รายการอ้างอิง

ภาษาไทย

ปราโมทย์ เดชะอำไพ. (รศ. ดร.). ไฟไนต์เอลลิเมนต์ในงานวิศวกรรม. กรุงเทพมหานคร. สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2537.

สิทธิพร รัตโนภาส การไฟฟ้าแห่งประเทศไทย. แผนการผลิตไฟฟ้า และ การประหยัดไฟฟ้า. การสัมนาเรื่อง ELECTRICITY UPDATE, 9 พฤษภาคม 2538.

ภาษาอังกฤษ

American Society of Heating Refigerating and Air-Conditioning Engineers, Inc. ASHRAE HANDBOOK FUNDAMENTALS. Chapter 25, 1977.

American Society of Heating Refigerating and Air-Conditioning Engineers, Inc. ASHRAE HANDBOOK FUNDAMENTALS. Chapter 26, 27 and 28, 1981.

American Society of Heating Refigerating and Air-Conditioning Engineers, Inc. ASHRAE HANDBOOK FUNDAMENTALS. pp. 26.63-26.64, 1993.

B. Mohanty. Energy Auditing and Conservation Methodologies. Workshop on BUILDING ENERGY MANAGEMENT. Bangkok Thailand, Apr 22-26, 1991.

B. Mohanty and P. Bouix. Proceedings of the 5th Asian School on ENERGY CONSERVATION IN BUILDINGS, Jun 15-23, 1987.

BLAST Support Office. BLAST User Reference. volume 1. Illinois. 1991

BLAST Support Office. BLAST User Reference. volume 2. Illinois. 1991

Douglas C. Hittle, CALCULATING BUILDING HEATING AND COOLING LOADS USING THE FREQUENCY RESPONSED OF MULTILAYERED SLABS. Chapter 2, Dec. 1979.

S. sairam and A.H. Azit. Proceedings of The Asean Special Sessions of The ASHRAE Far East Conferance on Air Conditioning in Hot Climates. Kuala Lumpur Malaysia, Oct 26-28, 1989.

Trane Company. TRACE600 (Engineering Manual), Nov 1992

Trane Company. TRACE600 (Output Interpretation Manual), Nov 1992

Trane Company. TRACE600 (User's Manual), Nov 1992

Trane Company. TRACE600 (Utility Reference Manual), Nov 1992

ภาคผนวก

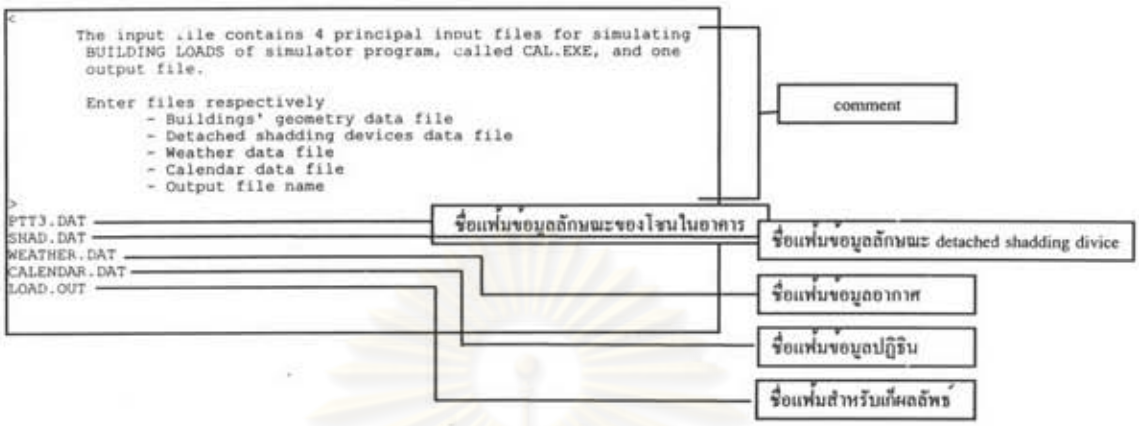
ภาคผนวก ก

คู่มือการใช้โปรแกรม BLN-ESP1

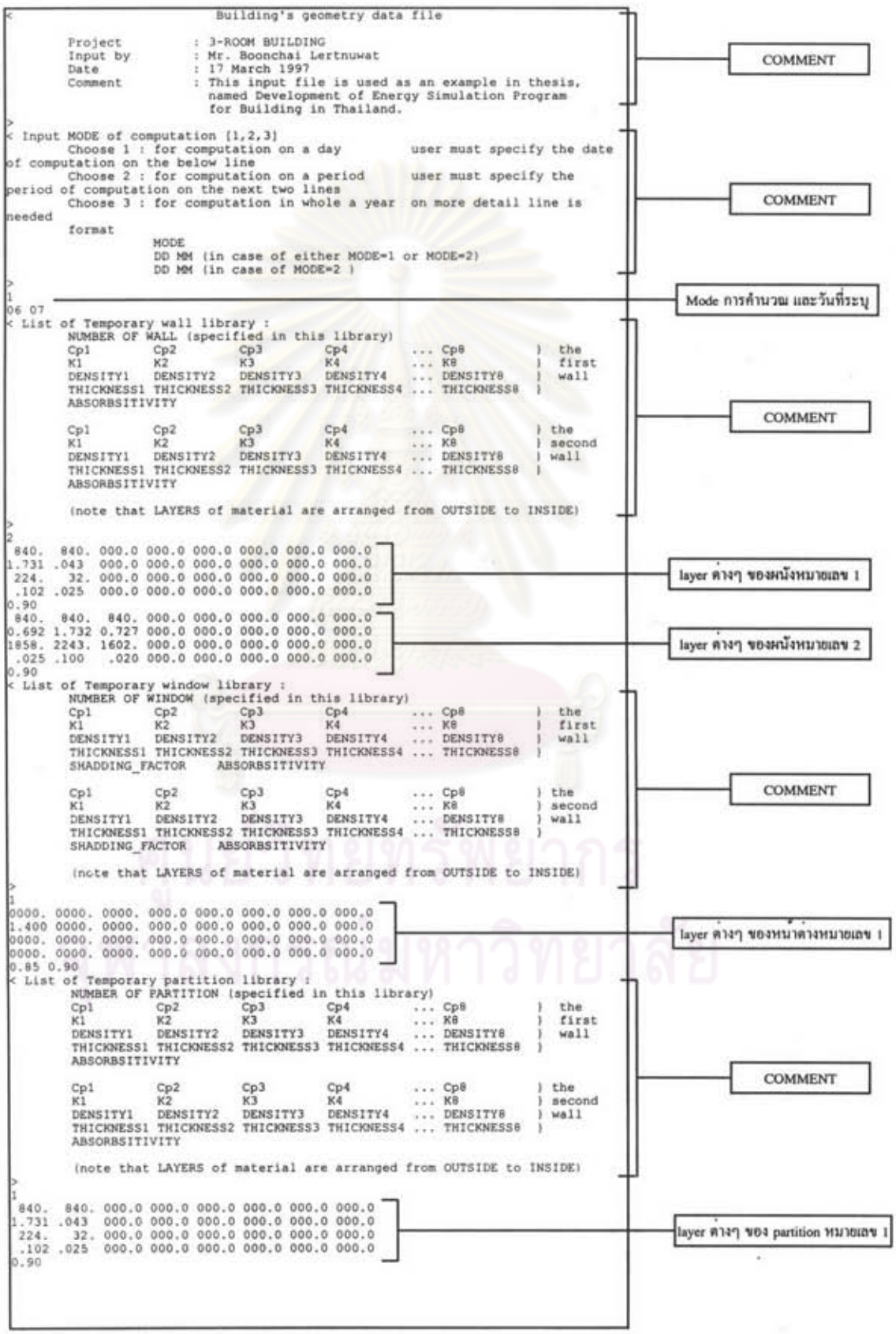
โปรแกรม BLN-ESP1 แบ่งออกเป็น 3 ส่วนคือ โปรแกรม CAL.EXE, SYSTEM.EXE และ REPORT.EXE โดยที่โปรแกรมแต่ละตัวจะมี file ที่เกี่ยวข้องแตกต่างกันไปตามจุดประสงค์ในการคำนวณของแต่ละตัว

CAL.EXE	เป็นโปรแกรมคำนวณ cooling loads, heat-extraction rate และ room temperature ของโซนต่างๆ ในอาคารที่ทำการศึกษา โปรแกรมนี้จะสอบถามผู้ใช้ถึงชื่อของ file ที่เกี่ยวข้องกับการทำงาน 5 files ด้วยกันคือ 1.1 แฟ้มข้อมูลลักษณะของโซนต่างๆ ในอาคาร (ดูรูป ก-2) 1.2 แฟ้มข้อมูลลักษณะของ detached shadding device (ดูรูป ก-3) 1.3 แฟ้มข้อมูลอากาศ (ดูรูป ก-4) 1.4 แฟ้มข้อมูลปฏิทิน (ดูรูป ก-5) 1.5 แฟ้มข้อมูลสำหรับเก็บผลลัพธ์ โดยที่ ชื่อของ file ทั้ง 5 นี้จะถูกใส่ไว้ในแฟ้มข้อมูลชื่อ INPUT (ดูรูป ก-1) เรียงตามลำดับตามที่ได้กล่าวมาข้างต้น
SYSTEM.EXE	เป็นโปรแกรมคำนวณ system loads, systems energy consumption rate, plant load และ plant energy consumption rate ซึ่งจะต้องการใช้ file ที่เกี่ยวข้องตามลำดับ คือ 2.1 แฟ้มข้อมูลลักษณะของ systems และ plant (ดูรูป ก-6) 2.2 แฟ้มข้อมูลของ zone load (แฟ้มข้อมูลที่ 5 ของ CAL.EXE) 2.3 แฟ้มข้อมูลอากาศ (ดูรูป ก-4) 2.4 แฟ้มข้อมูลปฏิทิน (ดูรูป ก-5) 2.5 แฟ้มข้อมูลสำหรับเก็บผลลัพธ์ สำหรับ system 2.6 แฟ้มข้อมูลสำหรับเก็บผลลัพธ์ สำหรับ plant โดยที่โปรแกรม REPORT.EXE จะทำการสอบถามชื่อของแฟ้มข้อมูล และให้ผู้ใช้กรอกชื่อแฟ้มผ่านทางแป้นพิมพ์ตามลำดับ
REPORT.EXE	เป็นโปรแกรมสำหรับสร้างตารางรายงาน หรือจัดระเบียบของผลลัพธ์ ซึ่งจะทำให้ผู้ใช้งานเข้าใจในผลลัพธ์ได้อย่างง่ายๆ โดยที่โปรแกรมจะสอบถามถึงแฟ้มข้อมูลที่เป็นผลลัพธ์ที่ยังมิได้มีการจัดระเบียบ (แฟ้มข้อมูลหมายเลข 1.5, 2.5 หรือ 2.6) เพื่อนำไปจัดพิมพ์ตามรูปแบบที่ผู้ใช้กำหนด

