

```

C-----
      PROGRAM INTERP
C-----
C Program to interpolate a straight line between successive data
C points in a file.
C-----
      PARAMETER (IDIM=10000)
      DIMENSION X(IDIM), Y(IDIM)
      DIMENSION XO(IDIM), YO(IDIM)
      DIMENSION CARD(10)
      LOGICAL LOG, FILE
      CHARACTER*60 IDENT, JDENT, KDENT
      CHARACTER*60 FILIN, FILOUT
      CHARACTER KR, BEL

C
      BEL = CHAR (7)
      WRITE (*,100)
10     WRITE (*,105)
      READ (*,125) FILIN
      IF (FILIN .EQ. ' ') STOP 'No input file name'

C
      CALL ASCIN (X,Y,IDIM,N,CARD,IDENT,FILIN,FILE)
      IF (.NOT. FILE) THEN
          WRITE (*,155) BEL
          GO TO 10
      END IF
      CALL FNAME (FILIN,JDENT)
      L = LEN (JDENT)
      KDENT = '<' // JDENT(1:L) // '> interpolated'
      L = L + 15

C
C Input data is now in (X,Y); Query for interpolation filespec.
C
12     WRITE (*,150)
      READ (*,125) FILIN
      IF (FILIN .EQ. ' ') GO TO 13

C
      CALL ASCIN (XO,YO,IDIM,NO,CARD,JDENT,FILIN,FILE)
      IF (.NOT. FILE) THEN
          WRITE (*,155) BEL
          GO TO 12
      END IF
      CALL FNAME (FILIN,JDENT)
      K = LEN (JDENT)
      KDENT = KDENT(1:L) // ' on <' // JDENT(1:K) // '>'
      GO TO 26

C
13     WRITE (*,140)
      READ (*,*) XSP,XEP
      WRITE (*,130)
      READ (*,125) KR
      LOG = (KR .EQ. 'Y') .OR. (KR .EQ. 'y')
      IF (LOG) GO TO 16

C
C Input desired interval DELX for interpolation;
C
      WRITE (*,115)
      READ (*,*) DELX

```

```

        WRITE (JDENT,110) DELX
        KDENT = KDENT(1:L) // ' linearly -- interval ' // JDENT
        GO TO 17
C
C Input desired ratio between successive X-values.
C
16     WRITE (*,135)
        READ (*,*) XRAT
        WRITE (JDENT,110) XRAT
        KDENT = KDENT(1:L) // ' logarithmically -- ratio ' // JDENT
        DELX = ALOG (XRAT)
C
C Now interpolate until end of input X-array is reached; input
C array is assumed to be in ascending order. Output is stored
C in (XO,YO).
C
17     X1 = AMAX1 (X(1),XSP)
        IF (LOG) X1 = ALOG (X1)
        X2 = AMIN1 (X(N),XEP)
        IF (LOG) X2 = ALOG (X2)
        NO = (X2 - X1)/DELX + 1.0
        NO = MIN0 (NO,IDIM)
        DO 22 I = 1,NO
            XN = X1 + (I-1)*DELX
            IF (LOG) XN = EXP (XN)
22     XO(I) = XN
26     CONTINUE
C Interpolation loop begins here.
        XF = X(1)
        YF = Y(1)
        XE = X(N)
        YE = Y(N)
        X1 = XF
        Y1 = YF
        ICT = 2
        DO 40 I = 1,NO
            XT = XO(I)
            IF ((XT .LE. XF) .OR. (XT .GE. XE)) GO TO 38
C XT in bounds of array X. Interpolate.
15     X2 = X(ICT)
            Y2 = Y(ICT)
            IF (XT .LT. X2) GO TO 20
C Look for next point in input.
            ICT = ICT + 1
            X1 = X2
            Y1 = Y2
            GO TO 15
C X1 .LE. XT .LT. X2: ==> interpolate.
20     YT = FNTERP (XT,X1,X2,Y1,Y2)
            YO(I) = YT
            GO TO 40
38     YO(I) = YF
            IF (XT .LE. XF) GO TO 40
            YO(I) = YE
40     CONTINUE
C Get output filespec from TT: and write data.
        WRITE (*,120) NO
        READ (*,125) FILOUT

