APPENDIX I

COMPUTER PROGRAMS

```
'************************************************************
' PROGRAM    : PMW-1 (Program #1 for Model Weather)
' WRITTEN BY : Kyoo Dong Song
'            College of Architecture
'            University of Oklahoma
' Program Language : Microsoft QuickBASIC
' Hardware         : IBM-PC/XT/AT and Compatible Machines
'
' This program is to generate the data files consisting of
' cumulative distribution function values of  month/year
' combination data and long term data for each weather element.
'************************************************************
   DIM TX(310),TN(310),TV(310),DT(310)
   DIM WV(310),WX(310),SS(310),SM(310)
   DIM STX(310),STN(310),STV(310),SDT(310)
   DIM SWV(310),SWX(310),SSS(310),SSM(310)
   DIM CUMTX(310),CUMTN(310),CUMTV(310),CUMDT(310)
   DIM CUMWV(310),CUMWX(310),CUMSS(310),CUMSM(310)
'*********************************************************
' LEGENDS :
' TMAX / *TX ... MAXIMUM DRI BULB  TEMP. [F]
' TMIN / *TN ... MINIMUM DRI BULB  TEMP. [F]
' TAVE / *TV ... AVERAGE DRI BULB  TEMP. [F]
' DEWT / *DT ... AVERAGE DEW POINT TEMP. [F]
' WAVE / *WV ... AVERAGE WIND VELOCITY [MPH]
' WMAX / *WX ... MAXIMUM WIND VELOCITY [MPH]
' SCSS / *SS ... SKY COVER FROM SUNRISE TO SUNSET  [10th]
' SCMM / *SM ... SKY COVER FROM MIDNITE TO MIDNITE [10th]
'*********************************************************
  CLS
  LINE INPUT " Enter drive with DATA DISKETTE (A: B: C:) --> ";D$
  PRINT     " Enter Month using below format"
  LINE INPUT " (JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,DEC ) --> ";MO$
  CLS

  FOR ITER = 1 TO 2   '---- 1 FOR MONTH/YEAR, 2 FOR LONG TERM
     N = 0
     RESTORE 40

FOR MOY=1 TO 10 '---------- 10 MONTH/YEAR COMBINATION
     READ YR$
     OPEN D$+"\"+MO$+"\"+MO$+YR$+".DAT" FOR INPUT  AS#1
     INPUT#1,DN
     IF ITER=1 THEN N=0
     IF ITER=1 THEN 1 ELSE IF ITER=2 THEN 2

1  LOCATE 12,20 : PRINT "Reading ";MO$;YR$;"  data  for  monthly CDF":GOTO 3
2  LOCATE 12,20 : PRINT "Reading ";MO$;YR$;"  data for long term CDF"
3  FOR DAY=1 TO DN '---------- DAYS OF A MONTH
     N=N+1
     INPUT#1,DATE,TMAX,TMIN,TAVE,DEWT,WAVE,WMAX,SCSS,SCMM
     TX(N)=TMAX:TN(N)=TMIN:TV(N)=TAVE:DT(N)=DEWT
     STX(N)=TMAX:STN(N)=TMIN:STV(N)=TAVE:SDT(N)=DEWT
     WV(N)=WAVE:WX(N)=WMAX:SS(N)=SCSS:SM(N)=SCMM
     SWV(N)=WAVE:SWX(N)=WMAX:SSS(N)=SCSS:SSM(N)=SCMM
   NEXT DAY

  CLOSE #1
```

```
    IF ITER=1 THEN GOSUB 10:GOSUB 20 '--SORTING / C.F./ FILE for MONTH
  NEXT MOY
    IF ITER=2 THEN GOSUB 10:GOSUB 30 '--SORTING/C.F./FILE for LONG TERM
  NEXT ITER

    CLS
    END ' of the program's main body
'************************************************
' SUBROUTINE to Generate Month/Year data file of
' Cummulative Function for each element
'************************************************

20    LOCATE 12,20 : PRINT "Writing ";MO$;YR$;"  monthly CDF statistic "

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"TX"+".DAT" FOR OUTPUT AS#1
        WRITE #1, O
        FOR I=1 TO O
        WRITE #1, TX(I),CUMTX(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"TN"+".DAT" FOR OUTPUT AS#2
        WRITE #2, P
        FOR I=1 TO P
        WRITE #2, TN(I),CUMTN(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"TV"+".DAT" FOR OUTPUT AS#3
        WRITE #3, Q
        FOR I=1 TO Q
        WRITE #3, TV(I),CUMTV(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"DT"+".DAT" FOR OUTPUT AS#4
        WRITE #4, R
        FOR I=1 TO R
        WRITE #4, DT(I),CUMDT(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"WV"+".DAT" FOR OUTPUT AS#5
        WRITE #5, S
        FOR I=1 TO S
        WRITE #5, WV(I),CUMWV(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"WX"+".DAT" FOR OUTPUT AS#6
        WRITE #6, T
        FOR I=1 TO T
        WRITE #6, WX(I),CUMWX(I)
        NEXT I

    OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"SS"+".DAT" FOR OUTPUT AS#7
        WRITE #7, U
        FOR I=1 TO U
        WRITE #7, SS(I),CUMSS(I)
        NEXT I

     OPEN D$+"\"+MO$+"\month\"+MO$+YR$+"SM"+".DAT" FOR OUTPUT AS#8
         WRITE #8, V
         FOR I=1 TO V
```

```
         WRITE #8, SM(I),CUMSM(I)
       NEXT I
   CLOSE
   RETURN


'*********************************************
' SUBROUTINE to Generate Long Term data file of
' Cummulative Function for each element
'*********************************************

30    LOCATE 12,20 : PRINT "Writing  ";MO$;"  long term  CDF statistic "

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LTX"+".DAT" FOR OUTPUT AS#1
          WRITE #1, O
          FOR I=1 TO O
          WRITE #1, TX(I),CUMTX(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LTN"+".DAT" FOR OUTPUT AS#2
          WRITE #2, P
          FOR I=1 TO P
          WRITE #2, TN(I),CUMTN(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LTV"+".DAT" FOR OUTPUT AS#3
          WRITE #3, Q
          FOR I=1 TO Q
             WRITE #3, TV(I),CUMTV(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LDT"+".DAT" FOR OUTPUT AS#4
          WRITE #4, R
          FOR I=1 TO R
             WRITE #4, DT(I),CUMDT(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LWV"+".DAT" FOR OUTPUT AS#5
          WRITE #5, S
          FOR I=1 TO S
          WRITE#5, WV(I),CUMWV(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LWX"+".DAT" FOR OUTPUT AS#6
          WRITE #6, T
          FOR I=1 TO T
          WRITE #6, WX(I),CUMWX(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LSS"+".DAT" FOR OUTPUT AS#7
          WRITE #7, U
          FOR I=1 TO U
          WRITE #7, SS(I),CUMSS(I)
          NEXT I

      OPEN D$+"\"+MO$+"\long\"+MO$+"-LSM"+".DAT" FOR OUTPUT AS#8
          WRITE #8, V
          FOR I=1 TO V
          WRITE #8, SM(I),CUMSM(I)
```

```
        NEXT I

  CLOSE
  RETURN


'************************************************
' SUBROUTINE  for Sorting & Cumulative Function
'************************************************
10  IF ITER = 1 THEN 11 ELSE IF ITER = 2 THEN 12
11  LOCATE 12,20
   PRINT "Sorting/Calculating ";MO$;YR$;" monthly CDF":GOTO 13
12  LOCATE 12,20
   PRINT "Sorting/Calculating ";MO$;"  long term  CDF"
13  FOR I=1 TO N-1
    FOR J=I+1 TO N

    IF STX(I) > STX(J) THEN SWAP STX(I),STX(J)
    IF STN(I) > STN(J) THEN SWAP STN(I),STN(J)
    IF STV(I) > STV(J) THEN SWAP STV(I),STV(J)
    IF SDT(I) > SDT(J) THEN SWAP SDT(I),SDT(J)
    IF SWV(I) > SWV(J) THEN SWAP SWV(I),SWV(J)
    IF SWX(I) > SWX(J) THEN SWAP SWX(I),SWX(J)
    IF SSS(I) > SSS(J) THEN SWAP SSS(I),SSS(J)
    IF SSM(I) > SSM(J) THEN SWAP SSM(I),SSM(J)

    NEXT J
     NEXT I

    O=0 : P=0 : Q=0 : R=0 : S=0 : T=0 : U=0 : V=0

    FOR I=1 TO N
      IF I=N THEN 100
        IF STX(I) < STX(I+1) THEN 100 ELSE 110
  100       O=O+1
       TX(O) = STX(I)
      CUMTX(O)= I/(N+1)
  110  NEXT I

    FOR I=1 TO N
      IF I=N THEN 200
        IF STN(I) < STN(I+1) THEN 200 ELSE 210
  200       P=P+1
       TN(P) = STN(I)
      CUMTN(P)= I/(N+1)
  210  NEXT I

    FOR I=1 TO N
       IF I=N THEN 300
         IF STV(I) < STV(I+1) THEN 300 ELSE 310
  300        Q=Q+1
        TV(Q) = STV(I)
       CUMTV(Q)= I/(N+1)
  310  NEXT I

    FOR I=1 TO N
      IF I=N THEN 400
          IF SDT(I) < SDT(I+1) THEN 400 ELSE 410
  400       R=R+1
```

```
      DT(R) = SDT(I)
     CUMDT(R)= I/(N+1)
410  NEXT I

  FOR I=1 TO N
    IF I=N THEN 500
      IF SWV(I) < SWV(I+1) THEN 500 ELSE 510
500       S=S+1
      WV(S) = SWV(I)
    CUMWV(S)= I/(N+1)
510  NEXT I

  FOR I=1 TO N
    IF I=N THEN 600
      IF SWX(I) < SWX(I+1) THEN 600 ELSE 610
600       T=T+1
      WX(T) = SWX(I)
    CUMWX(T)= I/(N+1)
610  NEXT I

  FOR I=1 TO N
    IF I=N THEN 700
      IF SSS(I) < SSS(I+1) THEN 700 ELSE 710
700       U=U+1
      SS(U) = SSS(I)
    CUMSS(U)= I/(N+1)
710  NEXT I

  FOR I=1 TO N
     IF I=N THEN 800
       IF SSM(I) < SSM(I+1) THEN 800 ELSE 810
800        V=V+1
      SM(V) = SSM(I)
     CUMSM(V)= I/(N+1)
810  NEXT I

  RETURN

40 DATA 76,77,78,79,80,81,82,83,84,85
```

```
'*********************************************************
' PROGRAM    : PMW-2 (Program #2 for Model Weather)
' WRITTEN BY : Kyoo Dong Song
'            College of Architecture
'            University of  Oklahoma
' Program Language : Microsoft QuickBASIC
' Hardware         : IBM-PC/XT/AT and Compatible Machines
'
' This  program  is to calculate  the  differences  between
' cumulative distribution functions of month/year combination
' data and long term data for each element by Finkelstein-
' Schafer(FS) statistics. Then, by examining weighted sums of
' FS statistic values of each month/year  combinations,  3
' candiate month/years are determined.