รูป ก-1 ตัวอย่างแฟ้มข้อมูล INPUT

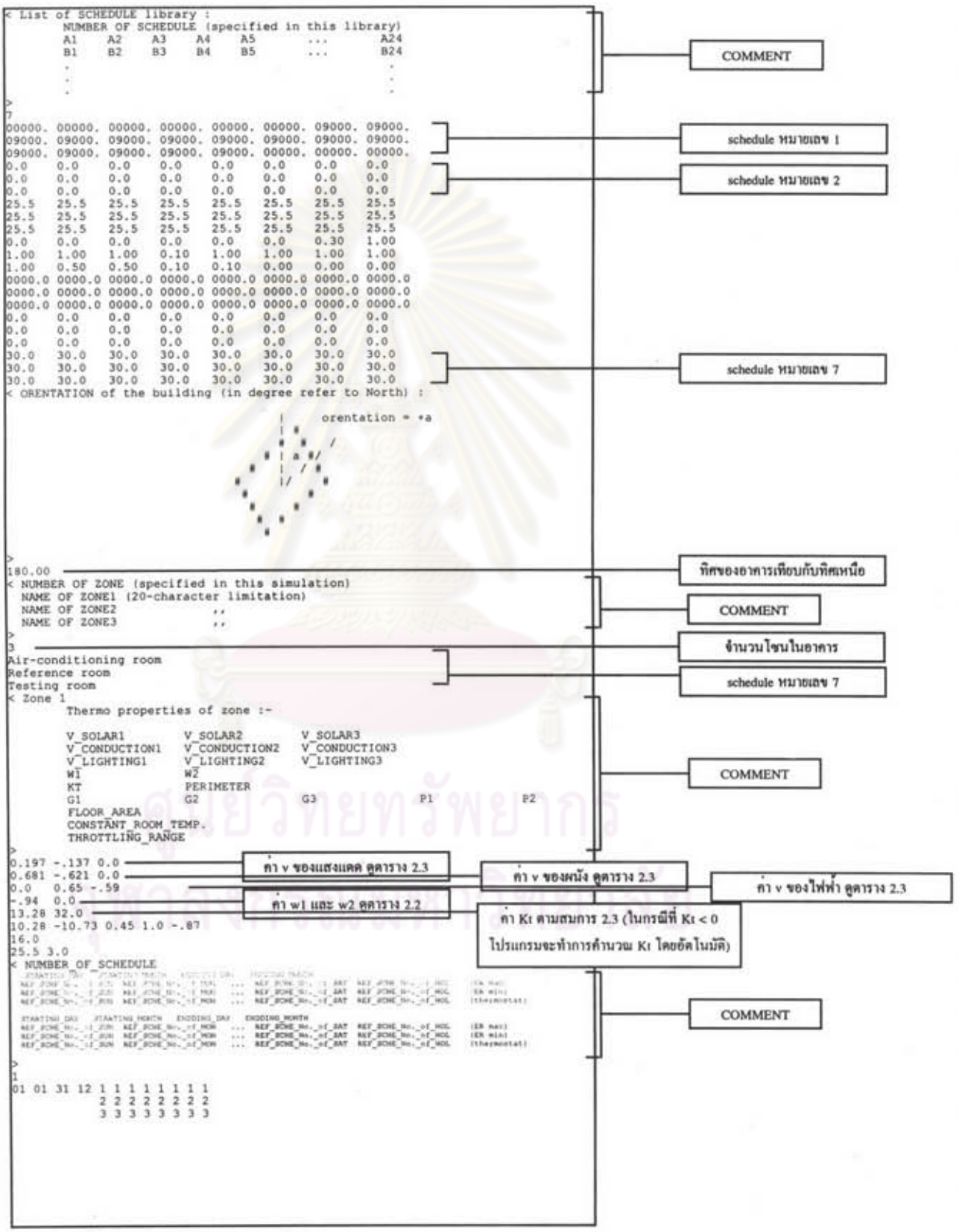


รูป ก-2 ตัวอย่างแฟ้มข้อมูลลักษณะของโซนต่างๆ ในอาคาร

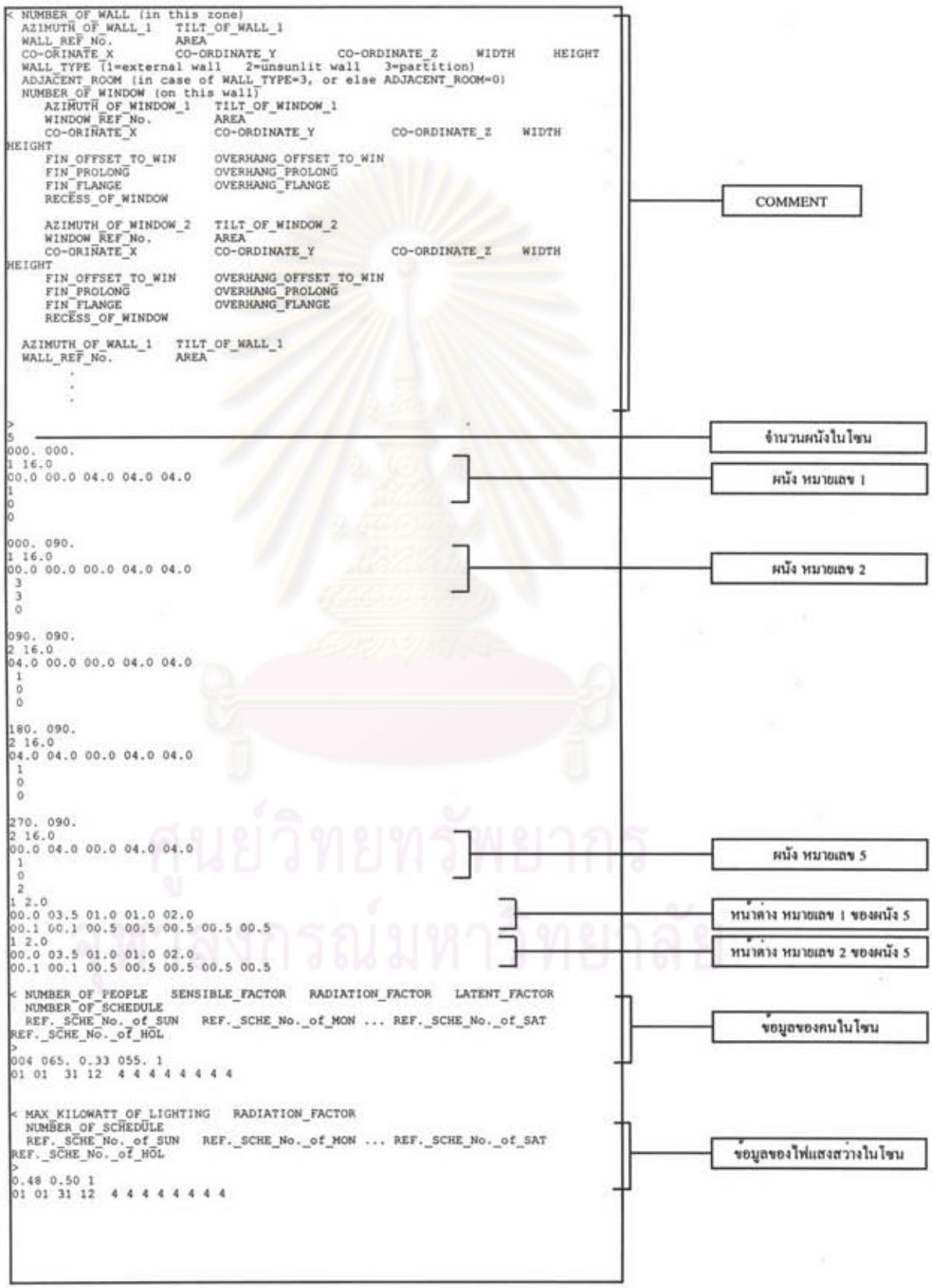



```
<                          Building's geometry data file

        Project          : 3-ROOM BUILDING
        Input by         : Mr. Boonchai Lertnuwat
        Date             : 17 March 1997
        Comment          : This input file is used as an example in thesis,
                           named Development of Energy Simulation Program
                           for Building in Thailand.
>
< Input MODE of computation [1,2,3]
        Choose 1 : for computation on a day          user must specify the date
of computation on the below line
        Choose 2 : for computation on a period       user must specify the
period of computation on the next two lines
        Choose 3 : for computation in whole a year   on more detail line is
needed
        format
                   MODE
                   DD MM (in case of either MODE=1 or MODE=2)
                   DD MM (in case of MODE=2 )
>
1
06 07
< List of Temporary wall library :
        NUMBER OF WALL (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
2
 840.  840. 000.0 000.0 000.0 000.0 000.0 000.0
1.731 .043  000.0 000.0 000.0 000.0 000.0 000.0
 224.   32. 000.0 000.0 000.0 000.0 000.0 000.0
 .102 .025  000.0 000.0 000.0 000.0 000.0 000.0
0.90
 840.  840.  840. 000.0 000.0 000.0 000.0 000.0
0.692 1.732 0.727 000.0 000.0 000.0 000.0 000.0
1858. 2243. 1602. 000.0 000.0 000.0 000.0 000.0
 .025 .100   .020 000.0 000.0 000.0 000.0 000.0
0.90
< List of Temporary window library :
        NUMBER OF WINDOW (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        SHADDING_FACTOR     ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        SHADDING_FACTOR     ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
1
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
1.400 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0.85 0.90
< List of Temporary partition library :
        NUMBER OF PARTITION (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8 )
        ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
1
 840.  840. 000.0 000.0 000.0 000.0 000.0 000.0
1.731 .043  000.0 000.0 000.0 000.0 000.0 000.0
 224.   32. 000.0 000.0 000.0 000.0 000.0 000.0
 .102 .025  000.0 000.0 000.0 000.0 000.0 000.0
0.90
```

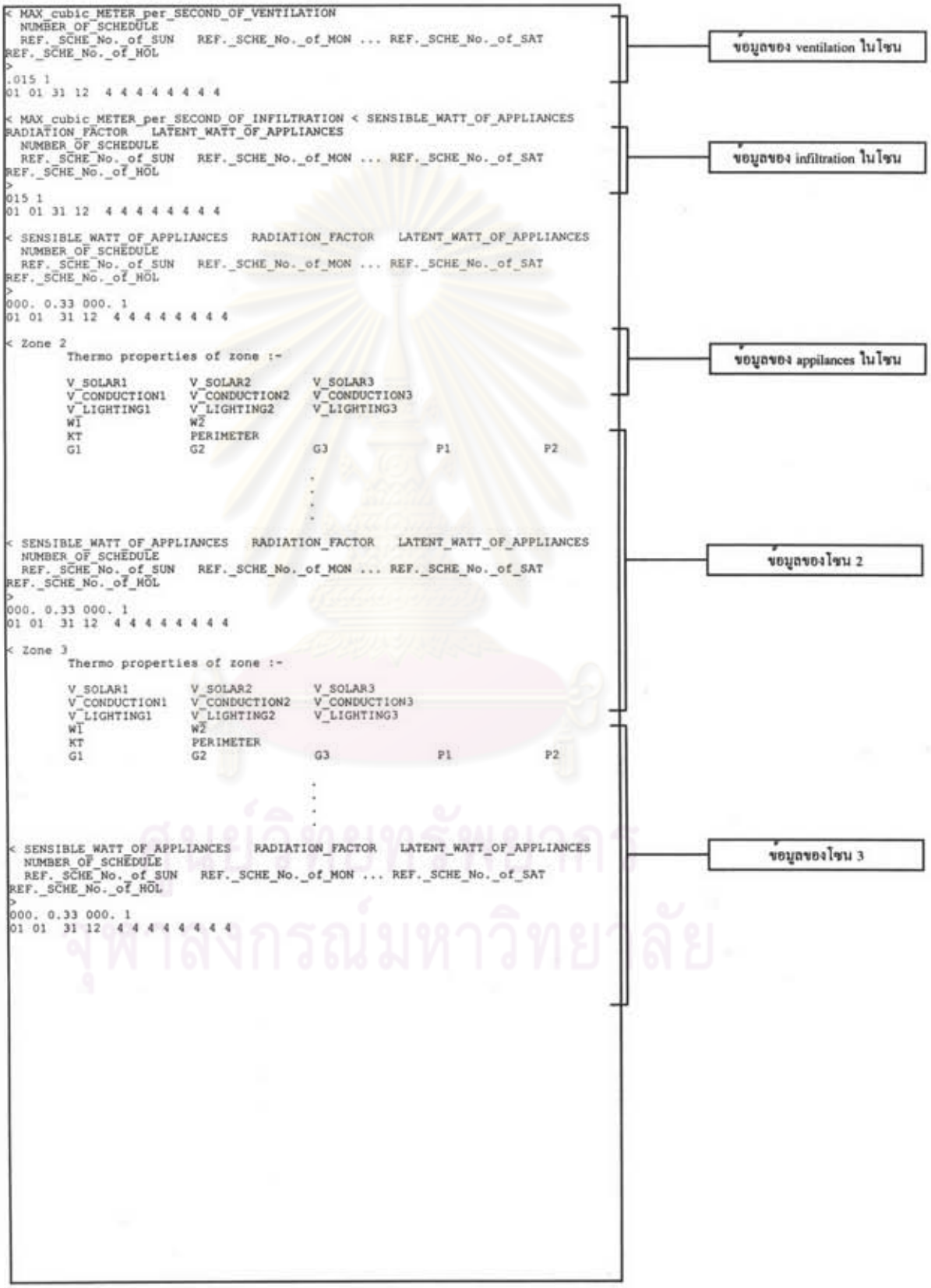
รูป ก-2 ตัวอย่างแฟ้มข้อมูลลักษณะของโซนต่างๆ ในอาคาร (ต่อ)



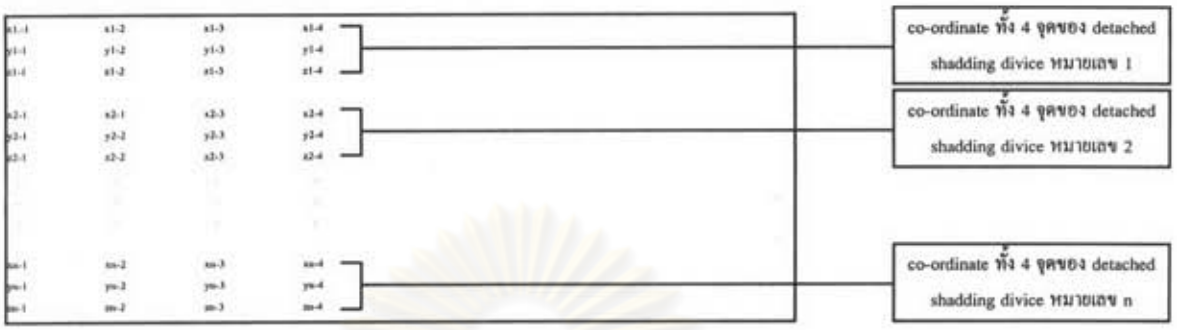
รูป ก-2 ตัวอย่างแฟ้มข้อมูลลักษณะของโซนต่างๆ ในอาคาร (ต่อ)



รูป ก-2 ตัวอย่างแฟ้มข้อมูลลักษณะของโซนต่างๆ ในอาคาร (ต่อ)



รูป ก-3 ตัวอย่างแฟ้มข้อมูลลักษณะของ detached shadding device



รูป ก-4 ตัวอย่างแฟ้มข้อมูลอากาศ

ละติจูด ลองจิจูด ลองจิจูดของเวลา จำนวนวันในปี

```
13.733  -100.567  -105.000      365.
        1         1           1991
 26.200   26.000   25.800   25.700   25.500   24.700   24.100   24.800
 27.000   29.200   30.000   32.400   31.800   33.000   32.800   33.000
 32.400   30.000   28.500   28.000   27.700   27.200   27.000   26.900
 24.800   24.700   24.800   24.700   24.600   23.900   23.400   23.900
 25.000   25.800   25.200   25.100   25.600   25.100   25.600   25.600
 24.900   24.600   24.600   25.000   25.000   25.100   25.400   25.500
   .000     .000     .000     .000     .000     .000     .000   28.056
195.556  443.333  458.056  418.056  201.944  442.778  251.944  730.833
653.611  274.722     .000     .000     .000     .000     .000     .000
   .000     .000     .000     .000     .000     .000     .000   50.833
172.222  234.167  295.000  347.500  394.444  331.111  319.722  132.222
 77.500   35.833     .000     .000     .000     .000     .000     .000
   .019     .019     .019     .019     .019     .018     .018     .018
   .019     .020     .018     .017     .018     .017     .018     .018
   .017     .017     .018     .019     .019     .019     .020     .020
   .000     .000     .000     .000     .000     .000     .000     .000
   .000     .000     .000     .000     .000    4.116    2.572    4.116
  1.543    2.572    3.087    2.572     .000    1.543    2.572     .000
   .000     .000     .000     .000     .000     .000     .000     .000
   .000     .000     .000     .000     .000  160.000  230.000  230.000
210.000  210.000  220.000  220.000     .000  210.000  200.000     .000
        2         1           1991
 27.000   26.800   26.500   26.200   26.000   25.100   25.000   26.100
 27.800   29.400   31.200   31.600   32.100   33.000   33.400   33.700
 32.500   30.100   28.800   28.200   27.900   27.500   27.300   27.000
 25.300   25.200   24.900   24.800   24.600   24.200   24.000   25.000
 25.800   25.800   26.000   26.100   25.400   25.800   25.700   24.500
 25.700   24.800   24.900   25.000   25.000   24.800   24.700   25.000
   .000     .000     .000     .000     .000     .000   20.833  128.611
220.278  282.778  306.111  221.944  156.389   95.000  416.667  396.389
245.278  109.444     .000     .000     .000     .000     .000     .000
   .000     .000     .000     .000     .000     .000    2.500   66.944
171.389  263.333  331.389  383.333  389.167  344.444  290.833  221.667
140.278   41.389     .000     .000     .000     .000     .000     .000
   .020     .020     .019     .019     .019     .019     .018     .020
   .020     .020     .019     .019     .018     .018     .018     .015
   .018     .018     .018     .019     .019     .019     .019     .019
   .000     .000     .000     .000     .000     .000     .000     .000
   .000     .000    2.572    3.601    2.572    3.601    2.572    1.029
  3.087    2.572    3.087    2.572    1.543    1.543    2.572    2.572
   .000     .000     .000     .000     .000     .000     .000     .000
   .000     .000  240.000  210.000  150.000  160.000  170.000  120.000
220.000  220.000  220.000  240.000  210.000  200.000  180.000  180.000
        3         1           1991
                    .
                    .
                    .
                    .
                    .
        31        12        1991
```

