```
C*************************************************************************
C
C  POSTHOLE
C
C  PURPOSE:
C  FINDS RECTANGLES FORMED OF FOUR POSTHOLES FROM THE LIST
C
C  OPERATION:
C  TAKES INPUT AND OUTPUT FILE NAMES AND ASKS FOR THREE PARAMETERS:
C  UPPER AND LOWER BOUNDS FOR THE RECTANGLE SIDES, AND THE TOLERANCE
C  OF POSTHOLE POSITION
C  GRAPHIC OUTPUT ASSUMES VGA COLOR DISPLAY
C
C  FILES:
C  LUN 3 INPUT, LIST OF POSTHOLES
C  LUN 2 OUTPUT, LIST OF RECTANGLES
C  LUN 7 SCRATCH
C
C  SUBROUTINES:
C  ALL SUBROUTINES USED ARE INCLUDED IN THE SOURCE CODE
C
C  GENERATION:
C  fl POSTHOLE.FOR /link GRAPHICS
C
C*************************************************************************
 INCLUDE'FGRAPH.FI'
 COMMON NR,SMIN,SMAX,TOL
 PARAMETER(NHO=1700,NSO=33,PI=3.14159265)
C --- IF FATAL ERROR F1027 OCCURS IN COMPILATION, COMPILE
C --- WITH /Gt OPTION
 CHARACTER*25 FIL
 DIMENSION WX(NHO),WY(NHO),IX(NHO),IY(NHO),IW1(NHO),IWI(NHO)
 CHARACTER*6 NAME(NHO),CW(NHO)
 DIMENSION IP(0:NSO,NSO),IP1(0:NSO)
 INTEGER*2 IHR,IMIN,ISEC,I100TH,JHR,JMIN,JSEC,J100TH
 EQUIVALENCE(IW1(1),CW(1),WX(1))
 EQUIVALENCE(IWI(1),WY(1))
C
C --- INITIALIZE FILES
C
 WRITE(*,'(A\)')' Enter the posthole file name: '
 READ(*,1010)FIL
1010 FORMAT(A25)
 OPEN(UNIT=3,FILE=FIL,MODE='READ',STATUS='OLD',ERR=910)
 WRITE(*,'(A\)')' Enter  the  result file name: '
 READ(*,1010)FIL
 OPEN(UNIT=2,FILE=FIL,STATUS='UNKNOWN',ERR=920)
C
C --- READ POSTHOLE DATA
C
 NH=0
 DO I=1,NHO
 READ(3,1020,END=10)IX(I),IY(I),NAME(I)
1020 FORMAT(8X,2I8,A6)
 NH=NH+1
 ENDDO
 READ(3,1020,END=10)
 WRITE(*,*)'Only',NHO,' postholes can be processed'
 WRITE(*,*)'Increase NHO in the source code'
 GO TO 990
10 WRITE(*,1030)NH
1030 FORMAT(I6,' postholes read')
C
C ---  FIND THE SCALE FOR COORDINATES
C
```

```
C --- HUGE(I) IS THE MAXIMUM INTEGER
 IHUGE=HUGE(I)-1
 IXMI=IHUGE
 IYMI=IHUGE
 IXMA=-IHUGE
 IYMA=-IHUGE
 DO I=1,NH
 IXMI=MIN0(IXMI,IX(I))
 IYMI=MIN0(IYMI,IY(I))
 IXMA=MAX0(IXMA,IX(I))
 IYMA=MAX0(IYMA,IY(I))
 ENDDO
 XSIZE=IXMA-IXMI
 YSIZE=IYMA-IYMI
 SC=AMIN1(639./XSIZE,479./YSIZE)
C
C --- DISPLAY THE POSTHOLES
C
 WRITE(*,*)'Press ENTER to continue'
 READ(*,*)
 CALL DSPHO(NH,IX,IY,IXMI,IYMI,SC)
C
C --- ENTER LENGTH UNIT !!!!!!!!!!