'*********************************************************
    DIM LX(310),LCDF(310)
    DIM MX(31), MCDF(31)
    DIM OELMNT$(8),LELMNT$(8),MELMNT$(8),W(8)
    DIM FS(8,10),WS(8,10),WSUM(10),YR$(10)
'*********************************************************
' LEGENDS :
'
' MX(i)   = MONTH/YEAR VALUES FOR EACH ELEMENT
' LX(i)   = LONG TERM  VALUES FOR EACH ELEMENT
' MCDF(i) = MONTH/YEAR CDF VALUES FOR EACH ELEMENT
' LCDF(i) = LONG TERM  CDF VALUES FOR EACH ELEMENT
' SUMD    = SUM OF ABSOLUTE DIFFERENCE BETWEEN
'         LONG TERM CDF AND MONTH/YEAR CDF AT MX(i)
' FS(i,j) = FS STATISTIC (SUMD / N)
'*********************************************************
  CLS
  LINE INPUT " Enter drive with DATA DISKETTE (A: B: C:) --> ";D$
  PRINT     " Enter Month using below format"
  LINE INPUT " (JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,DEC) --> ";MO$
  CLS
'---------------------------
'    READ VARIABLE NAMES
'---------------------------
    FORM$=" ###.#         #.######         #.######         #.######"
    FOR VA = 1 TO 8
       READ OELMNT$(VA)   '---- 8 ORIGINAL ELEMENTS
    NEXT VA

    FOR  VA = 1 TO 8   '---- 8 LONG TERM ELEMENTS
       READ LELMNT$(VA)
    NEXT VA

    FOR VA = 1 TO 8   '---- 8 MONTH/YEAR ELEMENTS
       READ MELMNT$(VA)
    NEXT VA

    FOR VA = 1 TO 8  '----- 8 WEIGHTING FACTORS
      READ W(VA)
    NEXT VA

    FOR MOY = 1 TO 10
      READ YR$(MOY)
    NEXT MOY
FOR VA = 1 TO 8   '--- INPUT LONG TERM CDF FOR EACH ELEMENT
```

```
OPEN D$+"\"+MO$+"\LONG\"+MO$+LELMNT$(VA)+".DAT" FOR INPUT AS#1
    INPUT #1, LN
    FOR K=1 TO LN
      INPUT #1, LX(K),LCDF(K)
    NEXT K

CLOSE

    RESTORE 50

  FOR MOY=1 TO 10 '--- INPUT MONTH/YEAR CDF FOR EACH ELEMENT
  LPRINT " [TABLE   ] CUMULATIVE DISTRIBUTION FUNCTION OF  ";MO$;YR$(MOY)
  LPRINT "============================================================="
  LPRINT TAB(3);OELMNT$(VA);TAB(17);"MONTHLY    LONG TERM    DIFFERENCE"
  LPRINT "============================================================="

OPEN D$+"\"+MO$+"\MONTH\"+MO$+YR$(MOY)+MELMNT$(VA)+".DAT" FOR INPUT AS#1
    INPUT#1, MN
  FOR I=1 TO MN
     INPUT #1, MX(I),MCDF(I)
  NEXT I

CLOSE

GOSUB 100 '---- CALCULATE FS STATISTICS

   NEXT MOY
   NEXT VA

'------------------------------
' CALCULATION OF WEIGHTED SUM
'------------------------------
   FOR MOY =  1 TO 10
   FOR VA =  1 TO 8
       WSUM(MOY) = WSUM(MOY) +  WS(VA,MOY)
   NEXT VA
   NEXT MOY
'---------------------------------------------
'              PRINT OUTPUT
'---------------------------------------------

LPRINT "                   F.S. STATISTIC VALUES FOR ";MO$
LPRINT
LPRINT " ------------------------------------------------------------------"
LPRINT " ELMNT   76    77    78    79    80    81    82    83    84    85
LPRINT " ------------------------------------------------------------------"
LPRINT

   FOR VA = 1 TO 8
      LPRINT TAB(2) OELMNT$(VA);
   FOR MOY = 1 TO 10
      LPRINT TAB(9+(MOY-1)*6);USING "#.###";FS(VA,MOY);
   NEXT MOY
      LPRINT:LPRINT
   NEXT VA

LPRINT " ------------------------------------------------------------------"

   FOR SPACE=1 TO 6: LPRINT : NEXT SPACE
```

```
LPRINT "                    WEIGHTED SUM OF FS STATISTICS"
LPRINT:LPRINT
LPRINT "-------------------------------------------------------------------------"
LPRINT " ELMNT   WF    76    77    78    79    80    81    82    83    84    85 "
LPRINT "-------------------------------------------------------------------------"
LPRINT
    FOR VA = 1 TO 8
      LPRINT TAB(2) OELMNT$(VA);TAB(9);USING"##.#";W(VA);
   FOR MOY = 1 TO 10
      LPRINT TAB(14+(MOY-1)*6);USING "##.##";WS(VA,MOY);
   NEXT MOY
   NEXT VA
LPRINT " -------------------------------------------------------------------------"
LPRINT " WSUM";

   FOR MOY = 1 TO 10
     LPRINT TAB(14+(MOY-1)*6);USING "##.##";WSUM(MOY);
   NEXT MOY

   LPRINT : LPRINT : LPRINT : LPRINT
'-----------------------------------------------------
' Selection of the 3 Candidate Month/Year Combination
'-----------------------------------------------------
  FOR I= 1 TO 9
   FOR J= I+1 TO 10
   IF WSUM(I) > WSUM(J) THEN SWAP WSUM(I),WSUM(J):SWAP YR$(I),YR$(J)
  NEXT J,I

  LPRINT TAB(2);"CANDIDATE   YEARS -----> ";YR$(1);"  ";YR$(2);"  ";YR$(3)
  LPRINT CHR$(12); : LPRINT

END '  of Main Body of the Program

20 DATA TMAX,TMIN,TAVE,DEWT,WAVE,WMAX,SCSS,SCMM
30 DATA -LTX,-LTN,-LTV,-LDT,-LWV,-LWX,-LSS,-LSM
35 DATA TX,TN,TV,DT,WV,WX,SS,SM
'------ Data on statement 40 are weighting factors
40 DATA 2.1,2.1,10.7,10.7,5.0,5.0,3.7,0.5
50 DATA 76,77,78,79,80,81,82,83,84,85
'*******************************************
' SUBROUTINE to print and calcualte FS statistic
'*******************************************
100 SUMD  = 0 : D = 0
    FOR I =  1 TO MN '--- NUMBER OF DAYS FOR  MONTH/YEAR
    FOR K =  1 TO LN '--- TOTAL NUMBER OF DAYS FOR LONGTERM
      IF MX(I) <> LX(K) THEN 60
      D=ABS( MCDF(I) - LCDF(K))
      LPRINT USING FORM$;MX(I),MCDF(I),LCDF(K),D
      SUMD = SUMD + D
60  NEXT K
70  NEXT I
LPRINT "============================================================="
LPRINT TAB(3);"SUM  OF  ABSOLUTE  DIFFERENCE  ";
LPRINT TAB(52);USING"##.######";SUMD
LPRINT : LPRINT : LPRINT
       FS(VA,MOY)= SUMD / MN
       WS(VA,MOY)=FS(VA,MOY) * W(VA)

RETURN
```

```
*****************************************************************
PROGRAM    : GENER.JCL
WRITTEN BY : KYOO DONG SONG
             COLLEGE OF ARCHITECTURE
             UNIVERSITY OF  OKLAHOMA

This program is to place the hourly weather data
from TD-1440 tape onto TSO disk of IBM SYSTEM 3081-K.