dry-bulb temperature 24 ชั่วโมง

wet-bulb temperature 24 ชั่วโมง

direct normal irradiation 24 ชั่วโมง

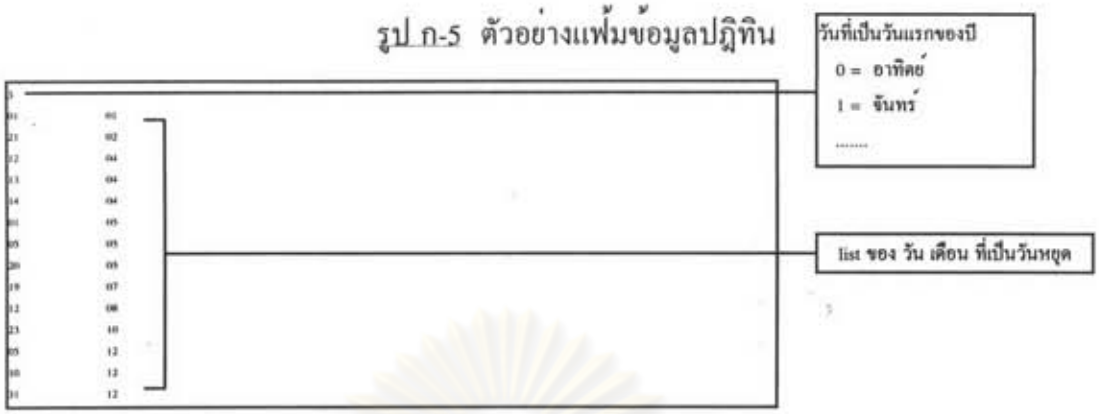
diffused irradiation 24 ชั่วโมง

humidity ratio 24 ชั่วโมง

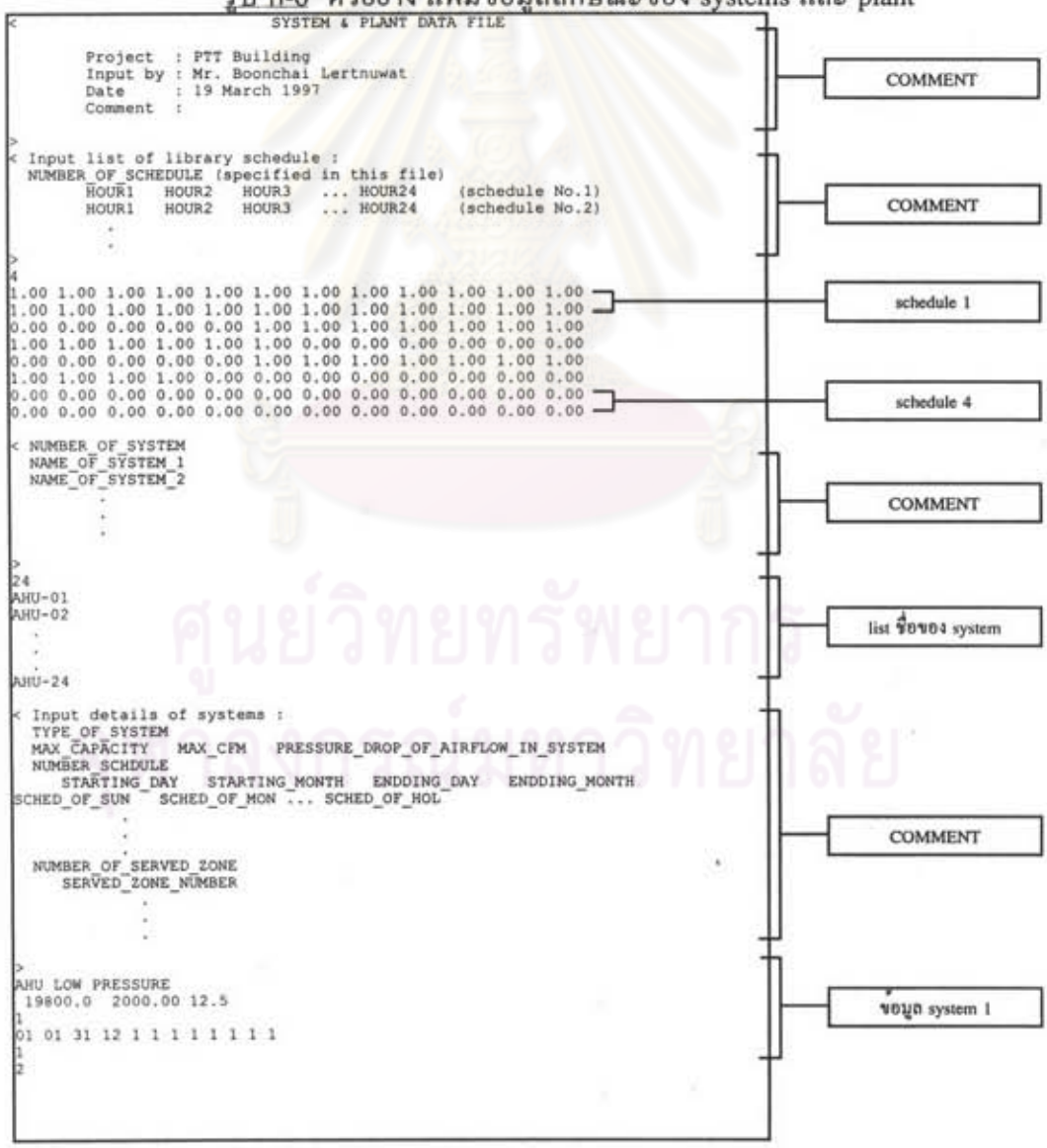
wind speed 24 ชั่วโมง

wind direction 24 ชั่วโมง

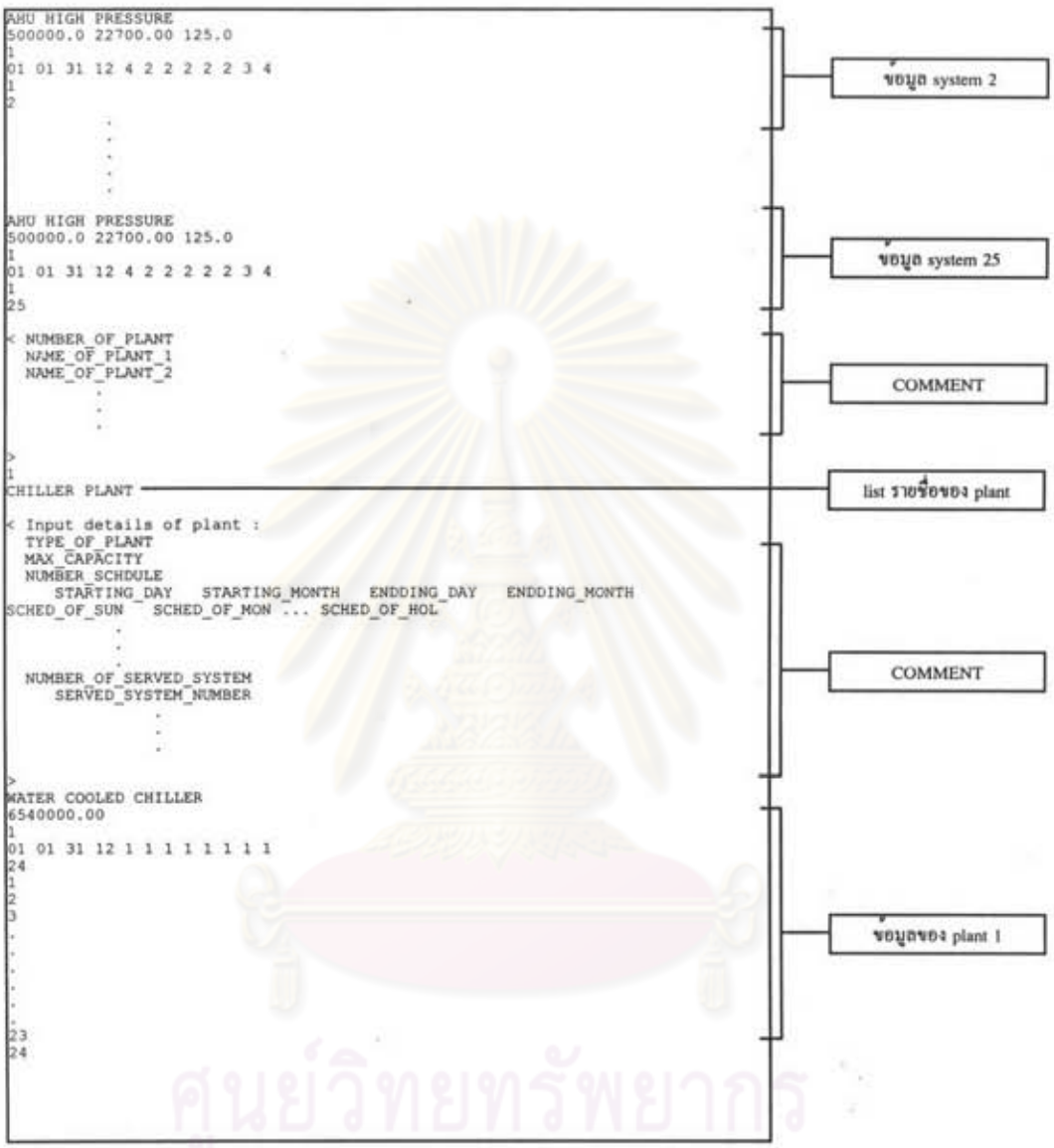
รูป ก-5 ตัวอย่างแฟ้มข้อมูลปฏิทิน



รูป ก-6 ตัวอย่าง แฟ้มข้อมูลลักษณะของ systems และ plant



รูป ก-6 ตัวอย่าง แฟ้มข้อมูลลักษณะของ systems และ plant (ต่อ)