C
C
C --- READ SEARCH PARAMETERS
C
 WRITE(*,*)'Enter the admissible rectangle size and tolerance'
 WRITE(*,*)'(in the same units as posthole coordinates)'
20 WRITE(*,'(A\)')'       the smallest rectangle side: '
 READ(*,*)SMIN
 IF(SMIN.LE..0)GO TO 20
30 WRITE(*,'(A\)')'       the  largest rectangle side: '
 READ(*,*)SMAX
 IF(SMAX.LE..0)GO TO 30
 IF(SMAX.LE.SMIN)GO TO 20
40 WRITE(*,'(A\)')'       tolerance of posthole position: '
 READ(*,*)TOL
 IF(TOL.LE..0)GO TO 40
C --- WRITE THE FIRST RECORD ON UNIT 2
 WRITE(2,1040)SMIN,SMAX,TOL
1040 FORMAT('MIN.DISTANCE',F7.2,' MAX.DISTANCE',F7.2,' TOLERANCE',
     1 F7.2)
C
C --- CALCULATE A GUESS BY FLETCHER AND LOCK
C
 AR=XSIZE*YSIZE
 RM=(SMAX**2-SMIN**2)*PI*NH*(NH-1)/AR
 RMU=(NH-2)*2*(SMAX-SMIN)*TOL/AR
 RLA=-2.41*TOL**2*(NH-2)/AR
 SI=RMU*(1.-EXP(RLA))
 NRE=.5*RM*EXP(-SI)*SI/(1.-SI)
 WRITE(*,1050)NRE
1050 FORMAT(' Expected number of rectangles is',I4)
C
C --- DIVIDE THE WHOLE AREA INTO LY BY LX (APPROXIMATE) SQUARES
C --- OF SIZE SMAX BY SMAX EACH
C
 CALL GETTIM(IHR,IMIN,ISEC,I100TH)
 LX=MAX0(1,IFIX(XSIZE/SMAX))
 LY=MAX0(1,IFIX(YSIZE/SMAX))
 IF(LX.GT.NSO)GO TO 50
 IF(LY.GT.NSO)GO TO 50
 XSQ=XSIZE/LX
 YSQ=YSIZE/LY
```

```
 GO TO 60
50 WRITE(*,*)'Too large area to be processed'
 WRITE(*,*)'Increase NSO in the source code at least to',
     1 MAX0(LX,LY)
 GO TO 990
C
C --- SORT THE ARRAY IX, AND ORDER IY AND NAME, TOO
C
60 CALL SORT3(NH,IX,IY,NAME,IW1,CW,IWI)
C
C --- CONSTRUCT THE COLUMN POINTER ARRAY IP1
C --- ENTRY IP1(JX-1) WILL POINT TO THE PLACE WHERE THE HOLES
C --- OF JXTH COLUMN OF SQUARES BEGIN IN THE ARRAYS IX, IY, AND NAME
C
 IP1(0)=0
 I=1
 K=1
70 IF(IX(K).LE.IXMI+INT(I*XSQ))THEN
     K=K+1
     IF(K.GT.NH)THEN
  DO III=I,LX
  IP1(III)=NH
  ENDDO
     ELSE
  GO TO 70
     ENDIF
 ELSE
     IP1(I)=K-1
     I=I+1
     IF(I.LE.LX)GO TO 70
 ENDIF
C
C --- CONSTRUCT A LY BY LX POINTER ARRAY IP
C --- ENTRY IP(JY-1,JX-1) WILL POINT TO THE PLACE WHERE THE HOLES
C --- OF SQUARE (JY,JX) BEGIN IN IX, IY, AND NAME
C
C --- TRIVIAL INITIAL VALUES
 DO I=1,LX
 IP(0,I)=IP1(I-1)
 ENDDO
C --- LOOP FOR THE ITH COLUMN OF IP
 DO I=1,LX
 IB=IP1(I-1)+1
 IE=IP1(I)
 NHI=IE-IB+1
 IF(NHI.EQ.0)THEN
     DO J=1,LY
     IP(J,I)=IE
     ENDDO
 ELSE
     IF(NHI.GT.1)