*****************************************************************

//TAPE JOB NO1NOZIT,'KYOO DONG SONG',CLASS=A,NOTIFY=LOGONID
/*SETUP T=1
// EXEC IEBGENER
//SYSUT1 DD UNIT=TAPE,VOL=SER=W04147,
//         DISP=(OLD,KEEP),LABEL=(1,NL),
//         DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//SYSUT2 DD DSN=LOGONID.RAW,DISP=(OLD,KEEP),UNIT=TSODISK
//
```

```
//PICKUP JOB NO1NOZIT,'KYOO DONG SONG',CLASS=J,NOTIFY=LOGONID
// EXEC SAS
//IN DD DSN=LOGONID.HOURLY.ALLMODEL,DISP=OLD
//OUT DD DSN=LOGONID.HOURLY.SELMODEL,DISP=OLD
*****************************************************************;
*    PROGRAM: PICKUP                                            *;
*                                                               *;
*    WRITTEN BY : KYOO DONG SONG                                *;
*                 COLLEGE OF ARCHITECTURE                       *;
*                 UNIVERSITY OF  OKLAHOMA                       *;
*                                                               *;
*    PROGRAM LANGUAGE : SAS                                     *;
*    HARDWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)     *;
*                                                               *;
*    THIS PROGRAM IS TO PICK UP MAJOR WEATHER ELEMENTS FROM     *;
*    THE ORIGINAL HOURLY WEATHER DATA SET.                      *;
*****************************************************************;

%MACRO PICKUP(FILE);
  DATA _NULL_;
     FILE OUT(&FILE);
     INFILE IN(&FILE);

       INPUT DATE  $ 6-11 TIME $ 12-13
            TDP 36-38 WDR 39-40 WVL 41-42 SPR 43-46
            TDB 47-49 TWB 50-52 RHU 53-55 CC 79;

       PUT  DATE  1 - 6
           TIME  8 - 9
           TDB  11 - 13
           TDP  15 - 17
           TWB  19 - 21
           RHU  23 - 25
           WVL  27 - 28
           WDR  30 - 31
           SPR  33 - 36
           CC   38  ;
%MEND PICKUP;

*-----------------------------------*;
*   M A I N   R O U T I N E         *;
*-----------------------------------*;
%PICKUP(JAN82);
%PICKUP(FEB85);
%PICKUP(MAR79);
%PICKUP(APR79);
%PICKUP(MAY79);
%PICKUP(JUN85);
%PICKUP(JUL83);
%PICKUP(AUG78);
%PICKUP(SEP81);
%PICKUP(OCT77);
%PICKUP(NOV84);
%PICKUP(DEC81);
//
```

```
//LINTER JOB NO1NOZIT,'KYOO DONG SONG',CLASS=J,NOTIFY=LOGONID
// EXEC FORTVCLG
//FORT.SYSIN DD *
C*****************************************************************
C    PROGRAM : LINTER (LINEAR INTERPOLATION)
C
C    WRITTEN BY : KYOO DONG SONG
C                 COLLEGE OF ARCHITECTURE
C                 UNIVERSITY OF OKLAHOMA
C    PROGRAM LANGUAGE : FORTRAN 77
C    HARDWARE         : IBM SYSEM 3081 MODEL K (MVS/XA-SP)
C
C    THIS PROGRAM IS TO INTERPOLATE THE WEATHER DATA
C    OBSERVED BY 3 HOUR INTERVALS TO GENERATE HOURLY DATA
C    BY LINEAR INTERPOLATION METHOD.
C-----------------------------------------------------------------
C 3 HOURLY VARIABLES
C       X  : SEQUENTIAL NEMBER WITH THE INTERVAL OF 3 (1,4,7,10,..)
C       D  : DATE (YYMMDD)
C       T  : TIME (TT)
C       TD : DRY BULB TEMPERATURE (F)
C       TP : DEW POINT TEMPERATURE (F)
C       WV : WIND VELOCITY (KNOT)
C       WD : WIND DIRECTION
C       SP : STATION PRESSURE (INCH HG)
C
C INTERPOLATED VARIABLES
C      TDB2: DRY BULB TEMP.
C      TDP2: DEWPOINT TEMPERATURE
C      WVL2: WIND VELOCITY
C      WDR2: WIND DIRECTION
C      SPR2: STATION PRESSURE
C
C CALCULATED VARIABLES
C      TWB2: WET BULB TEMPERATURE (F)
C      RHU2: RELATIVE HUMIDITY (%)
C*****************************************************************
       DIMENSION X(248),D(248),T(248)
       DIMENSION TD(248),TP(248)
       DIMENSION WV(248),WD(248),SP(248)
C--------------
C INPUT DATA
C--------------
       N = 248
       NN = 496

       DO 100 I=1,N
       READ(10,*) X(I),D(I),T(I),TD(I),TP(I),W,H,WV(I),WD(I),SP(I)
100     CONTINUE

       DO 200 J=1,NN
         READ(20,*) XX

         INDEX=INT(J/2)*2
C-----------------------------
C DETERMINE BASE POINTS FOR
C LINEAR INTERPOLATION
C-----------------------------
       IF(J.EQ.INDEX) THEN
```

```
         K = J / 2
         TM= T(K) + 2.
       ELSE IF(J.NE.INDEX) THEN
         K = (J + 1)/2
         TM = T(K) + 1.
       ENDIF
         DT = D(K)
C-----------------------
C  INTERPOLATE
C-----------------------
       HX = X(K + 1) - X(K)
       X1 = (X(K + 1) - XX) / HX
       X2 = (X(K) - XX) / HX

       TDB2 = X1 * TD(K) - X2 * TD(K + 1)
       TDP2 = X1 * TP(K) - X2 * TP(K + 1)
       WVL2 = X1 * WV(K) - X2 * WV(K + 1)
       WDR2 = X1 * WD(K) - X2 * WD(K + 1)
       SPR2 = X1 * SP(K) - X2 * SP(K + 1)
C---------------------------------------
C CALCULATE WET BULB TEMPERATURE
C---------------------------------------
       Q = (TDB2 - TDP2) * .1

       IF(TDB2.GE.0) THEN
            TWB2=TDB2-(.034*Q-.00072*Q*(Q-1.))*
    1           (TDB2+TDP2-2.*SPR2*.01+108.)

       ELSEIF(TDB2.LT.0) THEN
            TWB2=TDB2-(.34*Q-.006*Q*Q)*
    1           (.6*(TDB2-TDP2)-2.*SPR2*.01+108.)
       ENDIF
C-----------------------------------
C CALCULATE RELATIVE HUMIDITY
C-----------------------------------
        RHU2=100.*((173.-.1*TDB2+TDP2)/(173.+.9*TDB2))**8
C------------------
C OUTPUT RESULTS
C------------------
        WRITE(30,150) XX,DT,TM,TDB2,TDP2,TWB2,RHU2,WVL2,WDR2,SPR2
150      FORMAT(F4.0,1X,F7.0,1X,F3.0,1X,6F5.0,1X,F6.0)
200      CONTINUE
        STOP
        END

//GO.SYSIN DD DUMMY
//GO.FT10F001 DD DSN=LOGONID.INTERPOL.THREE(MAY79),DISP=OLD
//GO.FT20F001 DD DSN=LOGONID.INTERPOL.POINT(MAY79),DISP=OLD
//GO.FT30F001 DD DSN=LOGONID.INTERPOL.OUTPUT(MAY79),DISP=OLD
//
```

```
//SETTING JOB NO1NOZIT,'KYOO DONG SONG ',CLASS=J,NOTIFY=LOGONID
// EXEC SAS
//THREE DD DSN=LOGONID.INTERPOL.THREE,DISP=OLD
//INTER DD DSN=LOGONID.INTERPOL.OUTPUT,DISP=OLD
//OUT DD DSN=LOGONID.INTERPOL.HOURLY,DISP=OLD
*****************************************************************;
*    PROGRAM    : SETTING                                       *;
*                                                               *;
*    WRITTEN BY : KYOO DONG SONG                                *;
*                 COLLEGE OF ARCHITECTURE                       *;
*                 UNIVERSITY OF OKLAHOMA                        *;
*                                                               *;
*    PROGRAM LANGUAGE : SAS                                     *;
*    HARDWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)     *;
*                                                               *;
*                                                               *;
*    THIS PROGRAM IS TO INTERLEAVE EXISTING 3 HOURLY DATA       *;
*    AND INTERPOLATED HOURLY DATA.                              *;
*****************************************************************;
%MACRO SETTING(FILE);

    DATA THREE;
        INFILE THREE(&FILE);
        INPUT INDEX DATE TIME TDB TDP TWB RHU WVL WDR SPR SKY;

    DATA INTER;
        INFILE INTER(&FILE);
        INPUT INDEX DATE TIME TDB TDP TWB RHU WVL WDR SPR ;

    DATA ALL;
        FILE OUT(&FILE);
        SET THREE INTERPOL;
        BY INDEX;

        PUT  DATE  1 - 6
        TIME  8 - 9
        TDB  11 - 13
        TDP  15 - 17
        TWB  19 - 21
        RHU  23 - 25
        WVL  27 - 28
        WDR  30 - 31
        SPR  33 - 36
        SKY  38   ;
%MEND SETTING;

*********************;
* MAIN ROUTINE      *;
*********************;
%SETTING(MAR79);
%SETTING(APR79);
%SETTING(MAY79);
%SETTING(AUG78);
%SETTING(OCT77);
//
```

```
//SOLRAD JOB NO1NOZIT,'SONG KYOO DONG',CLASS=J,NOTIFY=LOGONID,
// REGION=1024K,TIME=(1,0)
// EXEC FORTVCLG
//FORT.SYSIN DD *
C*************************************************************
C      PROGRAM : SOLRAD (SOLAR RADIATION CALCULATION)
C
C      WRITTEN BY : KYOO DONG SONG
C                   COLLEGE OF ARCHITECTURE
C                   UNIVERSITY OF OKLAHOMA
C
C      PROGRAM LANGUAGE : FORTRAN 77
C      HARDWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)
C
C      THIS PROGRAM IS TO CALCULATE SOLAR RADIATION USING
C      THE METHODS OF ASHRAE AND KIMURA & STEPHENSON
C*************************************************************
C-----------------------------------------------------
C INPUT DATA
C
C SLAT = SITE LATITUDE
C SLON = SITE LONGITUDE
C STMR = STANDARD MERIDIAN
C CN  : SKY CLEARNESS
C-----------------------------------------------------
       SLAT = 35.28
       SLON = 97.32
       STMR = 90.