ภาคผนวก ข

SOURCE CODE โปรแกรม CAL, SYSTEM และ REPORT

SOURCE CODE โปรแกรม CAL

```
C     ================================================================
C     |                                                              |
C     |    PROGRAM FOR SIMULATING HEATEXTRACTION RATE IN BUILDINGS   |
C     |                          [SI Unit]                           |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         BUILDING        INPUTFILE OF GEOMETRY OF THE BUILDING @
C     @         CALENDAR        A FILE CONTAINS SCHEDULE OF HOLIDAY   @
C     @         DE_SHAD         FILENAME OF DETACHED SHADING DEVICES  @
C     @         OUTPUT          OUTPUTFILE OF LOADS                   @
C     @         WEATHER         INPUTFILE OF WEATHER DATA             @

      CHARACTER*40 BUILDING,DE_SHAD,WEATHER,CALENDAR,OUTPUT
      INTEGER      MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM

      OPEN(1,FILE='INPUT')
      CALL READ_COMM()                                              *new*
      READ(1,'(A)') BUILDING,DE_SHAD,WEATHER,CALENDAR,OUTPUT
      CLOSE(1)
      CALL SCREEN()
      CALL PARSE_DATA(BUILDING,MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM)
      CALL BLDG_LOAD(DE_SHAD,WEATHER,CALENDAR,OUTPUT,
     +               MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM)
      CALL TAT(0)
      CALL TAT(1)
      CALL TCL(36,40)

      END

      SUBROUTINE PARSE_DATA(BUILDING,
     +                      MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM)
      CHARACTER*40 BUILDING
      INTEGER      MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM
C     ================================================================
C     |                                                              |
C     |       THIS WILL PARSE DATA OF THE BUILDING INTO PARTS AND    |
C     |    THEN WILL SEND SOME DATA, INVOLVING TO PROPERTIES OF      |
C     |    ENVELOPE OF THE BUILDING TO ANOTHER SUBROUTINE TO PROCEED |
C     |    LOAD COMPUTATION IN THAT SUBROTUINE.                      |
C     |                          [SI Unit]                           |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         A               AREA OF WALL (m^2)                   @
C     @         ADJATEMP's      TEMPERATURE OF ADJACENT ROOM         @
C     @         ALPHA           ABSORBSITIVITY OF EXTERNAL WALL      @
C     @         AUTO_KT         AUTOMATIC COMPUTING KT [0=off,1=on]  @
C     @         B's             TRANFER FUNCTION COEFFICIENTS        @
C     @         CFM's           CFM OF SUPPLY AIR (cfm)              @
C     @         CP's            SPECIFIC HEAT CAPACITY (W/kg-C)      @
C     @         DEN's           DENSITY (kg/m^3)                     @
C     @         DUM's           ARRAY OF DUMMY VARIABLES             @
C     @         EQUI_LAT        EQUIPMENT LAT. HEAT GENERATION (W)   @
C     @         EQUI_SEN        EQUIPMENT SEN. HEAT GENERATION (W)   @
C     @         ERR             ERROR INDICATOR                      @
C     @         FA              FLOOR AREA (m^2)                     @
C     @         FH's            FACE OF OVERHANG (m)                 @
C     @         FV's            FACE OF FIN (m)                      @
C     @         GG's            NORMALIZED COEFF. OF THE ROOM AIR    @
C     @         HEIGHT          HEIGHT OF SURFACE (m)                @
C     @         I               COUNTER                              @
C     @         INFILAT's       LATENT HEAT LOAD OF INFILTRATION (W) @
C     @         J               COUNTER                              @
C     @         K               COUNTER                              @
C     @         KILOWATT        KILOWATT OF LIGHTING  (kWATT)        @
C     @         KK's            CONDUCTIVE HEAT TRANSFER COEFFICIENT @
C     @         KT              KT (W/m-C)                           @
C     @         LAT_FAC         OCCUPACY LAT. HEAT GENERATION (W/peo)@
C     @         MODE_CAL        MODE OF CALCULATION                  @
C     @         NUM             NUMBER OF TRASFER FUNCTION COEFF.    @
C     @         NUM_PEO         NUMBER OF PEOPLE                     @
C     @         NUM_ROOM        NUMBER OF ROOM, IN THE BUILDING      @
C     @         NUM_SCHE        NUMBER OF SCHEDULE, USED IN SIMULATION@
C     @         NUM_WALL        NUMBER OF WALL, IN THE ROOM          @
C     @         NUM_WIN         NUMBER OF WIN, ON THE WALL           @
C     @         ORIENTATION     THE ORIENTATION OF THE BUILDING      @
C     @         P's             NORMALIZED COEFF. OF THE ROOM AIR    @
C     @         PERIMETER       PERIMETER OF SPACE (m)               @
C     @         PH's            WIDTH OF OVERHANG (m)                @
C     @         PV's            WIDTH OF FIN (m)                     @
C     @         PX,PY,PZ        POSITION [X,Y,X] (m,m,m)             @
C     @         RAD_FAC         RADIANT FACTOR (%)                   @
C     @         RECESS          RECESS OF WINDOWS (m)                @
C     @         RH's            OFFSET DISTANCE FROM WINDOW (m) [VER] @
C     @         ROOMNAME        NAME OF THE ROOM                     @
C     @         RW's            OFFSET DISTANCE FROM WINDOW (m) [HOR] @
C     @         SAZ             SURFACE AZIMUTH (deg)                @
C     @         SC              SHADDING COEFFICIENT                 @
C     @         SCH_NUM's       No. OF SCHEDULE                      @
C     @         SCH_STPD        STOP DAY OF THE SCHEDULE             @
C     @         SCH_STPM        STOP MONTH OF THE SCHEDULE           @
C     @         SCH_STRD        START DAY OF THE SCHEDULE            @
C     @         SCH_STRM        START MONTH THE SCHEDULE             @
C     @         SEN_FAC         OCCUPACY SEN. HEAT GENERATION (W/peo) @
```

```
C     @          STP_DD          STOP DAY OF PERIOD                    @
C     @          STP_MM          STOP MONTH OF PERIOD                  @
C     @          STR_DD          START DAY OF PERIOD                   @
C     @          STR_MM          START MONTH OF PERIOD                 @
C     @          SUM_UA          SUM OF U*A OF EACH ZONE               @
C     @          THICKNESS's     THICKNESS OF WALL LAYER (m)           @
C     @          THTR            THROTTING RANGE (C)                   @
C     @          TILT            SURFACE TILT ANGLE (deg)              @
C     @          TRC             PRELIMITIVELY ASSUMED ROOM TEMP (C)   @
C     @          U               U-VALUE OF WALL (W/m^2-C)             @
C     @          VENT_RAT        VETILATION MAXIMUM RATE (m^3/s)       @
C     @          V_COND's        ROOM TRANSFER FUNC. COEFF. FOR COND.  @
C     @          V_LIGH's        ROOM TRANSFER FUNC. COEFF. FOR LIGHT  @
C     @          V_SOLA's        ROOM TRANSFER FUNC. COEFF. FOR SOLAR  @
C     @          W's             ROOM TRANSFER FUNCTION COEFF. W       @
C     @          WALLTYPE        WALL TYPE                             @
C     @          WALL_NUM        REFERENCE WALL NUMBER                 @
C     @          WIDTH           WIDTH OF SURFACE (m)                  @
C     @          WIN_NUM         REFERENCE WINDOW NUMBER               @

      REAL     ORIENTATION
      REAL     SAZ,TILT
      REAL     ALPHA
      REAL     CP(8),KK(8),DEN(8),THICKNESS(8),B(24)
      REAL     A,U
      INTEGER  NUM,WALL_NUM,WIN_NUM
      REAL     V_SOLA(3),V_COND(3),V_LIGH(3),W(2)
      REAL     GG(3),P(2)
      REAL     THTR,TRC,KT,SUM_UA,PERIMETER,FA
      INTEGER  AUTO_KT
      REAL     PX,PY,PZ,WIDTH,HEIGHT
      REAL     RW,RH,PV,PH,FV,FH,RECESS
      REAL     ERR
      REAL     DUM(24)
      INTEGER  MODE_CAL
      INTEGER  NUM_SCHE,NUM_ROOM,NUM_WALL,NUM_WIN
      INTEGER  WALLTYPE,ADJAROOM
      CHARACTER*20 ROOMNAME
      INTEGER  NUM_PEO
      REAL     RAD_FAC
      REAL     SEN_FAC,LAT_FAC
      REAL     KILOWATT
      REAL     VENT_RAT,INFI_RAT
      REAL     EQUI_SEN
      INTEGER  SCH_STRD,SCH_STRM
      INTEGER  SCH_STPD,SCH_STPM
      INTEGER  SCH_NUM(8),SCH_NUM2(8),SCH_NUM3(8)
      INTEGER  I,J,K
      INTEGER  STR_DD,STR_MM,STP_DD,STP_MM

      OPEN(1,FILE=BUILDING)
      CALL READ_COMM()                                          *new*

C     ----- READ THE PRERIOD TO BE CALCULATED OF THIS BUILDING  -----
      CALL READ_COMM()                                          *new*
      READ(1,*,ERR=6010,END=6011) MODE_CAL

C     ----- MODE_CAL IS THE MODE OF CALCULATION ::              -----
C     -----              1 ->    ONLY ONE DAY                  -----
C     -----              2 ->    ONE PART OF THE YEAR          -----
C     -----              3 ->    WHOLE A YEAR                  -----
      IF( MODE_CAL.EQ.1 ) THEN
         READ(1,*,ERR=6020,END=6020) STR_DD,STR_MM
         STP_DD = STR_DD
         STP_MM = STR_MM
      ELSE IF( MODE_CAL.EQ.2 ) THEN
         READ(1,*,ERR=6020,END=6020) STR_DD,STR_MM
         READ(1,*,ERR=6020,END=6020) STP_DD,STP_MM
      ELSE IF( MODE_CAL.EQ.3 ) THEN
         STR_DD = 1
         STR_MM = 1
         STP_DD = 31
         STP_MM = 12
      END IF

C     ----- READING DETAIL OF WALLS                           -----
      OPEN(13,FILE='WALL.TMP',ACCESS='DIRECT',RECL=10)
      CALL READ_COMM()                                          *new*
      READ(1,*,ERR=6030,END=6030) NUM_WALL
      DO 5 I=1,NUM_WALL
         READ(1,*,ERR=6040,END=6040) CP,KK,DEN,THICKNESS,ALPHA
         CALL COEFF(CP,KK,DEN,THICKNESS,17.0,B,U,NUM,ERR)
         IF( ERR.NE.0 ) GO TO 6050
         WRITE(13,REC=(I*27-26)) ALPHA
         WRITE(13,REC=(I*27-25)) U
         WRITE(13,REC=(I*27-24)) NUM
         DO 5 J=1,24
    5       WRITE(13,REC=(I*27+J-24)) B(J)
      CLOSE(13)

C     ----- READING DETAIL OF WINDOWS                         -----
      OPEN(14,FILE='WIN.TMP',ACCESS='DIRECT',RECL=10)
      CALL READ_COMM()                                          *new*
      READ(1,*,ERR=6060,END=6060) NUM_WIN
      DO 6 I=1,NUM_WIN
         READ(1,*,ERR=6070,END=6070) CP,KK,DEN,THICKNESS,SC,ALPHA
         CALL COEFF(CP,KK,DEN,THICKNESS,17.0,B,U,NUM,ERR)
         IF( ERR.NE.0 ) GO TO 6080
         WRITE(14,REC=(I*28-27)) SC
         WRITE(14,REC=(I*28-26)) ALPHA
         WRITE(14,REC=(I*28-25)) U
         WRITE(14,REC=(I*28-24)) NUM
         DO 6 J=1,24
    6       WRITE(14,REC=(I*28+J-24)) B(J)
```

```
      CLOSE(14)

C     ----- READING DETAIL OF PARTITIONS                              -----
      OPEN(15,FILE='PARTITIO.TMP',ACCESS='DIRECT',RECL=10)
      CALL READ_COMM()                                                *new*
      READ(1,*,ERR=6090,END=6090) NUM_PAR
      DO 8 I=1,NUM_PAR
         READ(1,*,ERR=6100,END=6100) CP,KK,DEN,THICKNESS,ALPHA
         CALL COEFF(CP,KK,DEN,THICKNESS,8.3,B,U,NUM,ERR)
         IF( ERR.NE.0 ) GO TO 6110
         WRITE(15,REC=(I*52-51)) ALPHA
         WRITE(15,REC=(I*52-50)) U
         WRITE(15,REC=(I*52-49)) NUM
         DO 7 J=1,24
    7       WRITE(15,REC=(I*52+J-49)) B(J)
         CALL RE_ARR(CP,KK,DEN,THICKNESS,8)
         CALL COEFF(CP,KK,DEN,THICKNESS,8.3,B,U,NUM,ERR)
         WRITE(15,REC=(I*52-24)) NUM
         DO 8 J=1,24
    8       WRITE(15,REC=(I*52+J-24)) B(J)
      CLOSE(15)

C     ----- READING DETAIL OF SCHEDULES                                -----
      CALL READ_COMM()                                                *new*
      READ(1,*,ERR=6120,END=6120) NUM_SCHE
      OPEN(12,FILE='SCHEDULE.TMP',ACCESS='DIRECT',RECL=10)
      DO 10 I=1,NUM_SCHE
         READ(1,*,ERR=6130,END=6130)  DUM
         DO 10 J=1,24
   10       WRITE(12,REC=(I*24+J-24)) DUM(J)
      CLOSE(12)

C     ----- COPY ONLY DATA,REVOLVING CONSTRUCTION OF THE BUILDING -----
C     ----- AT FIRST, READ THE ORIENTATION OF THE BUILDING         -----
      CALL READ_COMM()                                                *new*
      READ(1,*,ERR=6135,END=6135) ORIENTATION

C     ----- THEN KEEP ALL ROOMS' NAME OF THE BUILDING              -----