     1  CALL SORT3(NHI,IY(IB),IX(IB),NAME(IB),IW1,CW,IWI)
     J=1
     K=IB
80     IF(IY(K).LE.IYMI+INT(J*YSQ))THEN
  K=K+1
  IF(K.GT.IE)THEN
      DO JJJ=J,LY
      IP(JJJ,I)=IE
      ENDDO
  ELSE
      GO TO 80
  ENDIF
     ELSE
  IP(J,I)=K-1
```

```
  J=J+1
  IF(J.LE.LY)GO TO 80
     ENDIF
 ENDIF
 ENDDO
C
C --- RECTANGLES ARE LOOKED FOR IN THE LOOP OVER ALL
C --- THE LY BY LX SQUARES AND THE FOUND ONES ARE
C --- WRITTEN TO DIRECT ACCESS SCRATCH UNIT 7
C
 OPEN(UNIT=7,ACCESS='DIRECT',RECL=16)
 NR=0
 WRITE(*,*)
 JUP=MAX0(1,LY-1)
 IUP=MAX0(1,LX-1)
 DO J=1,JUP
 DO I=1,IUP
 WRITE(*,1060)J,I,JUP,IUP
1060 FORMAT('+Processing part',I3,',',I3,' of',I3,',',I3)
 IJB1=IP(J-1,I)+1
 IF(LY.GT.1)THEN
     NHIJ1=IP(J+1,I)-IJB1+1
 ELSE
     NHIJ1=IP(J,I)-IJB1+1
 ENDIF
 NHIJ=NHIJ1
 K=IJB1
 DO L=1,NHIJ1
 WX(L)=IX(K)
 WY(L)=IY(K)
 K=K+1
 ENDDO
 IF(LX.GT.1)THEN
     IJB2=IP(J-1,I+1)+1
     IF(LY.GT.1)THEN
  NHIJ2=IP(J+1,I+1)-IJB2+1
     ELSE
  NHIJ2=IP(J,I+1)-IJB2+1
     ENDIF
     NHIJ=NHIJ1+NHIJ2
     K=IJB2
     DO L=NHIJ1+1,NHIJ
     WX(L)=IX(K)
     WY(L)=IY(K)
     K=K+1
     ENDDO
 ELSE
     IJB2=1
 ENDIF
 IF(NHIJ.GE.4)CALL RECTAN(NHIJ,NHIJ1,IJB1-1,IJB2-1,WX,WY)
 ENDDO
 ENDDO
C
C --- WRITE THE RESULTING NR RECTANGLES ON UNIT 2
C --- (RECTANGLE OUTPUT UNIT) TOGETHER WITH THEIR NAMES
C
 DO I=1,NR
 READ(7,REC=I)IH1,IH2,IH3,IH4
 WRITE(2,1070)IX(IH1),IX(IH2),IX(IH3),IX(IH4),
     1 IY(IH1),IY(IH2),IY(IH3),IY(IH4),
     1 NAME(IH1),NAME(IH2),NAME(IH3),NAME(IH4)
1070 FORMAT(8I8,4A6)
 ENDDO
 WRITE(*,1080)NR
1080 FORMAT(I4,' rectangles found')
```

```
 CALL GETTIM(JHR,JMIN,JSEC,J100TH)
 SEC=(JHR-IHR)*3600.+(JMIN-IMIN)*60.+JSEC-ISEC+
     1 (J100TH-I100TH)*.01
 WRITE(*,1090)SEC
1090 FORMAT(' Time elapsed',F8.2,' seconds')
C
C --- DRAW THE RECTANGLES
C
 WRITE(*,*)'Press ENTER to continue'
 READ(*,*)
 CALL DSPREC(NR,NH,IX,IY,IXMI,IYMI,SC)
 GO TO 990
C
C --- ERROR PROCESSING
C
910 WRITE(*,*)'Error opening input file'
 GO TO 990
920 WRITE(*,*)'Error opening output file'
990 STOP
 END
C*********************************************************************
C
C     SUBROUTINE DSPHO DISPLAYS THE HOLES
C     PARAMETERS
C NH IS THE NUMBER OF HOLES.
C IX, IY ARE INTEGER ARRAYS OF HOLE COORDINATES.