       CN = .95
       MON = 3
       LDATE = 31
C--------------------------
C START OF CALCULATION
C--------------------------
       PI  = 3.1415927
       RAD = .017453293
       DEG = 57.29577951
       SLAT = SLAT * RAD
       RMO=FLOAT(MON)
C-----------------------------------------------------
C COEFFICIENTS FOR THE CALCULATION
C OF SOLAR RADIATION ON CLOUDY DAYS
C-----------------------------------------------------
       IF ((MON.EQ.2).OR.(MON.EQ.3).OR.(MON.EQ.4)) THEN
          P = 1.06
          Q = .012
          R = -.0084

        ELSEIF ((MON.EQ.5).OR.(MON.EQ.6).OR.(MON.EQ.7)) THEN
          P = .96
          Q = .033
          R = -.0106

        ELSEIF ((MON.EQ.8).OR.(MON.EQ.9).OR.(MON.EQ.10)) THEN
          P = .95
          Q = .03
          R = -.0108

        ELSEIF ((MON.EQ.1).OR.(MON.EQ.11).OR.(MON.EQ.12)) THEN
```

```
          P = 1.14
          Q = .003
          R = -.0082

        ENDIF
       DO 300 IDATE=1,LDATE
          DATE=FLOAT(IDATE)
C-------------------------------------------------------------
C DN : DAY NUMBER (JAN 1 IS 1, DEC 31 IS 365)
C DST : DAYLIGHT SAVINGS TIME
C      (DST IS ASSUMED TO EXIST FROM APRIL 30 TO OCTOBER 31)
C-------------------------------------------------------------
       IF((MON.EQ.1).OR.(MON.EQ.2)) THEN
          DN=31.*(RMO-1.)+DATE
       ELSEIF(MON.EQ.3) THEN
          DN=59.+DATE
       ELSEIF(MON.GT.3) THEN
          DN=90.+AINT(30.5*(RMO-4.))+DATE
          IF((MON.EQ.9).OR.(MON.EQ.11)) DN = DN + 1
       ENDIF
       DST=0.
       IF((DN.GT.120).AND.(DN.LT.304)) DST=1.0
C-----------------------------------------------------
C  DA : DAY ANGLE
C  E  : EQUATION OF TIME
C  D  : DECLINATION ANGLE
C-----------------------------------------------------
       DA=2.*PI*(DN-1.)/365.
       E=(.000075+.001868*COS(DA)-.032077*SIN(DA)
     1    -.014615*COS(2.*DA)-.04089*SIN(2.*DA))*229.18

       D=23.45*SIN((360.*(284.+DN)/365.*RAD))*RAD
 C-------------------------------------------------------------------
C CALCULATE SOLAR ALTITUDE, AZIMUTH AND
C          SOLAR RADIATION UNDER CLEAR SKY
C
C H   : HOUR ANGLE
C TM  : TIME OF THE DAY (LOCAL TIME)
C ALT : SOLAR ALTITUDE ANGLE
C AZI : SOLAR AZIMUTH ANGLE
C A   : APPARENT SOLAR CONSTANT AT AIR MASS OF 0
C B   : EXPONENTIAL ATTENUATION COEFFICIENT
C C   : DIFFUSE FRACTION FACTOR
C RDN : DIRECT RADIATION ON A NORMAL SURFACE (BTU/SQFT-HOUR)
C RDH : DIRECT RADIATION ON A HORIZONTAL SURFACE
C RSH : DIFFUSE RADIATION ON A HORIZONTAL SURFACE
C RTH : TOTAL RADIATION ON A HORIZONTAL SURFACE
C RTHC: TOTAL RADIATION ON A HORIZONTAL SURFACE (CLOUDY SKY)
C RDHC: DIRECT RADIATION ON A HORIZONTAL SURFACE(CLOUDY SKY)
C RSHC: DIFFUSE SKY RADIATION ON A HORIZONTAL SURFACE(CLOUDY SKY)
C-------------------------------------------------------------------
       CDA= COS(DA)
       A = 361.5 + 22.5*CDA
       B = .1745 - .0325*CDA
       C = .0965 - .0395*CDA
       DO 300 ITIME=0,23
       READ(10,*) N,CC
       TM=FLOAT(ITIME)
       SOLTM = TM+(E+4.*(STMR-SLON))/60.-DST
```

```
10      H = (12.-SOLTM)*15.*RAD
        SINALT=SIN(D)*SIN(SLAT)+COS(D)*COS(SLAT)*COS(H)
        ALT=ASIN(SINALT)*DEG
        SINAZI=COS(D)*SIN(H)/COS(ALT*RAD)
        AZI=ABS(ASIN(SINAZI)*DEG)
        IF(COS(H).LE.TAN(D)/TAN(SLAT)) AZI=180.-AZI
        IF(SOLTM.GT.12.) AZI=-AZI

        IF(SINALT.LE.0.) THEN
          RDN=0.
          RDH=0.
          RSH=0.
          RTH=0.
          GOTO 100
        ENDIF

        RDN = A*CN*EXP(-B*SINALT)
        RDH = RDN*SINALT
        RSH = C*RDN/(CN**2)
        RTH = RDH + RSH

        IF(CC.EQ.0) GOTO 100
C-------------------------------------------------------------------
C  CALCULATE SOLAR RADIATION UNDER CLOUDY SKY
C
C  CCF : CLOUD COVER FACTOR
C  RTHC: TOTAL RADIATION ON A HORIZONTAL SURFACE (CLOUDY SKY)
C  RDHC: DIRECT RADIATION ON A HORIZONTAL SURFACE(CLOUDY SKY)
C  RSHC: DIFFUSE SKY RADIATION ON A HORIZONTAL SURFACE(CLOUDY SKY)
C-------------------------------------------------------------------
        CCF = P+Q*CC+R*(CC**2)

        X= SINALT
        Y= .309-.137*X+.394*(X**2)
        RK=X/(C+X)+(P-1.)/(1.-Y)
        RTHC = RTH*CCF
        RDHC = RTH*RK*(1.-CC/10.)
        RSHC = RTHC - RDHC

        RDN = RDHC / X
        RTH = RTHC
        RDH = RDHC
        RSH = RSHC

100     WRITE(20,110) RDN,RDH,RSH,RTH
110     FORMAT(F5.1,1X,F5.1,1X,F5.1,1X,F5.1)

300     CONTINUE

        STOP
        END
//GO.SYSIN DD DUMMY
//GO.FT10F001 DD DSN=LOGONID.SKY.HOURLY(MAR79),DISP=OLD
//GO.FT20F001 DD DSN=LOGONID.HOURLY.SOLAR(MAR79),DISP=OLD
//
//CUBIC JOB NO1NOZIT,'KYOO DONG SONG',CLASS=N,NOTIFY=LOGONID,
// REGION=2048K,TIME=(12,0)
// EXEC FORTVCLG
//FORT.SYSIN DD *
```

```
C*****************************************************************
C    PROGRAM : CUBIC (CUBIC SPLINE INTERPOLATION)
C    WRITTEN BY : KYOO DONG SONG
C                 COLLEGE OF ARCHITECTURE
C                 UNIVERSITY OF OKLAHOMA
C    PROGRAM LANGUAGE : FORTRAN 77
C    HRADWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)
C
C    THIS PROGRAM IS TO SMOOTH THE CONNECTIONS OF
C    ADJACENT MONTHS FROM DIFFERENT YEARS.
C*****************************************************************
       DOUBLE PRECISION DX,XB
       DIMENSION XX(0:10),X2(0:10)
       DIMENSION XI(0:84),Y(0:84)
       DIMENSION TDB(0:84),TDP(0:84),SPR(0:84)
       DIMENSION TDB2(0:84),TDP2(0:84),SPR2(0:84)
       DIMENSION TWB(0:84),RHU(0:84)
       DIMENSION FX(0:84),H1(0:84),H2(0:84),H3(0:84)
       DIMENSION DX(0:84),G2X(0:84)
       DIMENSION A(0:84,0:84),B(0:84)

       COMMON N
       N=84
       NN=10
C------ INPUT EXISTING DATA
       DO 10 I=0,N
10     READ(10,*) XI(I),DATE,TDB(I),TDP(I)

C------ INPUT POINTS TO BE INTERPOLATED
       DO 20 KK=0,NN
         XX(KK)=FLOAT(KK) + 44
20     X2(KK)=XX(KK)
C******************************
C  M A I N   R O U T I N E     *
C******************************
C----- TRANSFER EACH WEATHER ELEMENT TO 'Y' VALUE
      DO 90 INDEX=1,2

        DO 30 I=0,N
30         Y(I)=0.

         IF(INDEX.EQ.1) THEN
             DO 31 I=0,N
31              Y(I)=TDB(I)

         ELSEIF(INDEX.EQ.2) THEN
             DO 32 I=0,N
32              Y(I)=TDP(I)
         ENDIF

C------ CALL CUBIC SPLINE SUBROUTINE

       DO 40 KK=0,NN
          X=XX(KK)
       CALL CUBIC(XI,Y,X,H1,H2,H3,DX,A,B,G2X,GX)

C------ ASSIGN INTERPOLATED VALUES TO EACH
C      INTERPOLATED WEATHER ELEMENT.