      OPEN(16,FILE='ROOM.TMP')
      CALL READ_COMM()                                                *new*
      READ(1,*,ERR=6140,END=6140) NUM_ROOM
      WRITE(16,*) NUM_ROOM
      DO 20  I=1,NUM_ROOM
         READ(1,'(A20)',ERR=6150,END=6150) ROOMNAME
         WRITE(16,'(A20)') ROOMNAME
   20 CONTINUE

C     ----- FINALLY, RETAIN DATA OF EACH ROOM IN THE BUILDING      -----
      OPEN(7,FILE='BUILDING.TMP')

      DO 100 I=1,NUM_ROOM
         OPEN(100,FILE='DUMMY.TMP')                                   *new*
         CALL READ_COMM()                                             *new*
         READ(1,*,ERR=6160,END=6160)
     +                V_SOLA,V_COND,V_LIGH,W,KT,PERIMETER,GG,P,FA
         WRITE(100,*) V_SOLA,V_COND,V_LIGH,W,PERIMETER,GG,P,FA
         WRITE(16,*) FA
         IF( KT.LT.0.0 ) THEN
            AUTO_KT = 1
            KT = 0.0
         ELSE
            AUTO_KT = 0
         END IF

         READ(1,*,ERR=6160,END=6160) TRC,THTR
         WRITE(100,*)                TRC,THTR

         CALL READ_COMM()                                             *new*
         READ(1,*,ERR=6170,END=6170)  NUM_SCHE
         WRITE(100,*)                 NUM_SCHE
         DO 25 J=1,NUM_SCHE
            READ(1,*,ERR=6180,END=6180)
     +                 SCH_STRD,SCH_STRM,
     +                 SCH_STPD,SCH_STPM,
     +                 SCH_NUM,SCH_NUM2,SCH_NUM3
            WRITE(100,*) SCH_STRD,SCH_STRM,
     +                   SCH_STPD,SCH_STPM,
     +                   SCH_NUM,SCH_NUM2,SCH_NUM3
   25    CONTINUE

         CALL READ_COMM()                                             *new*
         READ(1,*,ERR=6190,END=6190) NUM_WALL
         WRITE(100,*)                NUM_WALL

         SUM_UA = 0.0
         DO 30 J=1,NUM_WALL
            READ(1,*,ERR=6200,END=6200) SAZ,TILT,WALL_NUM,A,
     +              PX,PY,PZ,WIDTH,HEIGHT,WALLTYPE,ADJAROOM,NUM_WIN
            IF( HEIGHT.EQ.0. .OR. WIDTH.EQ.0. ) GO TO 6210
            IF( WALLTYPE.EQ.3 ) THEN
               IF( ADJAROOM.LE.0 ) GO TO 6211
               IF( ADJAROOM.GT.NUM_ROOM ) GO TO 6212
            END IF

            IF( WALLTYPE.EQ.1 .OR. WALLTYPE.EQ.2 ) THEN
               OPEN(13,FILE='WALL.TMP',ACCESS='DIRECT',RECL=10)
               READ(13,REC=(WALL_NUM*27-25),ERR=6320,END=6320) U
               CLOSE(13)
               IF( AUTO_KT.EQ.1 ) KT = KT + U*A/PERIMETER
            ELSE IF( WALLTYPE.EQ.3 ) THEN
               OPEN(15,FILE='PARTITIO.TMP',ACCESS='DIRECT',RECL=10)
               READ(15,REC=(ABS(WALL_NUM)*52-50),ERR=6320,END=6320) U
               CLOSE(15)
            END IF
```

```
            SUM_UA = SUM_UA + U*A

C     ----- CONVERT POSITION OF EACH WALL FROM LOCAL COORDINATE   -----
C     ----- INTO GOBAL COORDINATE.                                -----
            CALL COORDINATE(ORIENTATION,SAZ,PX,PY)

            WRITE(100,*) SAZ,TILT,WALL_NUM,A,
     +                   PX,PY,PZ,WIDTH,HEIGHT,WALLTYPE,ADJAROOM,NUM_WIN
            DO 30 K=1,NUM_WIN
               READ(1,*,ERR=6220,END=6220) WIN_NUM,A,
     +                PX,PY,PZ,WIDTH,HEIGHT,RW,RH,PV,PH,FV,FH,RECESS
               OPEN(14,FILE='WIN.TMP',ACCESS='DIRECT',RECL=10)
               READ(14,REC=(WIN_NUM*28-25),ERR=6321,END=6321) U
               CLOSE(14)
               IF( WALLTYPE.EQ.1 .OR. WALLTYPE.EQ.2 ) THEN
                  IF( AUTO_KT.EQ.1 ) KT = KT + U*A/PERIMETER
               END IF
               SUM_UA = SUM_UA + U*A

C     ----- CONVERT POSITION OF EACH WINDOW FROM LOCAL COORDINATE -----
C     ----- INTO GOBAL COORDINATE.                                -----
               CALL COORDINATE(ORIENTATION,SAZ,PX,PY)

               WRITE(100,*) WIN_NUM,A,
     +                      PX,PY,PZ,WIDTH,HEIGHT,RW,RH,PV,PH,FV,FH,RECESS

   30    CONTINUE
         CLOSE(100)                                                    *new*

         OPEN(100,FILE='DUMMY.TMP')                                    *new*
         READ(100,*) V_SOLA,V_COND,V_LIGH,W,PERIMETER,GG,P,FA
         WRITE(7,*)  V_SOLA,V_COND,V_LIGH,W,KT,SUM_UA,PERIMETER,GG,P,FA *new*

         READ(100,*) TRC,THTR
         WRITE(7,*)  TRC,THTR

         READ(100,*) NUM_SCHE
         WRITE(7,*)  NUM_SCHE
         DO 35 J=1,NUM_SCHE
            READ(100,*) SCH_STRD,SCH_STRM,
     +                  SCH_STPD,SCH_STPM,
     +                  SCH_NUM,SCH_NUM2,SCH_NUM3
            WRITE(7,*)  SCH_STRD,SCH_STRM,
     +                  SCH_STPD,SCH_STPM,
     +                  SCH_NUM,SCH_NUM2,SCH_NUM3
   35    CONTINUE

         READ(100,*) NUM_WALL
         WRITE(7,*)  NUM_WALL

         DO 40 J=1,NUM_WALL
            READ(100,*) SAZ,TILT,WALL_NUM,A,
     +                  PX,PY,PZ,WIDTH,HEIGHT,WALLTYPE,ADJAROOM,NUM_WIN
            WRITE(7,*)  SAZ,TILT,WALL_NUM,A,
     +                  PX,PY,PZ,WIDTH,HEIGHT,WALLTYPE,ADJAROOM,NUM_WIN
            DO 40 K=1,NUM_WIN
               READ(100,*) WIN_NUM,A,
     +                     PX,PY,PZ,WIDTH,HEIGHT,RW,RH,PV,PH,FV,FH,RECESS
               WRITE(7,*)  WIN_NUM,A,
     +                     PX,PY,PZ,WIDTH,HEIGHT,RW,RH,PV,PH,FV,FH,RECESS

   40    CONTINUE
         CLOSE(100)

         CALL READ_COMM()                                              *new*
         READ(1,*,ERR=6230,END=6230)
     +          NUM_PEO,SEN_FAC,RAD_FAC,LAT_FAC,NUM_SCHE
         WRITE(7,*) NUM_PEO,SEN_FAC,RAD_FAC,LAT_FAC,NUM_SCHE
         DO 50 J=1,NUM_SCHE
            READ(1,*,ERR=6231,END=6231) SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
            WRITE(7,*)                  SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
   50    CONTINUE

         CALL READ_COMM()                                              *new*
         READ(1,*,ERR=6240,END=6240)  KILOWATT,RAD_FAC,NUM_SCHE
         WRITE(7,*)                   KILOWATT,RAD_FAC,NUM_SCHE
         DO 60 J=1,NUM_SCHE
            READ(1,*,ERR=6241,END=6241) SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
            WRITE(7,*)                  SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
   60    CONTINUE

         CALL READ_COMM()                                              *new*
         READ(1,*,ERR=6250,END=6250)  VENT_RAT,NUM_SCHE
         WRITE(7,*)                   VENT_RAT,NUM_SCHE
         DO 70 J=1,NUM_SCHE
            READ(1,*,ERR=6251,END=6251) SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
            WRITE(7,*)                  SCH_STRD,SCH_STRM,
     +                                  SCH_STPD,SCH_STPM,
     +                                  SCH_NUM
   70    CONTINUE

         CALL READ_COMM()                                              *new*
         READ(1,*,ERR=6260,END=6260)  INFI_RAT,NUM_SCHE
         WRITE(7,*)                   INFI_RAT,NUM_SCHE
```

```
      DO 80 J=1,NUM_SCHE
         READ(1,*,ERR=6261,END=6261) SCH_STRD,SCH_STRM,
     +                               SCH_STPD,SCH_STPM,
     +                               SCH_NUM
         WRITE(7,*)                  SCH_STRD,SCH_STRM,
     +                               SCH_STPD,SCH_STPM,
     +                               SCH_NUM
   80    CONTINUE

      CALL READ_COMM()                                                  *new*
      READ(1,*,ERR=6270,END=6270)
     +           EQUI_SEN,RAD_FAC,EQUI_LAT,NUM_SCHE
      WRITE(7,*) EQUI_SEN,RAD_FAC,EQUI_LAT,NUM_SCHE
      DO 90 J=1,NUM_SCHE
         READ(1,*,ERR=6271,END=6271) SCH_STRD,SCH_STRM,
     +                               SCH_STPD,SCH_STPM,
     +                               SCH_NUM
         WRITE(7,*)                  SCH_STRD,SCH_STRM,
     +                               SCH_STPD,SCH_STPM,
     +                               SCH_NUM
   90    CONTINUE

  100 CONTINUE

      CLOSE(7)
      CLOSE(16)
      CLOSE(1)

      RETURN

C     ------------------------ ERROR JUNCTION -------------------------
 6010 CALL ERROR(1,0,0,0)
 6011 CALL ERROR(2,0,0,0)
 6020 CALL ERROR(3,0,0,0)
 6030 CALL ERROR(4,0,0,0)
 6040 CALL ERROR(5,I,0,0)
 6050 CALL ERROR(6,I,0,0)
 6060 CALL ERROR(7,0,0,0)
 6070 CALL ERROR(8,I,0,0)
 6080 CALL ERROR(9,I,0,0)
 6090 CALL ERROR(10,0,0,0)
 6100 CALL ERROR(11,I,0,0)
 6110 CALL ERROR(12,I,0,0)
 6120 CALL ERROR(13,0,0,0)
 6130 CALL ERROR(14,0,0,0)
 6135 CALL ERROR(70,0,0,0)
 6140 CALL ERROR(15,0,0,0)
 6150 CALL ERROR(16,I,0,0)
 6160 CALL ERROR(17,I,0,0)
 6170 CALL ERROR(18,I,0,0)
 6180 CALL ERROR(19,I,0,0)
 6190 CALL ERROR(20,I,0,0)
 6200 CALL ERROR(21,J,I,0)
 6210 CALL ERROR(22,J,I,0)
 6211 CALL ERROR(23,J,I,0)
 6212 CALL ERROR(24,J,I,0)
 6220 CALL ERROR(25,K,J,I)
 6230 CALL ERROR(26,I,0,0)
 6231 CALL ERROR(27,I,0,0)
 6240 CALL ERROR(28,I,0,0)
 6241 CALL ERROR(29,I,0,0)
 6250 CALL ERROR(30,I,0,0)
 6251 CALL ERROR(31,I,0,0)
 6260 CALL ERROR(32,I,0,0)
 6261 CALL ERROR(33,I,0,0)
 6270 CALL ERROR(34,I,0,0)
 6271 CALL ERROR(35,I,0,0)
 6320 CALL ERROR(46,J,I,0)
 6321 CALL ERROR(47,K,J,I)

      RETURN
      END

      SUBROUTINE READ_COMM()
C     ======================================================================
C     |                                                                    |
C     | THIS SUBROUTINE WILL ELIMINATE THE COMMENT ON THE INPUT FILE.      |
C     | IT READ "<" AS A BEGINING SIGN OF COMMENT AND ">" AS THE END.      |
C     |                                                                    |
C     ======================================================================

C     @ ---- ASSIGN VARIABLES                                             @
C     @         FIRST           THE FIRST CHARACTER OF LINE              @

      CHARACTER    FIRST

    5 READ(1,'(A)',END=6000) FIRST
      IF( FIRST.EQ.' ' ) GO TO 5
      IF( FIRST.NE.'<' ) GO TO 6000

   10 READ(1,'(A)',END=6010) FIRST
      IF( FIRST.NE.'>' ) GO TO 10

      RETURN

 6000 CALL ERROR(71,0,0,0)
 6010 CALL ERROR(72,0,0,0)
      END

      SUBROUTINE COORDINATE(ORIENTATION,SAZ,PX,PY)
      REAL ORIENTATION,SAZ,PX,PY
C     ======================================================================
C     |                                                                    |
C     | THIS SUBROUTINE IS SET FOR CONVERT POSITION OF POINT (PX,PY).      |
```

```
C     | FROM LOCAL PLAN INTO GOBAL PLAN, BY ROTATING LOCAL PLAN FOR   |
C     | AN ANGLE OF ORIENTATION.                                      |
C     |                                                               |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         ANGLE           ANGLE OF RADIUS                     @
C     @         DEG             CONVERTOR [deg -> rad]              @
C     @         RADIUS          DISTRANCE OF POINT (PX,PY) TO (0,0) @

C     ----- ASSIGN CONVERT VALUE INTO DEG                           -----
      DEG = ATAN(1.0)/45.0

C     ----- ROTATE SURFACE AZIMUTH FOR AN ANGLE OF ORIENTATION      -----
      SAZ = SAZ + ORIENTATION

C     ----- REWRITE COORDINATE (PX,PY) IN ORTHOGONAL TYPE TO BE     -----
C     ----- CYLINDRICAL TYPE.                                       -----
      RADIUS = (PX**2 + PY**2)**0.5
      IF( RADIUS.EQ.0.0 ) RETURN

      IF( PX.EQ.0.0 ) THEN
         IF( PY.GT.0.0 ) THEN
            ANGLE =  90.0
         ELSE
            ANGLE = 270.0
         END IF
      ELSE
         ANGLE = ATAN(PY/PX)/DEG
         IF( PX.LT.0.0 ) ANGLE = ANGLE + 180.0
      END IF

C     ----- ROTATE VALUE OF ANGLE FOR A VALUE OF ORIENTATION        -----
      ANGLE = ANGLE + ORIENTATION

C     ----- TRANSFER FROM CYLINDRICAL COORDINATE TO BE ORTHOGONAL   -----
C     ----- COORDINATE ANGAIN.                                      -----
      PX = RADIUS*COS(ANGLE*DEG)
      PY = RADIUS*SIN(ANGLE*DEG)

      RETURN
      END

      SUBROUTINE BLDG_LOAD(DE_SHAD,WEATHER,CALENDAR,OUTPUT,
     +                     MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM)
      CHARACTER*40 DE_SHAD,WEATHER,CALENDAR,OUTPUT
      INTEGER      MODE_CAL,STR_DD,STR_MM,STP_DD,STP_MM
C     ================================================================