C IXMI,IYMI ARE MINIMUM COORDINATES AND SC IS THE SCALE
C
C*********************************************************************
 SUBROUTINE DSPHO(NH,IX,IY,IXMI,IYMI,SC)
 DIMENSION IX(1),IY(1)
 INTEGER*2 IIX,IIY
 INCLUDE'FGRAPH.FD'
 INTEGER ico(0:15)/$BLACK,$BLUE,$GREEN,$CYAN,$RED,
     1  $MAGENTA,
     1  $BROWN, $WHITE,$GRAY, $LIGHTBLUE,$LIGHTGREEN,$LIGHTCYAN,
     1  $LIGHTRED,$LIGHTMAGENTA, $LIGHTYELLOW,$BRIGHTWHITE/
 INTEGER*2 I2,J2
 I2=setvideomode($VRES16COLOR)
 IF(I2.EQ.0)GO TO 990
 I4=setbkcolor(ico(8))
 J2=14
 I2=setcolor(J2)
 DO I=1,NH
 IIX=(IX(I)-IXMI)*SC
 IIY=(IY(I)-IYMI)*SC
C I2=setpixel(IIX,IIY)
 I2=ellipse($GFILLINTERIOR,IIX-2,IIY-2,
     1 IIX+2,IIY+2)
 ENDDO
 READ(*,*)
 I2=setvideomode($DEFAULTMODE)
 RETURN
990 WRITE(*,*)'This is not VGA. Change setvideomode argument'
 STOP
 END
C*********************************************************************
C
C     SUBROUTINE DSPREC DISPLAYS THE HOLES AND THE RECTANGLES FOUND
C     PARAMETERS
C NR IS THE NUMBER OF RECTANGLES.
C NH IS THE NUMBER OF HOLES.
C IX, IY ARE INTEGER ARRAYS OF HOLE COORDINATES.
C IXMI,IYMI ARE MINIMUM COORDINATES AND SC IS THE SCALE
C
```

```
C**********************************************************************
 SUBROUTINE DSPREC(NR,NH,IX,IY,IXMI,IYMI,SC)
 DIMENSION IX(1),IY(1)
 INTEGER*2 IIX,IIY,IIX1,IIY1,ITW
 PARAMETER(ITW=2)
 INCLUDE'FGRAPH.FD'
 RECORD/xycoord/S
 INTEGER ico(0:15)/$BLACK,$BLUE,$GREEN,$CYAN,$RED,
     1  $MAGENTA,
     1  $BROWN, $WHITE,$GRAY, $LIGHTBLUE,$LIGHTGREEN,$LIGHTCYAN,
     1  $LIGHTRED,$LIGHTMAGENTA, $LIGHTYELLOW,$BRIGHTWHITE/
 INTEGER*2 I2,J2
 I2=setvideomode($VRES16COLOR)
 I4=setbkcolor(ico(8))
C --- HOLES ARE DISPLAYED FIRST
 J2=14
 I2=setcolor(J2)
 DO I=1,NH
 IIX=(IX(I)-IXMI)*SC
 IIY=(IY(I)-IYMI)*SC
C I2=setpixel(IIX,IIY)
 I2=ellipse($GFILLINTERIOR,IIX-ITW,IIY-ITW,IIX+ITW,IIY+ITW)
 ENDDO
C --- RECTANGLES ARE DISPLAYED NOW
 J2=15
 I2=setcolor(J2)
 REWIND 2
 READ(2,1010)
1010 FORMAT(1X)
 DO I=1,NR
 READ(2,1020)IX1,IX2,IX3,IX4,IY1,IY2,IY3,IY4
1020 FORMAT(8I8)
 IIX1=(IX1-IXMI)*SC
 IIY1=(IY1-IYMI)*SC
 CALL moveto(IIX1,IIY1,S)
 IIX=(IX2-IXMI)*SC
 IIY=(IY2-IYMI)*SC
 I2=lineto(IIX,IIY)
 IIX=(IX4-IXMI)*SC
 IIY=(IY4-IYMI)*SC
 I2=lineto(IIX,IIY)
 IIX=(IX3-IXMI)*SC
 IIY=(IY3-IYMI)*SC
 I2=lineto(IIX,IIY)
 I2=lineto(IIX1,IIY1)
 ENDDO
 READ(*,*)
 I2=setvideomode($DEFAULTMODE)
 RETURN
 END
C**********************************************************************
C
C   SUBROUTINE INDEXX INDEXES AN ARRAY IARRIN OF LENGTH N,
C I.E. OUTPUTS THE ARRAY INDX SUCH THAT IARRIN(INDX(J))
C IS IN ASCENDING ORDER FOR J=1,...,N.