```

```
       IF (INDEX.EQ.1) THEN
          TDB2(KK)=GX
       ELSEIF(INDEX.EQ.2) THEN
          TDP2(KK)=GX
       ENDIF

40     CONTINUE
90     CONTINUE
C--------------------------------
C  PRINT OUTPUT OF INTERPOLATION
C--------------------------------
      DO 100 KK=0,NN

100    WRITE(20,110) X2(KK),TDB2(KK),TDP2(KK)
110    FORMAT(F4.0,1X,2F5.0)

      STOP
      END
C*****************************************
C  SUBROUTINE OF CUBIC SPLINE INTERPOLATION *
C*****************************************
       SUBROUTINE CUBIC(XI,FX,X,H1,H2,H3,DX,A,B,G2X,GX)
       COMMON N
       DIMENSION XI(0:N),FX(0:N)
       DIMENSION H1(0:N),H2(0:N),H3(0:N),DX(0:N)
       DIMENSION G2X(0:N),A(0:N,0:N),B(0:N)

C------ INTIALIZE ALL VALUES TO ZERO AND
C      CHECK 'I' IF ITS VALUE IS THE ASSIGNED VALUE
       DO 200 I=0,N
            DX(I)=0.
              G2X(I)=0.
                 B(I)=0.
               H1(I)=0.
              H2(I)=0.
            H3(I)=0.
       IF (I.NE.N) DX(I) = XI(I+1)-XI(I)
       IF((I.NE.N).AND.(I.NE.0)) H1(I)=XI(I+1)-XI(I-1)
       IF (I.NE.N) H2(I)=FX(I+1)-FX(I)
       IF (I.NE.0) H3(I)=FX(I) - FX(I-1)

200    CONTINUE

C------ INITIALIZE MATRIX (A) TO ZERO
       DO 210 I=0,N
         DO 211 J=0,N
            A(I,J)=0.
211      CONTINUE
210    CONTINUE
C------ A(I,J) AND B(I) VALUES ARE COMPUTED,WHERE 1<I<N+1.
C      IF I=0 THEN ALL THE VALUES ARE EQUAL TO ZERO

       DO 220 I=1,N-1
       B(I)=6.*((H2(I)/(DX(I)**2))-(H3(I)/(DX(I)*DX(I-1))))
       A(I,I-1) = DX(I-1)/DX(I)
       A(I,I)=(2.*H1(I))/DX(I)
       A(I,I+1)=1.
220    CONTINUE
```

```
C------ CALL SUBROUTINE TO SOLVE SIMULTANEOUS EQUATION.
       CALL SIMEQ(A,B,G2X)
C------ FIND I VALUES BY COMPARING X VALUES WITH XI VALUES.

       DO 230 I=0,N-1
       IF((X.GE.XI(I)).AND.(X.LE.XI(I+1))) K=I

230     CONTINUE
C------ FIND GX VALUE
       GX = 0.
       GX =(G2X(K)/6.)*(((XI(K+1)-X)**3)/DX(K)-DX(K)*
     1  (XI(K+1)-X))+(G2X(K+1)/6.)*(((X-XI(K))**3)/
     2  DX(K)-DX(K)*(X-XI(K)))+FX(K)*((XI(K+1)-X)/
     3  DX(K))+FX(K+1)*((X-XI(K))/DX(K))
       RETURN
       END
C*********************************************************
C      SUBROUTINE TO SOLVE A SET OF SIMULTANEOUS EQUATIONS *
C      USING GUASS-JORDAN METHOD.                          *
C      N UNKNOWNS IN N EQUATIONS,REPRESENTED IN THE MATRIX *
C      FORM OF (A)*(X)=(B).                                *
C*********************************************************
       SUBROUTINE SIMEQ(A,B,G2X)
       COMMON N
       DIMENSION A(0:N,0:N),B(0:N),G2X(0:N)
C------ DIVIDE ALL ELEMENTS OF MATRICES A,B BY A(I,I) (I=0,1,2,3,..,N)
C      TO MAKE DIAGONAL ELEMENTS OF MATRIX AS UNITY.

       DO 300 I=1,N-1
          XB=A(I,I)
        DO 301 J=0,N
           A(I,J)=A(I,J)/XB
301     CONTINUE
        B(I)=B(I)/XB
C------ SUBTRACT A(K,I)*A(I,J) FROM A(K,J)
C      TO MAKE ALL OTHER ELEMENTS IN KTH ROW AS ZERO (K=0,1,2,....,N
C      & K<>I) AND (J=0,1,2,...,N) AND THEN SUBTRACT A(K,I)*B(I)
C      FROM B(K) TO MAINTIAN EQUALITY,
C      I.E., SUBTRACT A(K,I) TIMES ITH ROW FROM KTH ROW

       DO 302 K=0,N
       IF(K.EQ.I) GOTO 302
       XA = A(K,I)
       DO 303 J=0,N
       A(K,J) = A(K,J) - XA*A(I,J)
303     CONTINUE
       B(K)=B(K)-XA * B(I)
302     CONTINUE
300     CONTINUE
C------ ASSIGN G2X(I)=B(I)
       DO 304 I=0,N
       G2X(I)=B(I)
304   CONTINUE
       RETURN
       END
//GO.SYSIN DD DUMMY
//GO.FT10F001 DD DSN=LOGONID.HOURLY.CONNEC2(NOVDEC),DISP=OLD
//GO.FT20F001 DD DSN=LOGONID.HOURLY.CONNEC3(NOVDEC),DISP=OLD
//
```

```
//UPDATE JOB NO1NOZIT,'KYOO DONG SONG',CLASS=J,NOTIFY=LOGONID,
// REGION=1024K
// EXEC SAS
//IN1 DD DSN=LOGONID.HOURLY.MODEL2,DISP=OLD
//IN2 DD DSN=LOGONID.HOURLY.CONNEC7,DISP=OLD
//OUT DD DSN=LOGONID.HOURLY.MODEL3,DISP=OLD
*****************************************************************;
*    SAS PROGRAM: UPDATE                                        *;
*    WRITTEN BY : KYOO DONG SONG                                *;
*                 COLLEGE OF ARCHITECTURE                       *;
*                 UNIVERSITY OF OKLAHOMA                        *;
*    PROGRAM LANGUAGE : SAS                                     *;
*    HARDWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)     *;
*    THIS PROGRAM IS TO REPLACE DRY BULB AND DEW POINT          *;
*    TEMPERATURES SHOWING SHARP CHANGES AT THE CONNECTIONS      *;
*    OF ADJACENT MONTHS SELECTED FROM DIFFERENT YEARS WITH      *;
*    CUBIC SPLINED DATA. AND WET BULB TEMPERATURE AND           *;
*    RELATIVE HUMIDITY ARE CALCULATED AGAIN.                    *;
*****************************************************************;
%MACRO UPDATE(FILE);
    DATA DSET1;
        INFILE IN1(&FILE);
        INPUT DATE $ TDB TDP TWB RHU WVL WDR SPR CC;
    DATA DSET2;
        INFILE IN2(&FILE);
        INPUT DATE $ TDB TDP;
     DATA DSET3;
        FILE OUT(&FILE);
        UPDATE DSET1 DSET2; BY DATE;
*----------------------------------;
*   CALCULATE WET BULB TEMPERATURE ;
*----------------------------------;
    Q = (TDB - TDP) * .1;
        IF TDB >=0 THEN DO; WB=TDB-(.034*Q-.00072*Q*(Q-1))*(TDB+TDP-2*SPR+108);
        END;
        ELSE IF TDB < 0 THEN DO;
        TWB=TDB-(.34*Q-.006*Q*Q)*(.6*(TDB-TDP)-2*SPR+108); END;
*----------------------------------;
*   CALCULATE RELATIVE HUMIDITY    ;
*----------------------------------;
   RHU=100*((173-.1*TDB+TDP)/(173+.9*TDB))**8 ;
      PUT   DATE  1-6  TDB 8-10 TDP 12-14 TWB 16-17 RHU 19-21
            WVL  23-24 WDR 26-27 SPR 29-33 .2 CC 35-36;
%MEND UPDATE;
*********************************;
*     M A I N  R O U T I N E      *;
*********************************;
%UPDATE(JAN82);
%UPDATE(FEB85);
%UPDATE(MAR79);
%UPDATE(MAY79);
%UPDATE(JUN85);
%UPDATE(JUL83);
%UPDATE(AUG78);
%UPDATE(SEP81);
%UPDATE(OCT77);
%UPDATE(NOV84);
%UPDATE(DEC81);
//
```

```
//SOLMERG JOB NO1NOZIT,'SONG KYOO DONG',CLASS=J,NOTIFY=LOGONID,
// REGION=1024K
// EXEC SAS
//IN1 DD DSN=LOGONID.HOURLY.MODEL3,DISP=OLD
//IN2 DD DSN=LOGONID.HOURLY.SOLAR,DISP=OLD
//OUT DD DSN=LOGONID.HOURLY.MODEL4,DISP=OLD
*************************************************************;
*    SAS PROGRAM: SOLMERG                                    *;
*                                                            *;
*    WRITTEN BY : KYOO DONG SONG                              *;
*                 COLLEGE OF ARCHITECTURE                    *;
*                 UNIVERSITY OF OKLAHOMA                     *;
*                                                            *;
*    PROGRAM LANGUAGE : SAS                                  *;
*    HARDWARE         : IBM SYSTEM 3081 MODEL K (MVS/XA-SP)  *;
*                                                            *;
*    THIS PROGRAM IS TO MERGE HOURLY SOLAR RADIATION DATA    *;
*    CALCULATED BY SOLRAD PROGRAM TO EXISTING HOURLY WEATHER *;
*    DATA WITH NO SOLAR RADIATION DATA.                       *;
*    THE FINAL HOURLY WEATHER DATA SET WAS COMPLETED         *;
*    BY THIS PROGRAM                                        *;
*************************************************************;
OPTIONS MISSING=' ';
%MACRO SOLMERG(INFILE,OUTFILE);
    DATA DSET1;
        INFILE IN1(&INFILE);
        INPUT DATE $
             TDB TDP TWB RHU WVL WDR SPR CC;
    DATA DSET2;
        INFILE IN2(&INFILE);
        INPUT RDN RDH RSH RTH ;

    DATA ALL;
        FILE OUT(&OUTFILE);
        MERGE DSET1 DSET2;

        PUT  DATE  1-6  TDB 8-10  TDP 12-14 TWB 16-17 RHU 19-21
            WVL 23-24  WDR 26-27 SPR 29-33 .2 CC 35-36  RDN 38-42 .1
            RDH 44-48 .1 RSH 50-54 .1 RTH 56-60 .1 ;
%MEND SOLMERG;
*********************;
* MAIN ROUTINE      *;
*********************;
%SOLMERG(JAN82,JAN);
%SOLMERG(FEB85,FEB);
%SOLMERG(MAR79,MAR);
%SOLMERG(APR79,APR);
%SOLMERG(MAY79,MAY);
%SOLMERG(JUN85,JUN);
%SOLMERG(JUL83,JUL);
%SOLMERG(AUG78,AUG);
%SOLMERG(SEP81,SEP);
%SOLMERG(OCT77,OCT);
%SOLMERG(NOV84,NOV);
%SOLMERG(DEC81,DEC);
//
```

```
'****************************************************************
'      PROGRAM : HGL.BAS (HEAT GAIN/LOSS CALCULATION)
'      WRITTEN BY : KYOO DONG SONG
'                   COLLEGE OF ARCHITECTURE
'                   UNIVERSITY OF OKLAHOMA
'                   MAY 1989
'      PROGRAM LANGUAGE : MicroSoft QuickBASIC
'      HARD WARE       : IBM-PC/XT/AT & Compatible Machines
'         THIS PROGRAM IS TO CALCULATE HOURLY BUILDING HEAT
'      GAIN/LOSS THROUGH ROOF, WALLS AND WINDOWS.