C     |                                                               |
C     |       PROGRAM FOR SIMULATING ENERGY USES IN BUILDINGS         |
C     |                        [SI Unit]                              |
C     |                                                               |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         A               AREA OF WALL (m^2)                  @
C     @         ADJAROOM        No. OF ADJACENT ROOM                @
C     @         ADJATEMP's      TEMPERATURE OF ADJACENT ROOM        @
C     @         ALPHA           ABSORBSITIVITY OF EXTERNAL WALL     @
C     @         ALT's           SOLAR ALTITUDE (rad)                @
C     @         AZM's           SOLAR AZIMUTH (rad)                 @
C     @         B's             TRANFER FUNCTION COEFFICIENTS       @
C     @         CFM's           CFM OF SUPPLY AIR (cfm)             @
C     @         COSINE's        COSINE OF SOLAR IRRADIATION ON SURFACE@
C     @         DA's            DAY                                 @
C     @         DATE            NUMBER OF DATE COMPARED TO MXDAY    @
C     @         DAY_T's         DUMMY OF DAY_TYPE                   @
C     @         DAY_TYPE        DAY TYPE (0=SUN,1=MON, ..and so on) @
C     @         DBT's           DRY BULB TEMPERATURE (C) [DAILY]    @
C     @         DBT1's          DRY BULB TEMPERATURE (C)  [WEEKLY]  @
C     @         DEG             CONVERTOR [deg -> rad]              @
C     @         DUMMY?          DUMMY VARIABLE                      @
C     @         END_PER         INDICATOR OF END OF PERIOD          @
C     @         ENERGY_C's      ENERGY CONSUMED BY PLANT (W)        @
C     @         EQUILAT's       LATENT HEAT LOAD OF EQUIPMENT (W)   @
C     @         EQUISEN's       SENSIBLE HEAT LOAD OF EQUIPMENT (W) @
C     @         EQUI_LAT        EQUIPMENT LAT. HEAT GENERATION (W)  @
C     @         EQUI_SEN        EQUIPMENT SEN. HEAT GENERATION (W)  @
C     @         ER's            SUPPLIED CAPACITY OF SYSTEM [EACH Hr] @
C     @         ERMN's          MIN. CAPACITY OF SYSTEM (W)         @
C     @         ERMX's          MAX. CAPACITY OF SYSTEM (W)         @
C     @         ERR             ERROR INDICATOR                     @
C     @         FA              FLOOR AREA (m^2)                    @
C     @         FH's            FACE OF OVERHANG (m)                @
C     @         FV's            FACE OF FIN (m)                     @
C     @         GG's            NORMALIZED COEFF. OF THE ROOM AIR   @
C     @         HEIGHT          HEIGHT OF SURFACE (m)               @
C     @         HG's            HEATGAIN (W) [DAILY]                @
C     @         HG1's           HEATGAIN (W) [WEEKLY]               @
C     @         HO's            CONVECTIVE HEATTRANSFER COEFFICIENT @
C     @         HO_DD           DAY OF HOLIDAY                      @
C     @         HO_MM           MONTH OF HOLIDAY                    @
C     @         HR's            RELATIVE HUMIDITY        [DAILY]    @
C     @         HR1's           RELATIVE HUMIDITY         [WEEKLY]  @
C     @         I               COUNTER                             @
C     @         ID's            DIFFUSED SOLAR RADIATION (W/m^2)    @
C     @                                                  [DAILY]    @
C     @         ID1's           DIFFUSED SOLAR RADIATION (W/m^2)    @
C     @                                                   [WEEKLY]  @
C     @         IDN'S           DIRECT NORMAL SOLAR RADIATION (W/m^2) @
C     @                                                    [DAILY]  @
C     @         IDN1's          DIRECT NORMAL SOLAR RADIATION (W/m^2) @
C     @                                                   [WEEKLY]  @
C     @         IDUMMY          DUMMY VARIABLE                      @
```

```
C     @         INFI_RAT      INFILTRATION MAXIMUM RATE (m^3/s)      @
C     @         INFILAT's     LATENT HEAT LOAD OF INFILTRATION (W)   @
C     @         INFISEN's     SENSIBLE HEAT LOAD OF INFILTRATION (W)@
C     @         ITERATION     ITERATION OF COMPUTING                 @
C     @         J             COUNTER                                @
C     @         K             COUNTER                                @
C     @         KILOWATT      KILOWATT OF LIGHTING  (kWATT)          @
C     @         KT            KT (W/m-C)                             @
C     @         L             COUNTER                                @
C     @         LAT_FAC       OCCUPACY LAT. HEAT GENERATION (W/peo)  @
C     @         LIGHT's       LIGHTING LOAD (W)                      @
C     @         LLT           LATITUDE (deg)                         @
C     @         LON           LONGITUDE (deg)                        @
C     @         LSM           LOCAL STANDARD TIME MERIDIAN (deg)     @
C     @         MO's          MONTH                                  @
C     @         MXDAY         MAXIMUM No. OF DAY IN COMPUTED YEAR    @
C     @         NUM           NUMBER OF TRASFER FUNCTION COEFF.      @
C     @         NUM_DAY       No. OF DAY THOSE HAVE TO BE COMPUTED   @
C     @         NUM_PEO       NUMBER OF PEOPLE                       @
C     @         NUM_ROOM      NUMBER OF ROOM, IN THE BUILDING        @
C     @         NUM_SCHE      NUMBER OF SCHEDULE, USED IN SIMULATION@
C     @         NUM_WALL      NUMBER OF WALL, IN THE ROOM            @
C     @         NUM_WIN       NUMBER OF WIN, ON THE WALL             @
C     @         P's           NORMALIZED COEFF. OF THE ROOM AIR      @
C     @         PEOLAT's      LATENT HEAT LOAD OF OCCUPANCY (W)      @
C     @         PEOSEN's      SENSIBLE HEAT LOAD OF OCCUPANCY (W)    @
C     @         PERIMETER     PERIMETER OF SPACE (m)                 @
C     @         PH's          WIDTH OF OVERHANG (m)                  @
C     @         POWER_F's     POWER USED BY FAN or BLOWER (W)        @
C     @         PV's          WIDTH OF FIN (m)                       @
C     @         PX,PY,PZ      POSITION [X,Y,X] (m,m,m)               @
C     @         QL's          COOLING LOAD (W) [DAILY]               @
C     @         QL1's         COOLING LOAD (W) [WEEKLY]              @
C     @         RAD_FAC       RADIANT FACTOR (%)                     @
C     @         RECESS        RECESS OF WINDOWS (m)                  @
C     @         RH's          OFFSET DISTANCE FROM WINDOW (m) [VER]  @
C     @         ROOMNAME      NAME OF THE ROOM                       @
C     @         ROOM_NUM      ROOM NUMBER OF COMPUTED ROOM           @
C     @         RW's          OFFSET DISTANCE FROM WINDOW (m) [HOR]  @
C     @         SAZ           SURFACE AZIMUTH (deg)                  @
C     @         SC            SHADDING COEFFICIENT                   @
C     @         SCHEDULE's    SCHEDULE                               @
C     @         SCH_NUM's     No. OF SCHEDULE                        @
C     @         SCH_STPD      STOP DAY OF THE SCHEDULE               @
C     @         SCH_STPM      STOP MONTH OF THE SCHEDULE             @
C     @         SCH_STRD      START DAY OF THE SCHEDULE              @
C     @         SCH_STRM      START MONTH THE SCHEDULE               @
C     @         SEN_FAC       OCCUPACY SEN. HEAT GENERATION (W/peo)  @
C     @         SHGF's        SLOAR HEAT GAIN FACTOR                 @
C     @         SOLTEMP's     SOL-AIR TEMPERATURE                    @
C     @         SUNLIT_A's    SUNLIT AREA (m^2)                      @
C     @         SUM_UA        SUM OF U*A OF EACH ZONE                @
C     @         THTR          THROTTING RANGE (C)                    @
C     @         TILT          SURFACE TILT ANGLE (deg)               @
C     @         TRC           PRELIMITIVELY ASSUMED ROOM TEMP (C)    @
C     @         TRTS's        THERMOSTAT SETTING POINT (C)           @
C     @         TRT's         TEMPERATURE OF ROOM (C)                @
C     @         U             U-VALUE OF WALL (W/m^2-C)              @
C     @         UNIT_NUM      UNIT_NUM OF DETACHED SHADING DEVICES   @
C     @                       FILE                                   @
C     @         V's           VENTILATION RATE [WEEKLY]              @
C     @         VENT_RAT      VETILATION MAXIMUM RATE (m^3/s)        @
C     @         VENTLAT's     LATENT HEAT LOAD OF VENTILATION (W)    @
C     @         VENTSEN's     SENSIBLE HEAT LOAD OF VENTILATION (W)  @
C     @         VI's          VENTILATION RATE [WEEKLY]              @
C     @         V_COND's      ROOM TRANSFER FUNC. COEFF. FOR COND.   @
C     @         V_LIGH's      ROOM TRANSFER FUNC. COEFF. FOR LIGHT   @
C     @         V_SOLA's      ROOM TRANSFER FUNC. COEFF. FOR SOLAR   @
C     @         W's           ROOM TRANSFER FUNCTION COEFF. W        @
C     @         WALLCOND's    CONDUCTIVE LOAD OF WALL (W)            @
C     @         WALLTYPE      WALL TYPE                              @
C     @         WALL_NUM      REFERENCE WALL NUMBER                  @
C     @         WBT's         WET BLUB TEMPERATURE (C) [DAILY]       @
C     @         WBT1's        WET BLUB TEMPERATURE (C)  [WEEKLY]     @
C     @         WD'S          WIND DIRECTION (deg)      [DAILY]      @
C     @         WD1's         WIND DIRECTION (deg)       [WEEKLY]    @
C     @         WIDTH         WIDTH OF SURFACE (m)                   @
C     @         WINCOND's     CONDUCTIVE LOAD OF WINDOW (W)          @
C     @         WINSOL's      SOLAR LOAD OF WINDOW (W)               @
C     @         WIN_NUM       REFERENCE WINDOW NUMBER                @
C     @         WSP's         WIND SPEED (m/s)          [DAILY]      @
C     @         WSP1's        WIND SPEED (m/s)           [WEEKLY]    @
C     @         YR's          YEAR                                   @

      REAL     LLT,SAZ,TILT,LSM,LON,MXDAY,DATE
      REAL     ALT(24),AZM(24),COSINE(24)
      REAL     DBT(24),WBT(24),IDN(24),ID(24),HR(24),WSP(24),WD(24)
      REAL     HO(24)
      INTEGER  DA(7),MO(7),YR(7)
      INTEGER  HO_DD,HO_MM
      REAL     DBT1(7,24),WBT1(7,24)
      REAL     IDN1(7,24),ID1(7,24),HR1(7,24),WSP1(7,24),WD1(7,24)
      REAL     ALPHA
      REAL     B(24)
      REAL     A,U
      INTEGER  NUM,WALL_NUM,WIN_NUM
      REAL     QL1(24),HG(24),HG1(24),SOLTEMP(24),SHGF(24)
      REAL     V_SOLA(3),V_COND(3),V_LIGH(3),W(2)
      REAL     QL(168),V(168),VI(168),TRTS(168)
      REAL     ERMX(168),ERMN(168),TRT(168)
      REAL     ER(168)
      REAL     GG(3),P(2)
      REAL     THTR,TRC,KT,SUM_UA,PERIMETER,FA
      INTEGER  NUM_DAY
```

```
      REAL    SUNLIT_A(24)
      REAL    PX,PY,PZ,WIDTH,HEIGHT
      REAL    RW,RH,FV,PH,FV,FH,RECESS
      INTEGER UNIT_NUM
      REAL    ERR
      REAL    DEG
      REAL    DUMMY,DUMMY1,DUMMY2
      INTEGER DAY_TYPE,DAY_T(7)
      INTEGER NUM_SCHE,NUM_ROOM,NUM_WALL,NUM_WIN
      INTEGER WALLTYPE,ADJAROOM,ROOM_NUM
      REAL    ADJATEMP(24)
      CHARACTER*20 ROOMNAME
      INTEGER NUM_PEO
      REAL    RAD_FAC
      REAL    SEN_FAC,LAT_FAC
      REAL    KILOWATT
      REAL    VENT_RAT,INFI_RAT
      REAL    EQUI_SEN,EQUI_LAT
      REAL    WALLCOND(168),WINCOND(168),WINSOL(168)
      REAL    PEOSEN(168),PEOLAT(168),LIGHT(168)
      REAL    VENTSEN(168),VENTLAT(168),INFISEN(168),INFILAT(168)
      REAL    EQUISEN(168),EQUILAT(168)
      REAL    SCHEDULE(168)
      INTEGER SCH_STRD,SCH_STRM
      INTEGER SCH_STPD,SCH_STPM
      INTEGER SCH_NUM(8),SCH_NUM2(8),SCH_NUM3(8)
      INTEGER END_PER
      INTEGER ITERATION
      INTEGER IDUMMY,IDUMMY2
      INTEGER I,J,K,L

      OPEN(2,FILE=WEATHER)
      OPEN(3,FILE=OUTPUT)
      UNIT_NUM = 4
      OPEN(11,FILE=CALENDAR)
      OPEN(12,FILE='SCHEDULE.TMP',ACCESS='DIRECT',RECL=10)
      OPEN(13,FILE='WALL.TMP'    ,ACCESS='DIRECT',RECL=10)
      OPEN(14,FILE='WIN.TMP'     ,ACCESS='DIRECT',RECL=10)
      OPEN(15,FILE='PARTITIO.TMP',ACCESS='DIRECT',RECL=10)
      OPEN(16,FILE='ROOM.TMP')
      DEG = ATAN(1.0)/45.
      END_PER = 0

C     ----- COPY ROOM NAMES INTO OUTPUT FILE                    -----