C   PARAMETERS
C THE INPUT QUANTITIES N AND IARRIN ARE NOT CHANGED.
C INDX IS ASSIGNED THE VALUES OF THE CORRESPONDING INDICES.
C SEE NUMERICAL RECIPES
C
C**********************************************************************
 SUBROUTINE INDEXX(N,IARRIN,INDX)
 DIMENSION IARRIN(1),INDX(1)
C --- INITIALIZATION OF THE INDEX ARRAY
 DO J=1,N
```

```
      INDX(J)=J
 ENDDO
C --- HEAPSORT WITH INDIRECT INDEXING THROUGH INDX
C --- IN ALL REFERENCES TO IARRIN
 L=N/2+1
 IR=N
10 CONTINUE
      IF(L.GT.1)THEN
  L=L-1
  INDXT=INDX(L)
  IQ=IARRIN(INDXT)
     ELSE
  INDXT=INDX(IR)
  IQ=IARRIN(INDXT)
  INDX(IR)=INDX(1)
  IR=IR-1
  IF(IR.EQ.1)THEN
       INDX(1)=INDXT
       RETURN
  ENDIF
     ENDIF
     I=L
     J=L+L
20      IF(J.LE.IR)THEN
  IF(J.LT.IR)THEN
       IF(IARRIN(INDX(J)).LT.IARRIN(INDX(J+1)))J=J+1
  ENDIF
  IF(IQ.LT.IARRIN(INDX(J)))THEN
       INDX(I)=INDX(J)
       I=J
       J=J+J
  ELSE
       J=IR+1
  ENDIF
     GO TO 20
     ENDIF
     INDX(I)=INDXT
 GO TO 10
 END
C*********************************************************************
C
C   SUBROUTINE SORT3 SORTS AN ARRAY IA OF LENGTH N INTO ASCENDING
C NUMERICAL ORDER WHILE MAKING THE CORRESPONDING REARRANGEMENTS
C OF THE ARRAYS IB AND C.
C AN INDEX TABLE IS CONSTRUCTED VIA THE SUBROUTINE INDEXX.
C   PARAMETERS
C N IS THE LENGTH OF ALL THE ARRAYS, IA AND IB ARE INPUT ARRAYS,
C KSP AND IWKSP ARE WORKING ARRAYS.
C INPUT ARRAY C IS CHARACTER*6, WORKSPACE CWKSP AS WELL.
C SEE NUMERICAL RECIPES
C
C*********************************************************************
 SUBROUTINE SORT3(N,IA,IB,C,KSP,CWKSP,IWKSP)
 DIMENSION IA(1),IB(1),KSP(1),IWKSP(1)
 CHARACTER*6 C(1),CWKSP(1)
C --- MAKE THE INDEX TABLE
 CALL INDEXX(N,IA,IWKSP)
C --- SAVE THE ARRAY IA
 DO J=1,N
 KSP(J)=IA(J)
 ENDDO
C --- COPY IT BACK IN THE REARRANGED ORDER
 DO J=1,N
 IA(J)=KSP(IWKSP(J))
 ENDDO
```

```
C --- SAVE THE ARRAY IB
 DO J=1,N
 KSP(J)=IB(J)
 ENDDO
C --- COPY IT BACK IN THE REARRANGED ORDER
 DO J=1,N
 IB(J)=KSP(IWKSP(J))
 ENDDO
C --- SAVE THE ARRAY C
 DO J=1,N
 CWKSP(J)=C(J)
 ENDDO
C --- COPY IT BACK IN THE REARRANGED ORDER
 DO J=1,N
 C(J)=CWKSP(IWKSP(J))
 ENDDO
 RETURN
 END
C***********************************************************************
C
C   SUBROUTINE RECTAN FINDS RECTANGLES FORMED OF FOUR POSTHOLES
C FROM THOSE PRESORTED AND SUPPLIED AS PARAMETERS (I.E. THOSE FROM
C FOUR BASIC SQUARES)
C THIS VERSION FINDS THE OTHER TWO VERTICES TO AN EDGE
C USING RIGHT ANGLES
C   PARAMETERS
C NH IS THE NUMBER OF POSTHOLES, NH1 IS THEIR NUMBER IN THE
C LEFT-HAND PAIR OF SQUARES.
