
174												0.00	0.00	0.00	0.00	0.00	0.00		
180													0.00	0.00	0.00	0.00	0.00		
186														0.00	0.00	0.00	0.00		
192															0.00	0.00	0.00	0.00	0.00
198																0.00	0.00		
204																	0.00		

HUAI TAKO WATERSHED
HYDROGRAPH
RAINFALL 28 AUG - 9 SEPT 1962

			SUM	M ³ /DAY	CMS
1	6	50.00	50.00		
2	12	1400.00	1400.00		
3	18	1337.50	1337.50		
4	24	2087.50	2087.50	4875.00	0.056
5	30	5256.25	5256.25		
6	36	2131.25	2131.25		
7	42	3312.50	3312.50		
8	48	3725.00	3725.00	14425.00	0.167
9	54	3525.00	3525.00		
10	60	6868.75	6868.75		
11	66	7475.00	7475.00		
12	72	6231.25	6231.25	24100.00	0.279
13	78	7475.00	7475.00		
14	84	10481.25	10481.25		
15	90	5331.25	5331.25		
16	96	6306.25	6306.25	29593.75	0.343
17	102	8375.00	8375.00		
18	108	3937.50	3937.50		
19	114	3150.00	3150.00		
20	120	3862.50	3862.50	47687.50	0.552
21	126	1044.00	1044.00		
22	132	5653.75	5653.75		
23	138	5600.00	5600.00		
24	144	4406.25	4406.25	137675.25	1.593
25	150	3743.75	3743.75		
26	156	6343.75	6343.75		
27	162	1618.75	1618.75		
28	168	2856.25	2856.25	80156.25	2.948
29	174	918.75	918.75		
30	180	306.25	306.25		
31	186	0.00	0.00		
32	192	0.00	0.00	526075.00	6.089

HYDROGRAPH			
DATE	DAY	OBSERVED	COMPUTED
28	1	0.85	0.056
29	2	0.73	0.167
30	3	0.55	0.279
31	4	0.64	0.343
1	5	0.82	0.552
2	6	1.30	1.593
3	7	3.10	3.414
4	8	11.28	6.089
5	9	27.40	12.129
6	10	33.40	20.106
7	11	37.90	30.036
8	12	33.40	25.963
9	13	27.40	8.765
10	14	18.16	6.013
11	15	17.12	4.920
12	16	13.32	4.271
13	17	11.96	0.503
14	18	10.60	0.000

33 198	0.00	167324.50	16362.50	183687.00		
34 204	0.00	179868.75	113731.25	293600.00		
35 210		117450.00	156075.00	273525.00		
36 216		127850.00	169250.00	297100.00	1047912.00	12.129
37 222		139350.00	275556.25	414906.25		
38 228		193837.50	287343.75	481181.25		
39 234		150906.25	241487.50	392393.75		
40 240		171837.50	276806.25	448643.75	1737125.00	20.106
41 246		149600.00	411231.25	560831.25		
42 252		180918.75	572056.25	752975.00		
43 258		88681.25	537118.75	625800.00		
44 264		68618.75	586887.50	655506.25	2595112.50	30.036
45 270		32275.00	598175.00	630450.00		
46 276		10050.00	743043.75	753093.75		
47 282		9425.00	482525.00	491950.00		
48 288		56.25	367631.25	367687.50	2243181.25	25.963
49 294	0.00	235793.75	981.25	236775.00		
50 300		172481.25	23775.00	196256.25		
51 306		142700.00	30250.00	172950.00		
52 312		120743.75	30562.50	151306.25	757287.50	8.765
53 318		103237.50	56193.75	159431.25		
54 324		115262.50	56162.50	171425.00		
55 330		52262.50	44606.25	96868.75		
56 336		40400.00	51431.25	91831.25	519556.25	6.013
57 342		19250.00	71012.50	90262.50		
58 348		7731.25	112131.25	119862.50		
59 354		5981.25	95231.25	101212.50		
60 360		1006.25	112781.25	113787.50	425125.00	4.920
61 366		268.75	102487.50	102756.25		
62 372		81.25	137087.50	137168.75		
63 378		0.00	72731.25	72731.25		
64 384		0.00	56318.75	56318.75	368975.00	4.271
65 390		0.00	27368.75	27368.75		
66 396			8368.75	8368.75		
67 402			7687.50	7687.50		
68 408			50.00	50.00	43475.00	0.503
69 414			0.00	0.00		
70 420			0.00	0.00		
71 426			0.00	0.00		
72 432			0.00	0.00	0.00	0.000