```

```

        IF (FILOUT .NE. ' ') CALL ASCOUT (XO,YO,NO,IDENT,
*       KIDENT,FILOUT)
        GO TO 10
100     FORMAT (' INTERP -- Version 1.1a -- Copyright (c) 1993 ',
*       'by C.D. Porter.'/' Program to perform linear ',
*       'interpolation on a data set:'/' Enter blank line to ',
*       'terminate program.')
105     FORMAT (' Input filespec? '$)
110     FORMAT (G14.6)
115     FORMAT (' Input desired X-interval: '$)
120     FORMAT (' ',I5,' points;'/' Output filespec? '$)
125     FORMAT (A)
130     FORMAT (' Do you wish to interpolate on a logarithmic scale?',
1 ' (N) '$)
135     FORMAT (' Input desired ratio between successive ',
1 'X-values: '$)
140     FORMAT (' Input starting and ending X-values: '$)
150     FORMAT (' Enter filespec containing X-values at which you'/
1 ' wish to interpolate; if no filespec press RETURN '$)
155     FORMAT (' ',A1,'?-INTERP-W-File not found -- Try again;')
        END
        FUNCTION FNTERP (X,X1,X2,Y1,Y2)
C-----
C  Function to interpolate a straight line between (X1,Y1) and (X2,Y2).
C-----
        A = (Y2 - Y1)/(X2 - X1)
        B = Y1 - A*X1
        FNTERP = A*X + B
        RETURN
        END
        SUBROUTINE FNAME (FILES,FILEN)
C-----
C  Subroutine to find and separate the file name proper from any
C  information about the path which may be present.
C-----
        CHARACTER*(*) FILES,FILEN
        CHARACTER CH
C
        L = LEN (FILES)
        K = L
        DO 5 I = 1,L
            CH = FILES (K:K)
            IF ((CH .EQ. '\') .OR. (CH .EQ. ':')) GO TO 10
            K = K - 1
5           CONTINUE
10          CONTINUE
        FILEN = FILES (K+1:L)
        RETURN
        END
        SUBROUTINE ASCIN (X,Y,NDIM,NP,CARD,IDENT,FILIN,FILE)
C-----
C  Subroutine to read a data file in ASCII.
C-----
        DIMENSION X(NDIM), Y(NDIM)
        DIMENSION CARD(1)
        LOGICAL FILE
        CHARACTER*(*) IDENT,FILIN
        CHARACTER*1 AXK(2)

```

```

DATA AXK /'X','Y'/
C
INQUIRE (FILE=FILIN,EXIST=FILE)
IF (FILE) THEN
  OPEN (UNIT=3,FILE=FILIN,STATUS='OLD')
  ICX=1
  WRITE (*,155) AXK(1),ICX
  READ (*,*) ICX
  ICY=2
  WRITE (*,155) AXK(2),ICY
  READ (*,*) ICY
  ICOL = MAX0 (ICX,ICY)
  CALL DATIN (3,X,Y,NDIM,NP,CARD,ICOL,ICX,ICY,IDENT)
  CLOSE (UNIT=3)
END IF
RETURN
155 FORMAT (' Column no. for ',A1,'-data? (/=',I2,') '$)
END
SUBROUTINE DATIN (IDEV,X,Y,NDIM,N,CARD,
1 ICOL,ICX,ICY,IDENT)
C-----
C This subroutine reads data arrays x and y from a data set containing
C data columns in free - format. Description of parameter list:
C
C IDEV - Logical unit no. for input device.
C
C X - Array to contain X-data.
C
C Y - Array to contain Y-data.
C
C NDIM - Dimension of arrays X and Y.
C
C N - Number of data pairs to be computed
C by this subroutine. Reading will
C stop if N .EQ. NDIM.
C
C CARD - Auxiliary storage REAL array. CARD
C must be dimensioned at least as large
C as the largest column to be read.
C
C ICOL - Column no. of last column to be read.
C (i.e. maximum of ICX and ICY).
C
C ICX - No. of column from which X-data is read.
C
C ICY - No. of column from which Y-data is read.
C
C IDENT - BYTE array to contain line to be read.
C
C NDIM,ICOL,ICX,ICY must be assigned values before DATIN is called.
C All other parameters are computed by DATIN.
C
C DATIN stores the first (non-blank) card in IDENT. If this
C card does not contain a sentinel in cols. 1 and 2 it then continues
C to read the first two columns of each succeeding card until it finds
C one with a '/*'. This card serves as a sentinel for the numeric data,
C which is assumed to follow immediately.
C If end-of-file is reached without finding a '/*' in cols. 1 and 2