C NB1 AND NB2 ARE THE POSITIONS OF THE POSTHOLES OF THE LEFT-HAND
C AND RIGHT-HAND PAIRS OF SQUARES IN THE GLOBAL POSTHOLE ARRAY,
C RESPECTIVELY, AND X, Y ARE ARRAYS OF COORDINATES OF POSTHOLES
C JUST BEING PROCESSED.
C
C***********************************************************************
 SUBROUTINE RECTAN(NH,NH1,NB1,NB2,X,Y)
C
C --- NR IS THE CURRENT NUMBER OF RECTANGLES FOUND.
C --- SMIN, SMAX, AND TOL ARE THE MINIMUM AND MAXIMUM LENGTHS OF
C --- RECTANGLE EDGES, AND THE TOLERANCE
C
 COMMON NR,SMIN,SMAX,TOL
 PARAMETER(PIH=1.5707963)
 DIMENSION X(1),Y(1)
 LOGICAL SECOND
C
C --- STATEMENT FUNCTION DEFINITION
C
 DIST(I,J)=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
C
C --- MAKE EDGES OF ADMISSIBLE PAIRS OF POSTHOLES
C --- AND LOOK FOR THE OTHER EDGES
C
 DO IH1=1,NH-1
 DO IH2=IH1+1,NH
C --- EDGE INADMISSIBLE
 DH12=DIST(IH1,IH2)
 IF(SMIN.GT.DH12.OR.DH12.GT.SMAX)GO TO 90
C --- THE ANGLE OF THE EDGE (RAY) IH1,IH2 WITH THE POSITIVE X-AXIS
 PHI=ATAN2(Y(IH2)-Y(IH1),X(IH2)-X(IH1))
C --- ALPHA IS THE ANGLE OF THE SOUGHT LATERAL EDGE.
C --- TWO CASES ARE TO BE CONSIDERED
 ALPHA=PHI-PIH
 SECOND=.TRUE.
 GO TO 20
10 ALPHA=PHI+PIH
```

```
 SECOND=.FALSE.
20 SI3I=X(IH1)+SMIN*COS(ALPHA)
 SI3A=X(IH1)+SMAX*COS(ALPHA)
 SIMIN=AMIN1(SI3I,SI3A)-TOL
 SIMAX=AMAX1(SI3I,SI3A)+TOL
C
C --- LOOK FOR THE ENDPOINT IH3 OF THE LATERAL EDGE
C
 DO IH3=1,NH
 IF(IH3.EQ.IH1.OR.IH3.EQ.IH2)GO TO 80
C --- THE ENDPOINT X-COORDINATE IS NOT IN THE RANGE
 IF(SIMIN.GT.X(IH3).OR.X(IH3).GT.SIMAX)GO TO 80
C --- THE ENDPOINT Y-COORDINATE DOES NOT AGREE
 TA3=Y(IH1)+(X(IH3)-X(IH1))*TAN(ALPHA)
 IF(ABS(TA3-Y(IH3)).GT.TOL)GO TO 80
C --- LATERAL EDGE INADMISSIBLE
 DH13=DIST(IH1,IH3)
 IF(SMIN.GT.DH13.OR.DH13.GT.SMAX)GO TO 80
C
C --- NOW THREE VERTICES ARE FOUND. CALCULATE THE POSITION OF THE
C --- FOURTH ONE AND LOOK FOR IT
C
 SI4=X(IH3)+X(IH2)-X(IH1)
 DO IH4=1,NH
C --- THE FOURTH VERTEX COINCIDES WITH ANY OF THE THREE
 IF(IH4.EQ.IH1.OR.IH4.EQ.IH2.OR.IH4.EQ.IH3)GO TO 70
C --- THE FOURTH VERTEX X-COORDINATE DOES NOT AGREE