'****************************************************************
       CLS
       DIM TSRF(7), TSFW(7), TSBW(7), TSLW(7), TSRW(7)
       DIM BNW(7), DNW(7), BNR(7), DNR(7)
       DIM QCONRF(7), QCONWAL(7), QCONFW(7), QCONBW(7), QCONLW(7), QCONRW(7)
       PI = 3.141592654#
       RAD = .017453293#
       DEG = 57.29577951#
       INPUT "DRIVE FOR DATA INPUT  (A: B: C: etc.) = "; INDRV$
       INPUT "DRIVE FOR DATA OUTPUT (A: B: C: etc.) = "; OUTDRV$
       INPUT "FIRST MONTH (1..12) ="; FMON
       INPUT "LAST  MONTH (1..12) ="; LMON
       PRINT

'--------------------
' DATA FOR BATCH JOB
'--------------------
       AFLOOR = 1000          '---- AREA OF FLOOR
         BH = 10             '---- BUILDING HEIGHT
           WZFW = 0: ORI$ = "S" '---- FRONT WALL ORIENTATION
             NROOF = 30         '---- ASHRAE ROOF #30
              NWALL = 32       '---- ASHRAE WALL #32
            RFWIN = .1      '---- FRONT WALL WINDOW RATIO (10% OF FLOOR AREA)
           RBWIN = .04       '---- BACK WALL WINDOW RATIO (4% OF FLOOR AREA)
         RLWIN = .04         '---- LEFT WALL WINDOW RATIO (4% OF FLOOR AREA)
       RRWIN = .04         '---- RIGHT WALL WINDOW RATIO (4% OF FLOOR AREA )
       GT = .125             '---- GLASS THICKNESS (INCH)
         RF = 1.52           '---- REFRACTION INDEX
          EX = .41           '---- EXTINCTION COEFFICIENT
           UGLASS = 1.05      '---- OVERALL COEFFICIENT OF HEAT TRANSMISSION
          WF = .8            '---- NET GLASS AREA RATIO
         MF = .8             '---- MAINTENANCE FACTOR OF GLASS
       GR = .2               '---- ADJACENT GROUND REFLECTANCE

      OPEN OUTDRV$ + "SIM-" + ORI$ FOR OUTPUT AS #6

'------------------------------------------------------
' INPUT ROOF RESPONSE FACTORS FROM EXTERNAL DATA FILES
'      CNR = SUMMATION CN TERMS OF ROOF TRANSFER FUNCTION
'      BNR = ROOF TRANSFER FUNCTION B VALUES
'      DNR = ROOF TRANSFER FUNCTION D VALUES
'------------------------------------------------------
       PRINT "READING BUILDING RESONSE FACTORS ..."
       OPEN INDRV$ + "ROOF.TRN" FOR INPUT AS #1
       OPEN INDRV$ + "WALL.TRN" FOR INPUT AS #2
10     IF EOF(1) GOTO 20
       INPUT #1, NUMBER, BNROOF, DNROOF, CNR
       IF (NUMBER <> NROOF) THEN
          FOR BB = 0 TO BNROOF
```

```
            INPUT #1, DUMMY
         NEXT BB
         FOR DD = 0 TO DNROOF
            INPUT #1, DUMMY
         NEXT DD
       ELSEIF (NUMBER = NROOF) THEN
         FOR BB = 0 TO BNROOF
            INPUT #1, BNR(BB)
         NEXT BB
         FOR DD = 0 TO DNROOF
            INPUT #1, DNR(DD)
         NEXT DD
         GOTO 20
       END IF
       GOTO 10
20     CLOSE 1

'-------------------------------------------------------
' INPUT THERMAL TRANSFER FUNCTIONS OF WALLS
'      CNW = SUMMATION CN TERMS OF WALL TRANSFER FUNCTION
'      BNW = WALL TRANSFER FUNCTION B VALUES
'      DNW = WALL TRANSFER FUNCTION D VALUES
'-------------------------------------------------------
30     IF EOF(2) GOTO 40
       INPUT #2, NUMBER, BNWALL, DNWALL, CNW
       IF (NUMBER <> NWALL) THEN
         FOR BB = 0 TO BNWALL
            INPUT #2, DUMMY
         NEXT BB
       FOR DD = 0 TO DNWALL
            INPUT #2, DUMMY
         NEXT DD
       ELSEIF (NUMBER = NWALL) THEN
         FOR BB = 0 TO BNWALL
            INPUT #2, BNW(BB)
         NEXT BB
         FOR DD = 0 TO DNWALL
            INPUT #2, DNW(DD)
         NEXT DD
          GOTO 40
       END IF
       GOTO 30
40     CLOSE 2
       CLS
       TIMELAG = BNROOF - 1
       IF (TIMELAG < DNROOF - 1) THEN TIMELAG = DNROOF - 1
       IF (TIMELAG < BNWALL - 1) THEN TIMELAG = BNWALL - 1
       IF (TIMELAG < DNWALL - 1) THEN TIMELAG = DNWALL - 1
       NOW = TIMELAG + 1
       GOSUB REPORT1 '--- REPORT MAXIMUM TIME LAG

'-------------------------------------------
' DETERMINE WALL AZIMUTH ANGLES
'-------------------------------------------
' WZBW = AZIMUTH ANGLE OF BACK WALL (DEGREE)
' WZLW = AZIMUTH ANGLE OF LEFT WALL (DEGREE)
' WZRW = AZIMUTH ANGLE OF RIGHT WALL (DEGREE)
'-------------------------------------------
       WZBW = WZFW + 180
```

```
       WZLW = WZFW + 90
       WZRW = WZFW - 90
       IF (WZBW > 180) THEN
          WZBW = WZBW - 360
            ELSEIF (WZBW < -180) THEN
          WZBW = WZBW + 360
       END IF
       IF (WZLW > 180) THEN
          WZLW = WZLW - 360
            ELSEIF (WZLW < -180) THEN
          WZLW = WZLW + 360
       END IF

       IF (WZRW > 180) THEN
          WZRW = WZRW - 360
            ELSEIF (WZRW < -180) THEN
          WZRW = WZRW + 360
       END IF

FOR CONFIG = 1 TO 41
        SMRGAIN = 0
        WINLOSS = 0
        READ BWR, BLR

'-------------------------------------------------------
' CALCULATE SURFACE AREAS
'-------------------------------------------------------
' ARF = AREA OF ROOF (SQFT)
' AFW = AREA OF FRONT WALL (SQFT)
' ABW = AREA OF BACK ALL (SQFT)
' ALW = AREA OF LEFT WALL (SQFT)
' ARW = AREA OF SIDE WALL #2 (SQFT)
'-------------------------------------------------------

          W$ = STR$(BWR * 10): L$ = STR$(BLR * 10)
             BW$ = RIGHT$(W$, 2): BL$ = RIGHT$(L$, 2)
               WRATIO = BWR / BLR
             BWSQ = AFLOOR * WRATIO
          BW = SQR(BWSQ)  '------- BUILDING WIDTH
        BL = AFLOOR / BW '------- BUILDING LENGTH
       AFWIN = AFLOOR * RFWIN         '---- FRONT WINDOW AREA
        ABWIN = AFLOOR * RBWIN      '---- BACK WINDOW AREA
         ALWIN = AFLOOR * RLWIN       '---- LEFT WINDOW AREA
       ARWIN = AFLOOR * RRWIN         '---- RIGHT WINDOW AREA
       ARF = AFLOOR                    '---- ROOF AREA
         AFW = BW * BH - AFWIN    '---- FRONT WALL AREA
           ABW = BW * BH - ABWIN  '---- BACK WALL AREA
         ALW = BL * BH - ALWIN    '---- LEFT WALL AREA
       ARW = BL * BH - ARWIN      '---- RIGHT WALL AREA

       OPEN OUTDRV$ + ORI$ + BW$ + "." + BL$ FOR OUTPUT AS #3
       GOSUB REPORT2 '--- REPORT BUILDING CONFIGURATION

FOR MONTH = FMON TO LMON

          MHLOSS = 0
          MHGAIN = 0
          MO = MONTH
```

```
       IF (MO = 1) THEN MOCUR$ = "JAN": MOPRE$ = "DEC"
         IF (MO = 2) THEN MOCUR$ = "FEB": MOPRE$ = "JAN"
       IF (MO = 3) THEN MOCUR$ = "MAR": MOPRE$ = "FEB"
         IF (MO = 4) THEN MOCUR$ = "APR": MOPRE$ = "MAR"
       IF (MO = 5) THEN MOCUR$ = "MAY": MOPRE$ = "APR"