      WRITE(3,'(A)') 'BUILDING LOADS'                            *new*
      READ(16,*) NUM_ROOM
      WRITE(3,*) NUM_ROOM
      DO 100 I=1,NUM_ROOM
         READ(16,'(A)') ROOMNAME
         WRITE(3,'(A20)') ROOMNAME
  100 CONTINUE

      DO 101 I=1,NUM_ROOM                                        *new*
         READ(16,*) FA                                           *new*
         WRITE(3,*) FA                                           *new*
  101 CONTINUE                                                   *new*

      CLOSE(16)

C     ----- READ day type OF THE FIRST DAY OF THE YEAR TO MAKE A  -----
C     ----- PROPER CALENDAR, [0:sun], [1:mom], ..., [6:sat].      -----
      READ(11,*,ERR=6280,END=6281) DAY_TYPE

C     ----- SEEK FOR THE FIRST DATE OF THIS CALCULATION          -----
      READ(2,*,ERR=6290,END=6291) LLT,LON,LSM,MXDAY

      DATE = 0.
      DAY_TYPE = DAY_TYPE - 1
  110 READ(2,*,ERR=6292,END=6293)
     +                    DA(1),MO(1),YR(1),DBT,WBT,IDN,ID,HR,WSP,WD
      DATE = DATE + 1./MXDAY
      DAY_TYPE = DAY_TYPE + 1
      IF( DAY_TYPE.EQ.7 ) DAY_TYPE = 0
      IF( STR_DD.NE.DA(1) ) THEN
         GO TO 110
      ELSE IF( STR_MM.NE.MO(1) ) THEN
         GO TO 110
      END IF

C     ----- COPY SOME PARTS FOR WEATHER DATA FOR CALCULATING LOADS-----
      NUM_DAY = 1
      DO 120 I=1,24
         DBT1(1,I) = DBT(I)
         WBT1(1,I) = WBT(I)
         IDN1(1,I) = IDN(I)
         ID1(1,I)  = ID(I)
         HR1(1,I)  = HR(I)
         WSP1(1,I) = WSP(I)
         WD1(1,I)  = WD(I)
  120 CONTINUE

C     ----- ASSIGN DAY_TYPE OF EACH DAYS, WITHIN THE SCHEDULE    -----
  121 READ(11,*,END=122,ERR=122) HO_DD,HO_MM
      GO TO 123

  122 HO_DD = 32
      HO_MM = 13

  123 IF( HO_MM.LT.MO(1) ) THEN
         GO TO 121
      ELSE IF( HO_MM.EQ.MO(1)) THEN
         IF( HO_DD.LT.DA(1) ) GO TO 121
         IF( HO_DD.EQ.DA(1) ) DAY_T(1) = 7
         IF( HO_DD.GT.DA(1) ) DAY_T(1) = DAY_TYPE
```

```
      ELSE
         DAY_T(1) = DAY_TYPE
      END IF

  130 NUM_DAY = NUM_DAY + 1
      IF( NUM_DAY.LE.7 ) THEN
         IF( MODE_CAL.EQ.1 ) GO TO 150
         READ(2,*,ERR=6300,END=150) DA(NUM_DAY),MO(NUM_DAY),YR(NUM_DAY),
     +                              DBT,WBT,IDN,ID,HR,WSP,WD

         DATE = DATE + 1./MXDAY
         DAY_TYPE = DAY_TYPE + 1
         IF( DAY_TYPE.EQ.7 ) DAY_TYPE = 0
         DO 140 I=1,24
            DBT1(NUM_DAY,I) = DBT(I)
            WBT1(NUM_DAY,I) = WBT(I)
            IDN1(NUM_DAY,I) = IDN(I)
            ID1(NUM_DAY,I)  = ID(I)
            HR1(NUM_DAY,I)  = HR(I)
            WSP1(NUM_DAY,I) = WSP(I)
            WD1(NUM_DAY,I)  = WD(I)
  140    CONTINUE

C     ----- ASSIGN DAY_TYPE OF EACH DAYS, WITHIN THE SCHEDULE      -----
         GO TO 144

  141    READ(11,*,ERR=142,END=143) HO_DD,HO_MM
         GO TO 144

  142    CALL WARNNING(1,0,0,0)
  143    HO_DD = 32
         HO_MM = 13

  144    IF( HO_MM.LT.MO(1) ) THEN
            GO TO 141
         ELSE IF( HO_MM.EQ.MO(1)) THEN
            IF( HO_DD.LT.DA(NUM_DAY) ) GO TO 141
            IF( HO_DD.EQ.DA(NUM_DAY) ) DAY_T(NUM_DAY)  = 7
            IF( HO_DD.GT.DA(NUM_DAY) ) DAY_T(NUM_DAY) = DAY_TYPE
         ELSE
            DAY_T(NUM_DAY) = DAY_TYPE
         END IF

         IF( DA(NUM_DAY).EQ.STP_DD ) THEN
            IF( MO(NUM_DAY).EQ.STP_MM ) THEN
               NUM_DAY = NUM_DAY + 1
               GO TO 150
            END IF
         END IF

         GO TO 130
      END IF
      NUM_DAY = 7
      GO TO 160

  150 NUM_DAY = NUM_DAY - 1
      IF( DA(NUM_DAY).NE.STP_DD .OR. MO(NUM_DAY).NE.STP_MM )
     +                                        CALL WARNNING(2,0,0,0)
      END_PER = 1

  160 CONTINUE

      ITERATION = 1

C     --- SHOW THE CURRENT DATE THAT THE PROGRAM'S SYMULATING.    -----
      CALL TAT(5)
      CALL TCL(33,40)
      CALL XY(11,15)
      WRITE(6,'(I2,A3,I2)') DA(NUM_DAY),' | ',MO(NUM_DAY)
      CALL TAT(0)
      CALL TAT(1)
      CALL TCL(36,40)

C     ----- STARTING TO CALCULATE LOADS OF THIS BUILDING           -----
  170 OPEN(7,FILE='BUILDING.TMP')
      IF( (ITERATION/2)*2.NE.ITERATION ) THEN
         OPEN(8,FILE='CONVER1.TMP')
      ELSE
         OPEN(8,FILE='CONVER2.TMP')
      END IF

      ROOM_NUM = 0
C     ----- ACQURE DATA OF THE ROOM (THERMAL PROPERTIES).          -----
  180 ROOM_NUM = ROOM_NUM + 1
      READ(7,*,END=810)  V_SOLA,V_COND,V_LIGH,W,KT,SUM_UA,
     +                   PERIMETER,GG,P,FA
      READ(7,*)  TRC,THTR

C     ----- INQUIRE DATA OF THERMAL CONTROL FROM 'SCHEDULE.TMP'.    -----
      K = 1
      READ(7,*)  NUM_SCHE
      DO 210 J=1,NUM_SCHE
         READ(7,*)  SCH_STRD,SCH_STRM,
     +              SCH_STPD,SCH_STPM,
     +              SCH_NUM,SCH_NUM2,SCH_NUM3
  190    IF(K.LE.NUM_DAY) THEN
            IDUMMY =
     +         WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
         ELSE
            IDUMMY = 1
         END IF

         IF(IDUMMY.EQ.0) THEN
            DO 200 L=1,24
```

```
            READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                              ERR=6310,END=6310) ERMX(K*24+L-24)
            READ(12,REC=(SCH_NUM2(DAY_T(K)+1)*24+L-24),
     +                              ERR=6310,END=6310) ERMN(K*24+L-24)
            READ(12,REC=(SCH_NUM3(DAY_T(K)+1)*24+L-24),
     +                              ERR=6310,END=6310) TRTS(K*24+L-24)
  200       CONTINUE
            K=K+1
            GO TO 190
         ELSE IF( IDUMMY.EQ.-1 ) THEN
            GO TO 6311
         ENDIF

  210 CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6312

C     ----- SET SOME VARIABLES BEING ZERO BEFORE CALCULATING LOAD.-----
      DO 220 I=1,NUM_DAY*24
         WALLCOND(I) = 0.
         WINCOND(I)  = 0.
  220    WINSOL(I)   = 0.

C     ----- COMPUTE LOADS FROM CONDUCTIVE HEATTRANSFER THRU WALLS -----
C     ----- AND WINDOWS AND SOLAR LOAD THRU WINDOWS.              -----
      READ(7,*)  NUM_WALL
      DO 370 I=1,NUM_WALL
         READ(7,*) SAZ,TILT,WALL_NUM,A,
     +             PX,PY,PZ,WIDTH,HEIGHT,WALLTYPE,ADJAROOM,NUM_WIN
         IF( WALLTYPE.EQ.1 .OR. WALLTYPE.EQ.2 ) THEN
            READ(13,REC=(WALL_NUM*27-26),ERR=6320,END=6320) ALPHA
            READ(13,REC=(WALL_NUM*27-25)) U
            READ(13,REC=(WALL_NUM*27-24)) NUM
            DO 225 J=1,24
  225          READ(13,REC=(WALL_NUM*27+J-24)) B(J)
         ELSE IF( WALLTYPE.EQ.3 ) THEN
            READ(15,REC=(ABS(WALL_NUM)*52-51),ERR=6320,END=6320) ALPHA
            READ(15,REC=(ABS(WALL_NUM)*52-50)) U
            IF( WALL_NUM.GT.0 ) THEN
               READ(15,REC=(WALL_NUM*52-49)) NUM
               DO 226 J=1,24
  226             READ(15,REC=(WALL_NUM*52+J-49)) B(J)
            ELSE
               READ(15,REC=(WALL_NUM*(-52)-24)) NUM
               DO 227 J=1,24
  227             READ(15,REC=(WALL_NUM*(-52)+J-24)) B(J)
            END IF
         ELSE
            GO TO 6330
         END IF

         DO 280 J=1,NUM_DAY
            IF( WALLTYPE.EQ.1 .OR. WALLTYPE.EQ.2 ) THEN
               DO 230 K=1,24
                  DBT(K) = DBT1(J,K)
                  IDN(K) = IDN1(J,K)
                  ID(K)  = ID1(J,K)
                  WSP(K) = WSP1(J,K)
                  WD(K)  = WD1(J,K)
  230             CONTINUE
            END IF

            IF( WALLTYPE.EQ.3 ) THEN
               IF( ITERATION.EQ.1 ) THEN
                  DO 231 K=1,24
  231                ADJATEMP(K) = 25.5
               ELSE
                  OPEN(10,FILE='TEMP.TMP',ACCESS='DIRECT',RECL=10)
                  IDUMMY = (ADJAROOM-1)*NUM_DAY*24 + (J-1)*24
                  DO 232 K=1,24
  232                READ(10,REC=(IDUMMY+K)) ADJATEMP(K)
                  CLOSE(10)
               END IF
            END IF

            IF( WALLTYPE.EQ.1 ) THEN
               CALL ANGLE(LLT,SAZ,TILT,LSM,LON,
     +                    (DATE-FLOAT(NUM_DAY-J)/MXDAY),ALT,AZM,COSINE)
               DO 240 K=1,24
                  IF( TILT.EQ.0.0 .OR. TILT.EQ.180.0 ) THEN
                     HO(K) = HO(DBT(K),
     +                         ABS(WSP(K)),WIDTH,ERR)
                     IF( ERR.NE.0 ) CALL WARNNING(3,0,0,0)
                  ELSE
                     HO(K) = HO(DBT(K),
     +                         ABS(WSP(K)*SIN((SAZ-WD(K))*DEG)),WIDTH,ERR)
                     IF( ERR.NE.0 ) CALL WARNNING(3,0,0,0)
                  END IF
  240             CONTINUE
               ERR = 0

               CALL SUNLITAREA(ALT,AZM,COSINE,PX,PY,PZ,WIDTH,HEIGHT,SAZ,
     +                         TILT,0.,0.,0.,0.,0.,0.,0.,
     +                         DE_SHAD,UNIT_NUM,SUNLIT_A,ERR)
               IF( ERR.EQ.1 ) THEN
                  CALL WARNNING(4,I,ROOM_NUM,0)
                  ERR = 0
               ELSE IF( ERR.EQ.2 ) THEN
                  GO TO 6340
               ELSE IF( ERR.EQ.3 ) THEN
                  GO TO 6341
               ELSE IF( ERR.EQ.4 ) THEN
                  GO TO 6342
               END IF
```

```
            DO 250 K=1,24
250            IDN(K) = IDN(K)*SUNLIT_A(K)/HEIGHT/WIDTH

               CALL SOL_AIR(DATE,DBT,IDN,ID,HO,ALT,COSINE,ALPHA,TILT,
     +                     SHGF,SOLTEMP)

               CALL HEATGAIN(B,NUM,SOLTEMP,1.0,U,TRC,HG1)

            ELSE IF( WALLTYPE.EQ.2 ) THEN
               DO 255 K=1,24
                  HO(K) = 8.3
255               IDN(K) = 0.0
               CALL ANGLE(LLT,SAZ,TILT,LSM,LON,
     +                    (DATE-FLOAT(NUM_DAY-J)/MXDAY),ALT,AZM,COSINE)
               CALL SOL_AIR(DATE,DBT,IDN,ID,HO,ALT,COSINE,ALPHA,TILT,
     +                     SHGF,SOLTEMP)
               CALL HEATGAIN(B,NUM,SOLTEMP,1.0,U,TRC,HG1)
            ELSE IF( WALLTYPE.EQ.3 ) THEN
               CALL HEATGAIN(B,NUM,ADJATEMP,1.0,U,TRC,HG1)
            END IF

            DO 260 K=1,24
260            HG(K) = HG1(K)*A

            CALL HEATLOAD(QL1,HG,V_COND,W,KT,ERR)
            IF( ERR.NE.0 ) CALL WARNNING(5,I,ROOM_NUM,0)

            DO 270 K=1,24
               IDUMMY = J*24+K-24
270            WALLCOND(IDUMMY) = WALLCOND(IDUMMY) + QL1(K)

280      CONTINUE

         DO 360 J=1,NUM_WIN
            READ(7,*) WIN_NUM,A,
     +           PX,PY,PZ,WIDTH,HEIGHT,RW,RH,PV,PH,FV,FH,RECESS
            READ(14,REC=(WIN_NUM*28-27),ERR=6321,END=6321) SC
            READ(14,REC=(WIN_NUM*28-26)) ALPHA
            READ(14,REC=(WIN_NUM*28-25)) U
            READ(14,REC=(WIN_NUM*28-24)) NUM
            DO 285 K=1,24
285            READ(14,REC=(WIN_NUM*28+K-24)) B(K)

            DO 350 K=1,NUM_DAY
               IF( WALLTYPE.EQ.1 .OR. WALLTYPE.EQ.2 ) THEN
                  DO 290 L=1,24
                     DBT(L) = DBT1(K,L)
                     IDN(L) = IDN1(K,L)
                     ID(L)  = ID1(K,L)
                     WSP(L) = WSP1(K,L)
                     WD(L)  = WD1(K,L)
290               CONTINUE
               END IF

               IF( WALLTYPE.EQ.3 ) THEN
                  IF( ITERATION.EQ.1 ) THEN
                     DO 291 L=1,24
291                     ADJATEMP(L) = 25.5
                  ELSE
                     OPEN(10,FILE='TEMP.TMP',ACCESS='DIRECT',RECL=10)
                     IDUMMY = (ADJAROOM-1)*NUM_DAY*24+(K-1)*24
                     DO 292 L=1,24
292                     READ(10,REC=(IDUMMY+L)) ADJATEMP(L)
                     CLOSE(10)
                  END IF
               END IF

               IF( WALLTYPE.EQ.1 ) THEN
                  CALL ANGLE(LLT,SAZ,TILT,LSM,LON,
     +                       (DATE-FLOAT(NUM_DAY-J)/MXDAY),ALT,AZM,COSINE)

                  DO 300 L=1,24
                     IF( TILT.EQ.0.0 .OR. TILT.EQ.180.0 ) THEN
                        HO(L) = HO(DBT(L),
     +                            ABS(WSP(L)),WIDTH,ERR)
                     ELSE
                        HO(L) = HO(DBT(L),
     +                       ABS(WSP(L)*SIN((SAZ-WD(L))*DEG)),WIDTH,ERR)
                     END IF
300               CONTINUE

                  CALL SUNLITAREA(ALT,AZM,COSINE,PX,PY,PZ,WIDTH,HEIGHT,
     +                            SAZ,TILT,RW,RH,PV,PH,FV,FH,RECESS,