 IF(ABS(SI4-X(IH4)).GT.TOL)GO TO 70
C --- THE FOURTH VERTEX Y-COORDINATE DOES NOT AGREE
 TA4=Y(IH3)+Y(IH2)-Y(IH1)
 IF(ABS(TA4-Y(IH4)).GT.TOL)GO TO 70
C
C --- RECTANGLE FOUND. TEST IT
C
C --- DIFFERENT LENGTHS OF EDGES
 DH34=DIST(IH3,IH4)
 IF(ABS(DH12-DH34).GT.TOL)GO TO 70
C --- EDGE INADMISSIBLE
 IF(SMIN.GT.DH34.OR.DH34.GT.SMAX)GO TO 70
C --- DIFFERENT LENGTHS OF LATERAL EDGES
 DH24=DIST(IH2,IH4)
 IF(ABS(DH13-DH24).GT.TOL)GO TO 70
C --- DIFFERENT LENGTHS OF DIAGONALS
 DH14=DIST(IH1,IH4)
 DH23=DIST(IH2,IH3)
 IF(ABS(DH14-DH23).GT.TOL)GO TO 70
C
C --- FIND GLOBAL VERTEX NUMBERS OF THE RECTANGLE
C
 IIH1=IH1+NB1
 IF(IH1.GT.NH1)IIH1=IH1+NB2-NH1
 IIH2=IH2+NB1
 IF(IH2.GT.NH1)IIH2=IH2+NB2-NH1
 IIH3=IH3+NB1
 IF(IH3.GT.NH1)IIH3=IH3+NB2-NH1
 IIH4=IH4+NB1
 IF(IH4.GT.NH1)IIH4=IH4+NB2-NH1
C
C --- CHECK FOR DUPLICATES AMONG THE RECTANGLES RECORDED
C
 DO IR=1,NR
 READ(7,REC=IR)IG1,IG2,IG3,IG4
C --- LOOK FOR IIH1 AMONG THE FOUR IG'S
 IF(IIH1.NE.IG1)GO TO 30
C --- IIH1=IG1, TRY TO IDENTIFY THE TWO RECTANGLES
```

```
 IF(IIH2.EQ.IG2.AND.IIH4.EQ.IG4.AND.IIH3.EQ.IG3)GO TO 70
 IF(IIH2.EQ.IG3.AND.IIH4.EQ.IG4.AND.IIH3.EQ.IG2)GO TO 70
 GO TO 60
30 IF(IIH1.NE.IG2)GO TO 40
C --- IIH1=IG2, TRY TO IDENTIFY THE TWO RECTANGLES
 IF(IIH2.EQ.IG4.AND.IIH4.EQ.IG3.AND.IIH3.EQ.IG1)GO TO 70
 IF(IIH2.EQ.IG1.AND.IIH4.EQ.IG3.AND.IIH3.EQ.IG4)GO TO 70
 GO TO 60
40 IF(IIH1.NE.IG3)GO TO 50
C --- IIH1=IG3, TRY TO IDENTIFY THE TWO RECTANGLES
 IF(IIH2.EQ.IG1.AND.IIH4.EQ.IG2.AND.IIH3.EQ.IG4)GO TO 70
 IF(IIH2.EQ.IG4.AND.IIH4.EQ.IG2.AND.IIH3.EQ.IG1)GO TO 70
 GO TO 60
50 IF(IIH1.NE.IG4)GO TO 60
C --- IIH1=IG4, TRY TO IDENTIFY THE TWO RECTANGLES
 IF(IIH2.EQ.IG3.AND.IIH4.EQ.IG1.AND.IIH3.EQ.IG2)GO TO 70
 IF(IIH2.EQ.IG2.AND.IIH4.EQ.IG1.AND.IIH3.EQ.IG3)GO TO 70
60 ENDDO
C
C --- THE RECTANGLE DOES NOT COINCIDE WITH OTHERS. RECORD IT
C
 NR=NR+1
C WRITE(*,*)NR,IIH1,IIH2,IIH3,IIH4
 WRITE(7,REC=NR)IIH1,IIH2,IIH3,IIH4
70 ENDDO
C --- THE FOURTH VERTEX NOT FOUND
80 ENDDO
C --- THE THIRD VERTEX NOT FOUND
 IF(SECOND)GO TO 10
90 ENDDO
 ENDDO
 RETURN
 END
```