         IF (MO = 6) THEN MOCUR$ = "JUN": MOPRE$ = "MAY"
       IF (MO = 7) THEN MOCUR$ = "JUL": MOPRE$ = "JUN"
         IF (MO = 8) THEN MOCUR$ = "AUG": MOPRE$ = "JUL"
       IF (MO = 9) THEN MOCUR$ = "SEP": MOPRE$ = "AUG"
         IF (MO = 10) THEN MOCUR$ = "OCT": MOPRE$ = "SEP"
       IF (MO = 11) THEN MOCUR$ = "NOV": MOPRE$ = "OCT"
         IF (MO = 12) THEN MOCUR$ = "DEC": MOPRE$ = "NOV"
       OPEN INDRV$ + MOPRE$ + ".CLM" FOR INPUT AS #4
       OPEN INDRV$ + MOCUR$ + ".CLM" FOR INPUT AS #5

'-----------------------------------------------------------
' TI = INDOOR TEMPERATURE (F) (SUMMER 75, WINTER 70)
'-----------------------------------------------------------
       IF ((MO >= 5) AND (MO <= 10)) THEN
          TI = 75
       ELSE
          TI = 70
       END IF

'-----------------------------------------------------------
' DETERMINE NUMBER OF DAYS OF PREVIOUS AND CURRENT MONTHS
'-----------------------------------------------------------
 IF (MO = 1 OR MO = 8) THEN NDCUR = 31: NDPRE = 31
 IF (MO = 5 OR MO = 7 OR MO = 10 OR MO = 12) THEN NDCUR = 31: NDPRE = 30
 IF (MO = 2) THEN NDCUR = 28: NDPRE = 31
 IF (MO = 3) THEN NDCUR = 31: NDPRE = 28
 IF (MO = 4 OR MO = 6 OR MO = 9 OR MO = 11) THEN NDCUR = 30: NDPRE = 31
'-------------------------------------------------------------------
' READ HOURLY WEATHER DATA FROM WEATHER DATA FILE
'-------------------------------------------------------------------
' MDT = MONTH,DATE, AND TIME (MMDDYY)
' TDB = DRY BULB TEMP (DEGREE F)
' RDN = DIRECT SOLAR RADIATION ON A NORMAL SURFACE (BTU/SQFT/HR)
' RSH = DIFFUSE SKY RADIATION ON A HORIZONTAL SURFACE
' RTH = TOTAL RADIATION ON A HORIZONTAL SURFACE
' ALT = SOLAR ALTITUDE ANGLE (DEGREE)
' AZI = SOLAR AZIMUTH ANGLE (DEGREE)
'-------------------------------------------------------------------
       COUNT = 0
       I = 0
       LTIME = NDPRE * 24
       INCLUD = LTIME - TIMELAG

'---------------------------------------
' INPUT LAST 6 HOURS' WEATHER DATA OF
' PREVIOUS MONTH AND CALCULATE SOL-AIR
' TEMPERATURES ON ROOF AND WALL SURFACES
'---------------------------------------
       FOR DATE = 1 TO NDPRE
         FOR TIME = 0 TO 23
            COUNT = COUNT + 1
            INPUT #4, MDT, TDB, RDN, RSH, RTH, ALT, AZI
            IF ((COUNT >= INCLUD) AND (COUNT <= LTIME)) THEN
               I = TIMELAG - (LTIME - COUNT)
```

```
              TSRF(I) = TDB + .15 * RTH - 7
              GOSUB SOLAIR
            END IF
         NEXT TIME
       NEXT DATE

'-----------------------------------
' INPUT CURRENT HOUR'S WEATHER DATA
' AND COMPUTE SOL-AIR TEMPERATURES
' AND HEAT GAIN/LOSS.
'-----------------------------------
FOR DATE = 1 TO NDCUR
          DHLOSS = 0
          DHGAIN = 0
   GOSUB REPORT3 '--- REPORT CURRENT DATE OF CALCULATION
  FOR TIME = 0 TO 23
          INPUT #5, MDT, TDB, RDN, RSH, RTH, ALT, AZI
          I = I + 1
          TSRF(I) = TDB + .15 * RTH - 7

'---------------------
' CALL SUBROUTINES
'---------------------
          GOSUB SOLAIR
          GOSUB GLASS
          GOSUB ROOF
          GOSUB WALLS
          QRF = QCONRF(NOW) * ARF
          QFW = QCONFW(NOW) * AFW + QFWIN
          QBW = QCONBW(NOW) * ABW + QBWIN
          QLW = QCONLW(NOW) * ALW + QLWIN
          QRW = QCONRW(NOW) * ARW + QRWIN
          QHOUR = QRF + QFW + QBW + QLW + QRW '---HOURLY HEAT GAIN OR LOSS
     IF (QHOUR < 0) THEN DHLOSS = DHLOSS + QHOUR '--DAILY SUM OF HEAT LOSS
     IF (QHOUR >= 0) THEN DHGAIN = DHGAIN + QHOUR '--DAILY SUM OF HEAT GAIN

'-----------------------------------
' UPDATE SOL-AIR TEMP. AND HEAT FLOW
'-----------------------------------
         FOR I = 0 TO TIMELAG
           TSRF(I) = TSRF(I + 1)
             TSFW(I) = TSFW(I + 1)
              TSBW(I) = TSBW(I + 1)
             TSLW(I) = TSLW(I + 1)
           TSRW(I) = TSRW(I + 1)
           QCONRF(I) = QCONRF(I + 1)
             QCONFW(I) = QCONFW(I + 1)
              QCONBW(I) = QCONBW(I + 1)
             QCONLW(I) = QCONLW(I + 1)
           QCONRW(I) = QCONRW(I + 1)
         NEXT I
              I = I - 1
      NEXT TIME

'------------------------
' PRINT OUTPUT ON DISK
'------------------------
        FORM1$ = " ##    ########    ########"
        MHLOSS = MHLOSS + DHLOSS '----MONTHLY SUM OF HEAT LOSS
```

```
        MHGAIN = MHGAIN + DHGAIN '----MONTHLY SUM OF HEAT GAIN
NEXT DATE
        MAHLOSS = MHLOSS / NDCUR '----DAILY AVERAGE HEAT LOSS
        MAHGAIN = MHGAIN / NDCUR '----DAILY AVERAGE HEAT GAIN
        PRINT #3, USING FORM1$; MO, MAHLOSS, MAHGAIN
        IF (MO = 6 OR MO = 7 OR MO = 8) THEN SMRGAIN = SMRGAIN + MAHGAIN
        IF (MO = 12 OR MO = 1 OR MO = 2) THEN WINLOSS = WINLOSS + MAHLOSS
        CLOSE 4, 5
NEXT MONTH

       AVESMR = SMRGAIN / 3
       AVEWIN = WINLOSS / 3
       GOSUB REPORT4
       CLOSE 3

NEXT CONFIG
     CLOSE 6
     END

'**************************
'SUBROUTINE SOLAIR
'**************************
SOLAIR: FOR WALL = 1 TO 4
         RTWAL = 0
         RVWAL = 0
         TSWAL = 0
         IF RTH = 0 THEN 1100
         IF (WALL = 1) THEN WZ = WZFW
         IF (WALL = 2) THEN WZ = WZBW
         IF (WALL = 3) THEN WZ = WZLW
         IF (WALL = 4) THEN WZ = WZRW
        WSZ = ABS(WZ - AZI)
        IF (WSZ > 180) THEN WSZ = 360 - WSZ
'-----------------------------------
' DIRECT RADIATION ON WALL
' RINC = RADIATION INCIDENCE ANGLE
'-----------------------------------
       COSALT = COS(ALT * RAD)
       IF (WSZ >= 90) GOTO 1000
         RINC = COSALT * COS(WSZ * RAD)
         RVWAL = RDN * RINC
'-----------------------------------
' HOURLY TOTAL RADIATION ON WALL
'-----------------------------------
' RGR = RADIATION REFLECTED FROM GROUND
' RSV = HOURLY SKY DIFFUSE RADIATION ON WALL
' RTWAL = HOURLY TOTAL RADIATION ON WALL
'----------------------------------------
1000    RGR = RTH * GR * .5
       RSV = RSH * .5 + RGR
       RTWAL = (RVWAL + RSV)
'----------------------------------------
' HOURLY SOL-AIR TEMPERATURES
' TSWAL = SOL-AIR TEMPERATURE ON A WALL
'----------------------------------------
1100    TSWAL = TDB + .15 * RTWAL
       IF (WALL = 1) THEN
          RTFW = RTWAL
            WSFW = WSZ
```

```
          FWINC = RINC
         TSFW(I) = TSWAL
        ELSEIF (WALL = 2) THEN
           RTBW = RTWAL
             WSBW = WSZ
           BWINC = RINC
         TSBW(I) = TSWAL
        ELSEIF (WALL = 3) THEN