```

C then DATIN rewinds to the beginning of the file and starts to read
C each line as numeric data, counting the lines (points) and skipping
C those lines which are unintelligible.

C The numeric data must be written in columns, separated by at
C least one space/comma. Otherwise the data format is free. DATIN
C continues to read data until either end-of-file is set or the
C number of data pairs read becomes equal to NDIM.

C
C
C The file to be read must be opened as Unit #IDEV
C in calling program.

C-----

```
DIMENSION X (NDIM), Y (NDIM)
DIMENSION CARD (ICOL)
CHARACTER*(*) IDENT
CHARACTER*2 JUNK
CHARACTER BEL
```

C
C BEL = CHAR (7)
C First look for a '/' in cols. 1&2; if none present, then reset
C to the top.

```
10 READ (IDEV,'(A)',END=2) IDENT
    IF (IDENT .EQ. ' ') GO TO 10
    IF (IDENT(1:2) .EQ. '/') GO TO 3
1   READ (IDEV,'(A)',END=2) JUNK
    IF (JUNK .EQ. '/') GO TO 3
    GO TO 1
2   REWIND IDEV
3   K = 0
4   K = K + 1
    IF (K .GT. NDIM) GO TO 5
    READ (IDEV, *, ERR=7, END=5) CARD
    X (K) = CARD (ICX)
    Y (K) = CARD (ICY)
    GO TO 4
5   N = K - 1
    RETURN
```

C Error-handling portion of code.

```
7   K = K - 1
    GO TO 4
    END
SUBROUTINE ASCOUT (X,Y,N,IDENT,JDENT,FILEN)
```

C-----

C This subroutine writes an output file in ASCII mode.
C Parameter list:

C
C X,Y - output data arrays.
C
C N - no. of data points in each array.
C
C IDENT - 1st header.
C
C JDENT - 2nd header.
C
C FILEN - Output file name.

C
C A card with a '/' in cols. 1,2 appears in front of first

C data card. The data follows in columnar format, one (X,Y)
C pair per line.

C-----

```
DIMENSION X(N),Y(N)
CHARACTER*(*) FILEN,IDENT,JDENT
DATA IDEV /3/
```

C
C First executable statement.

C
C OPEN (UNIT=IDEV,FILE=FILEN,STATUS='UNKNOWN')

C
C LI = LEN (IDENT)
C LJ = LEN (JDENT)
C IF (LI .GT. 0) WRITE (IDEV,'(A)') IDENT(1:LI)
C IF (LJ .GT. 0) WRITE (IDEV,'(A)') JDENT(1:LJ)

C
C WRITE (IDEV,100) N
C WRITE (IDEV,105)
C
C WRITE (IDEV,150) (X(J),Y(J),J=1,N)

C
C CLOSE (UNIT=IDEV)
C RETURN

C-----

C Format statements follow:

C-----

C 100 FORMAT (I5,' data points')

C 105 FORMAT ('/*')

150 FORMAT (G14.6,1X,G15.7)

C-----

END