           RTLW = RTWAL
             WSLW = WSZ
           LWINC = RINC
         TSLW(I) = TSWAL
        ELSEIF (WALL = 4) THEN
            RTRW = RTWAL
              WSRW = WSZ
            RWINC = RINC
         TSRW(I) = TSWAL
        END IF
      NEXT WALL
RETURN
'*************************************************
'SUBROUTINE FOR HEAT GAIN/LOSS THROUGH WINDOW
'*************************************************
GLASS:  FOR WIN = 1 TO 4
        TRAN = 0
        REFL = 0
        ABSO = 0

        IF (WIN = 1) THEN
          WSAZ = WSFW
            INC = FWINC
             RTWAL = RTFW
          GAREA = AFWIN
        END IF
        IF (WIN = 2) THEN
          WSAZ = WSBW
            INC = BWINC
             RTWAL = RTBW
          GAREA = ABWIN
        END IF
        IF (WIN = 3) THEN
          WSAZ = WSLW
            INC = LWINC
             RTWAL = RTLW
          GAREA = ALWIN
        END IF
        IF (WIN = 4) THEN
          WSAZ = WSRW
            INC = RWINC
             RTWAL = RTRW
          GAREA = ARWIN
        END IF
        IF GAREA = 0 THEN 3200
        IF RTH = 0 THEN 3000
        IF (WSAZ >= 90) THEN 3100
'---------------------------------
'CALCULATE GLASS TRANSMITIVITY
'---------------------------------
        SINI = SIN(INC)
```

```
        SINR = SINI / RF
        COSR = SQR(1 - SINR * SINR)
           R = ATN(SINR / COSR)
        IF INC = 0 THEN
           RO = ((RF - 1) / (RF + 1)) ^ 2
           TX = EXP(-EX * L)
        ELSE
           A = SIN(INC - R): AA = A * A
           B = SIN(INC + R): BB = B * B
           C = TAN(INC - R): CC = C * C
           D = TAN(INC + R): DD = D * D
           RO = .5 * ((AA / BB) + (CC / DD))
           TX = EXP(-EX * L / COSR)
        END IF

        ROSQ = RO * RO
          RO1SQ = (1 - RO) * (1 - RO)
             TXSQ = TX * TX
             TRAN = (TX * RO1SQ) / (1 - TXSQ * ROSQ)
          REFL = RO * (1 + (TXSQ * RO1SQ) / (1 - TXSQ * ROSQ))
        ABSO = 1 - TRAN - REFLC
3000    QWIN = TRAN * RTWAL * GAREA * WF * MF + UGLASS * GAREA * (TDB - TI)
        GOTO 3110
3100    QWIN = RSV * GAREA * WF * MF + UGLASS * GAREA * (TDB - TI)
3110    IF (WIN = 1) THEN QFWIN = QWIN
        IF (WIN = 2) THEN QBWIN = QWIN
        IF (WIN = 3) THEN QLWIN = QWIN
        IF (WIN = 4) THEN QRWIN = QWIN
3200    NEXT WIN

RETURN
'************************************************
' SUBROUTINE FOR CONDUCTIVE HEAT GAIN/LOSS
' THROUGH ROOF
'************************************************
ROOF:  SUM1 = 0
       SUM2 = 0

       FOR J = 0 TO BNROOF
         INDEX = NOW - J
         SUM1 = SUM1 + BNR(J) * TSRF(INDEX)
       NEXT J

       FOR K = 1 TO DNROOF
         INDEX = NOW - K
         SUM2 = SUM2 + DNR(K) * QCONRF(INDEX)
       NEXT K

       QCONRF(NOW) = SUM1 - SUM2 - TI * CNR

RETURN
'************************************************
' SUBROUTINE FOR CONDUCTIVE HEAT GAIN/LOSS
' THROUGH OPAQUE WALLS
'************************************************
WALLS:  FOR WALL = 1 TO 4

        SUM1 = 0
        SUM2 = 0
```

```
     FOR T = 0 TO TIMELAG + 1
         IF (WALL = 1) THEN
            TSOL(T) = TSFW(T)
            QCONWAL(T) = QCONFW(T)
         ELSEIF (WALL = 2) THEN
            TSOL(T) = TSBW(T)
            QCONWAL(T) = QCONBW(T)
         ELSEIF (WALL = 3) THEN
            TSOL(T) = TSLW(T)
            QCONWAL(T) = QCONLW(T)
         ELSEIF (WALL = 4) THEN
            TSOL(T) = TSRW(T)
            QCONWAL(T) = QCONRW(T)
         END IF
       NEXT T
       FOR J = 0 TO BNWALL
         INDEX = NOW - J
         SUM1 = SUM1 + BNW(J) * TSOL(INDEX)
       NEXT J
       FOR K = 1 TO DNWALL
         INDEX = NOW - K
         SUM2 = SUM2 + DNW(K) * QCONWAL(INDEX)
       NEXT K
       QCONWAL(NOW) = SUM1 - SUM2 - TI * CNW
       IF (WALL = 1) THEN QCONFW(NOW) = QCONWAL(NOW)
       IF (WALL = 2) THEN QCONBW(NOW) = QCONWAL(NOW)
       IF (WALL = 3) THEN QCONLW(NOW) = QCONWAL(NOW)
       IF (WALL = 4) THEN QCONRW(NOW) = QCONWAL(NOW)
     NEXT WALL
RETURN
'================================================
' SUBROUTINES FOR REPORT ON SCREEN AND OUTPUT FILE
'================================================
REPORT1:
   LOCATE 2, 20: PRINT "MAXIMUM TIME LAG ........"; TIMELAG + 1; "(HOURS)"
RETURN

REPORT2:
   FOR CL = 3 TO 13
      LOCATE CL, 20
      PRINT STRING$(60, " ")
   NEXT CL
   LOCATE 3, 20: PRINT "FRONT WALL ORIENTATION .. "; ORI$
   LOCATE 4, 20: PRINT "WIDTH RATIO TO LENGTH ..."; BWR; ":"; BLR
   LOCATE 6, 20: PRINT "FLOOR AREA ..............."; AFLOOR; " (SQFT)"
   LOCATE 7, 20: PRINT "BUILDING WIDTH ..........."; BW; " (FT)"
   LOCATE 8, 20: PRINT "BUILDING LENGTH ........."; BL; " (FT)"
   LOCATE 9, 20: PRINT "BUILDING HEIGHT ........."; BH; " (FT)"
  LOCATE 10, 20: PRINT "FRONT WINDOW AREA ......."; AFWIN; " (SQFT)"
  LOCATE 11, 20: PRINT "BACK WINDOW AREA ........"; ABWIN; " (SQFT)"
  LOCATE 12, 20: PRINT "LEFT WINDOW AREA ........"; ALWIN; " (SQFT)"
  LOCATE 13, 20: PRINT "RIGHT WINDOW AREA ......."; ARWIN; " (SQFT)"
       PRINT #3, "BLDG ORIENTATION = "; ORI$
       PRINT #3, "WIDTH : LENGTH   = "; BWR; ":"; BLR
       PRINT #3, "============================="
       PRINT #3, "MON    HEAT LOSS    HEAT GAIN "
       PRINT #3, "============================="
RETURN
```

```
REPORT4:
       PRINT #3, "============================="
       PRINT #3, "SUMMER GAIN : "; USING "#######(BTU/DAY)"; AVESMR
       PRINT #3, "WINTER LOSS : "; USING "#######(BTU/DAY)"; AVEWIN
        FORM2$ = " #.#:#.# ######## ########"
       PRINT #6, USING "#.#:#.# ######## ########"; BWR, BLR, AVEWIN, AVESMR
RETURN

REPORT3:
   LOCATE 15, 20
   PRINT "CALCULATING : "; MOCUR$; " "; DATE
RETURN
'=================================================
' DATA FOR BUILDING WIDTH RATIO TO LENGTH (41 CASES)
'=================================================
DATA 5.0,1.0
DATA 4.5,1.0
DATA 4.0,1.0
DATA 3.5,1.0
DATA 3.0,1.0
DATA 2.5,1.0
DATA 2.4,1.0
DATA 2.3,1.0
DATA 2.2,1.0
DATA 2.1,1.0
DATA 2.0,1.0
DATA 1.9,1.0
DATA 1.8,1.0
DATA 1.7,1.0
DATA 1.6,1.0
DATA 1.5,1.0
DATA 1.4,1.0
DATA 1.3,1.0
DATA 1.2,1.0
DATA 1.1,1.0
'
DATA 1.0,1.0
'
DATA 1.0,1.1
DATA 1.0,1.2
DATA 1.0,1.3
DATA 1.0,1.4
DATA 1.0,1.5
DATA 1.0,1.6
DATA 1.0,1.7
DATA 1.0,1.8
DATA 1.0,1.9
DATA 1.0,2.0
DATA 1.0,2.1
DATA 1.0,2.2
DATA 1.0,2.3
DATA 1.0,2.4
DATA 1.0,2.5
DATA 1.0,3.0
DATA 1.0,3.5
DATA 1.0,4.0
DATA 1.0,4.5
DATA 1.0,5.0
```