     +                            DE_SHAD,UNIT_NUM,SUNLIT_A,ERR)
                  IF( ERR.EQ.1 ) THEN
                     CALL WARNNING(6,J,I,ROOM_NUM)
                     ERR = 0
                  ELSE IF( ERR.EQ.2 ) THEN
                     GO TO 6340
                  ELSE IF( ERR.EQ.3 ) THEN
                     GO TO 6341
                  ELSE IF( ERR.EQ.4 ) THEN
                     GO TO 6343
                  END IF

                  DO 304 L=1,24
304               IDN(L) = IDN(L)*SUNLIT_A(L)/HEIGHT/WIDTH

                  CALL SOL_AIR(DATE,DBT,IDN,ID,HO,ALT,COSINE,ALPHA,TILT,
     +                        SHGF,SOLTEMP)

                  DO 305 L=1,24
                     SHGF(L) = SHGF(L)*SC*A
```

```
305             CONTINUE

                CALL HEATLOAD(QL1,SHGF,V_SOLA,W,KT,ERR)
                IF( ERR.NE.0 ) CALL WARNNING(7,J,I,ROOM_NUM)

                DO 310 L=1,24
                   IDUMMY = K*24+L-24
310                WINSOL(IDUMMY) = WINSOL(IDUMMY) + QL1(L)

                   CALL HEATGAIN(B,NUM,SOLTEMP,1.0,U,TRC,HG1)
             ELSE IF( WALLTYPE.EQ.2 ) THEN
                DO 320 L=1,24
                   HO(L) = 8.3
320                IDN(L) = 0.0
                CALL ANGLE(LLT,SAZ,TILT,LSM,LON,
     +                (DATE-FLOAT(NUM_DAY-J)/MXDAY),ALT,AZM,COSINE)
                CALL SOL_AIR(DATE,DBT,IDN,ID,HO,ALT,COSINE,ALPHA,TILT,
     +                     SHGF,SOLTEMP)
                CALL HEATGAIN(B,NUM,DBT,1.0,U,TRC,HG1)
             ELSE IF( WALLTYPE.EQ.3 ) THEN
                CALL HEATGAIN(B,NUM,ADJATEMP,1.0,U,TRC,HG1)
             END IF

             DO 330 L=1,24
330             HG(L) = HG1(L)*A

             CALL HEATLOAD(QL1,HG,V_COND,W,KT,ERR)
             IF( ERR.NE.0 ) CALL WARNNING(7,J,I,ROOM_NUM)

             DO 340 L=1,24
                IDUMMY = K*24+L-24
340             WINCOND(IDUMMY) = WINCOND(IDUMMY) + QL1(L)

350       CONTINUE
360    CONTINUE
370 CONTINUE

C     ----- CALCULATE OCCUPANCY LOADS [SENSIBLE & LATENT LOAD].   -----
      READ(7,*)  NUM_PEO,SEN_FAC,RAD_FAC,LAT_FAC,NUM_SCHE

C     ----- INQUIRE DATA OF OCCUPANCY SCHED. FROM 'SCHEDULE.TMP'. -----
      K = 1
      DO 400 J=1,NUM_SCHE
         READ(7,*)  SCH_STRD,SCH_STRM,
     +              SCH_STPD,SCH_STPM,
     +              SCH_NUM
380      IF(K.LE.NUM_DAY) THEN
            IDUMMY =
     +         WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
         ELSE
            IDUMMY = 1
         END IF

         IF(IDUMMY.EQ.0) THEN
            DO 390 L=1,24
390            READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                        ERR=6350,END=6350) SCHEDULE(K*24+L-24)
            K=K+1
            GO TO 380
         ELSE IF( IDUMMY.EQ.-1 ) THEN
            GO TO 6351
         ENDIF

400   CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6352

      DO 430 I=1,NUM_DAY
         DO 410 J=1,24
410         HG(J) = SCHEDULE(I*24+J-24)*NUM_PEO*SEN_FAC*RAD_FAC

         CALL HEATLOAD(QL1,HG,V_SOLA,W,KT,ERR)
         IF( ERR.NE.0 ) CALL WARNNING(8,ROOM_NUM,0,0)
         DO 420 J=1,24
            IDUMMY = I*24+J-24
            PEOSEN(IDUMMY) =
     +         SCHEDULE(IDUMMY)*NUM_PEO*SEN_FAC*(1.0-RAD_FAC) + QL1(J)
420         PEOLAT(IDUMMY) = SCHEDULE(IDUMMY)*NUM_PEO*LAT_FAC
430   CONTINUE

C     ----- CALCULATE LIGHTING LOAD.                              -----
      READ(7,*)  KILOWATT,RAD_FAC,NUM_SCHE

C     ----- INQUIRE DATA OF LIGHTING SCHED.  FROM 'SCHEDULE.TMP'. -----
      K = 1
      DO 460 J=1,NUM_SCHE
         READ(7,*)  SCH_STRD,SCH_STRM,
     +              SCH_STPD,SCH_STPM,
     +              SCH_NUM
440      IF(K.LE.NUM_DAY) THEN
            IDUMMY =
     +         WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
         ELSE
            IDUMMY = 1
         END IF

         IF(IDUMMY.EQ.0) THEN
            DO 450 L=1,24
450            READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                        ERR=6360,END=6360) SCHEDULE(K*24+L-24)
            K=K+1
            GO TO 440
         ELSE IF( IDUMMY.EQ.-1 ) THEN
            GO TO 6361
```

```
         ENDIF

  460 CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6362

      DO 490 I=1,NUM_DAY
         DO 470 J=1,24
  470       HG(J) = SCHEDULE(I*24+J-24)*KILOWATT*1000.*RAD_FAC

         CALL HEATLOAD(QL1,HG,V_LIGH,W,KT,ERR)
         IF( ERR.NE.0 ) CALL WARNNING(9,ROOM_NUM,0,0)
         DO 480 J=1,24
            IDUMMY = I*24+J-24
  480       LIGHT(IDUMMY) =
     +        SCHEDULE(IDUMMY)*KILOWATT*1000.*(1.0-RAD_FAC) + QL1(J)
  490 CONTINUE

C     ----- CALCULATE VENTILATION LOADS (SENSIBLE & LATENT LOAD). -----
      READ(7,*)  VENT_RAT,NUM_SCHE

C     ----- INQUIRE DATA OF VENTILATION SCH. FROM 'SCHEDULE.TMP'. -----
      K = 1
      DO 520 J=1,NUM_SCHE
         READ(7,*)  SCH_STRD,SCH_STRM,
     +              SCH_STPD,SCH_STPM,
     +              SCH_NUM
  500    IF(K.LE.NUM_DAY) THEN
            IDUMMY =
     +        WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
         ELSE
            IDUMMY = 1
         END IF

         IF(IDUMMY.EQ.0) THEN
            DO 510 L=1,24
  510          READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                        ERR=6370,END=6370) SCHEDULE(K*24+L-24)
            K=K+1
            GO TO 500
         ELSE IF( IDUMMY.EQ.-1 ) THEN
            GO TO 6371
         ENDIF

  520 CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6372

      DO 550 I=1,NUM_DAY
         DO 540 J=1,24
            IDUMMY = I*24+J-24
            V(IDUMMY) = SCHEDULE(IDUMMY)*VENT_RAT
            DUMMY = DBT1(I,J) - TRC
            VENTSEN(IDUMMY) = 1230.*V(IDUMMY)*DUMMY
            DUMMY = HR1(I,J) - 0.01
  540       VENTLAT(IDUMMY) = 3010.E+3*V(IDUMMY)*DUMMY
  550 CONTINUE

C     ----- CALCULATE INFILTRATION LOADS (SENSIBLE & LATENT LOAD).-----
      READ(7,*)  INFI_RAT,NUM_SCHE

C     ----- INQUIRE DATA OF INFILTRATION SCH. FROM 'SCHEDULE.TMP'.-----
      K = 1
      DO 580 J=1,NUM_SCHE
         READ(7,*)  SCH_STRD,SCH_STRM,
     +              SCH_STPD,SCH_STPM,
     +              SCH_NUM
  560    IF(K.LE.NUM_DAY) THEN
            IDUMMY =
     +        WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
         ELSE
            IDUMMY = 1
         END IF

         IF(IDUMMY.EQ.0) THEN
            DO 570 L=1,24
  570          READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                        ERR=6380,END=6380) SCHEDULE(K*24+L-24)
            K=K+1
            GO TO 560
         ELSE IF( IDUMMY.EQ.-1 ) THEN
            GO TO 6381
         ENDIF

  580 CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6382

      DO 610 I=1,NUM_DAY
         DO 600 J=1,24
            IDUMMY = I*24+J-24
            VI(IDUMMY) = SCHEDULE(IDUMMY)*INFI_RAT
            DUMMY = DBT1(I,J) - TRC
            INFISEN(IDUMMY) = 1230.*VI(IDUMMY)*DUMMY
            DUMMY = HR1(I,J) - 0.01
  600       INFILAT(IDUMMY) = 3010.E+3*VI(IDUMMY)*DUMMY
  610 CONTINUE

C     ----- CALCULATE EQUIPMENT LOADS (SENSIBLE & LATENT LOAD).   -----
      READ(7,*)  EQUI_SEN,RAD_FAC,EQUI_LAT,NUM_SCHE

C     ----- INQUIRE DATA OF EQUIPMENT SCHED. FROM 'SCHEDULE.TMP'. -----
      K = 1
      DO 640 J=1,NUM_SCHE
```

```
      READ(7,*)  SCH_STRD,SCH_STRM,
     +           SCH_STPD,SCH_STPM,
     +           SCH_NUM
620   IF(K.LE.NUM_DAY) THEN
         IDUMMY =
     +      WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA(K),MO(K))
      ELSE
         IDUMMY = 1
      END IF

      IF(IDUMMY.EQ.0) THEN
         DO 630 L=1,24
630         READ(12,REC=(SCH_NUM(DAY_T(K)+1)*24+L-24),
     +                  ERR=6390,END=6390) SCHEDULE(K*24+L-24)
         K=K+1
         GO TO 620
      ELSE IF( IDUMMY.EQ.-1 ) THEN
         GO TO 6391
      ENDIF

640   CONTINUE

      IF( K.LE.NUM_DAY ) GO TO 6392

      DO 670 I=1,NUM_DAY
         DO 650 J=1,24
650         HG(J) = SCHEDULE(I*24+J-24)*EQUI_SEN*RAD_FAC

         CALL HEATLOAD(QL1,HG,V_SOLA,W,KT,ERR)
         IF( ERR.NE.0 ) CALL WARNNING(10,ROOM_NUM,0,0)
         DO 660 J=1,24
            IDUMMY = I*24+J-24
            EQUISEN(IDUMMY) =
     +           SCHEDULE(IDUMMY)*EQUI_SEN*(1.0-RAD_FAC) + QL1(J)
660         EQUILAT(IDUMMY) = SCHEDULE(IDUMMY)*EQUI_LAT
670   CONTINUE

C     ----- SUM ALL COMPONENT LOADS INTO TOTAL COOLING LOAD       -----
C     ----- OF THIS ROOM.                                        -----
      IDUMMY = 24*NUM_DAY
      DO 675 I=1,IDUMMY
         QL(I) = WALLCOND(I)
         QL(I) = QL(I) + WINCOND(I)
         QL(I) = QL(I) + WINSOL(I)
         QL(I) = QL(I) + PEOSEN(I)
         QL(I) = QL(I) + PEOLAT(I)
         QL(I) = QL(I) + LIGHT(I)
         QL(I) = QL(I) + VENTSEN(I)
         QL(I) = QL(I) + VENTLAT(I)
         QL(I) = QL(I) + INFISEN(I)
         QL(I) = QL(I) + INFILAT(I)
         QL(I) = QL(I) + EQUISEN(I)
675      QL(I) = QL(I) + EQUILAT(I)

C     ----- COMPUTE HEATEXTRACTION & ROOM TEMPERATURE             -----
      CALL HEATEXTC(QL,SUM_UA,V,VI,TRTS,THTR,TRC,GG,P,FA,
     +                                  ERMX,ERMN,TRT,ER,NUM_DAY,ERR)
      IF( ERR.NE.0 ) CALL WARNNING(11,ROOM_NUM,0,0)

C     ----- COLLECT ALL COMPONENTS OF THE LOAD OF THIS ROOM AND   -----
      DO 795 I=1,NUM_DAY
         IDUMMY = (I-1)*24

         OPEN(17,FILE='WALL_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*24 + (ROOM_NUM-1)*NUM_DAY*24
         DO 680 J=1,24
680         WRITE(17,REC=(IDUMMY2+J)) WALLCOND(IDUMMY+J)
         CLOSE(17)

         OPEN(18,FILE='WIN_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
         DO 690 J=1,24
690         WRITE(18,REC=(IDUMMY2+J*2-1)) WINCOND(IDUMMY+J)
         DO 700 J=1,24
700         WRITE(18,REC=(IDUMMY2+J*2  )) WINSOL(IDUMMY+J)
         CLOSE(18)

         OPEN(19,FILE='PEO_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
         DO 710 J=1,24
710         WRITE(19,REC=(IDUMMY2+J*2-1)) PEOSEN(IDUMMY+J)
         DO 720 J=1,24
720         WRITE(19,REC=(IDUMMY2+J*2  )) PEOLAT(IDUMMY+J)
         CLOSE(19)

         OPEN(20,FILE='LIGH_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*24 + (ROOM_NUM-1)*NUM_DAY*24
         DO 730 J=1,24
730         WRITE(20,REC=(IDUMMY2+J)) LIGHT(IDUMMY+J)
         CLOSE(20)

         OPEN(21,FILE='VENT_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
         DO 740 J=1,24
740         WRITE(21,REC=(IDUMMY2+J*2-1)) VENTSEN(IDUMMY+J)
         DO 750 J=1,24
750         WRITE(21,REC=(IDUMMY2+J*2  )) VENTLAT(IDUMMY+J)
         CLOSE(21)

         OPEN(22,FILE='INFI_OUT.TMP',ACCESS='DIRECT',RECL=10)
         IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
         DO 760 J=1,24
760         WRITE(22,REC=(IDUMMY2+J*2-1)) INFISEN(IDUMMY+J)
         DO 770 J=1,24
```

```
  770       WRITE(22,REC=(IDUMMY2+J*2  )) INFILAT(IDUMMY+J)
            CLOSE(22)

            OPEN(23,FILE='EQUI_OUT.TMP',ACCESS='DIRECT',RECL=10)
            IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
            DO 780 J=1,24
  780       WRITE(23,REC=(IDUMMY2+J*2-1)) EQUISEN(IDUMMY+J)
            DO 790 J=1,24
  790       WRITE(23,REC=(IDUMMY2+J*2  )) EQUILAT(IDUMMY+J)
            CLOSE(23)

            OPEN(9,FILE='ER_OUT.TMP',ACCESS='DIRECT',RECL=10)
            IDUMMY2 = (I-1)*48 + (ROOM_NUM-1)*NUM_DAY*48
            DO 791 J=1,24
  791       WRITE(9,REC=(IDUMMY2+J*2-1)) TRT(IDUMMY+J)
            DO 793 J=1,24
  793       WRITE(9,REC=(IDUMMY2+J*2  )) ER(IDUMMY+J)
  795 CONTINUE

C     ----- COLLECT ROOM TEMPERATURE AT THE LAST HOUR OF EACH DAY -----
C     ----- INTO FILE 'CONVER_.TMP TO CHECK IF THE COPUTATION     -----
C     ----- CONVERSES.                                            -----
      DO 800 I=1,NUM_DAY
  800    WRITE(8,*) TRT(24*I)

C     ----- COLLECT ROOM TEMPERATURE INTO FILE 'TEMP.TMP'.        -----
      OPEN(10,FILE='TEMP.TMP',ACCESS='DIRECT',RECL=10)
      DO 805 I=1,24*NUM_DAY
         IDUMMY = (ROOM_NUM-1)*NUM_DAY*24
  805    WRITE(10,REC=(IDUMMY+I)) TRT(I)
      CLOSE(10)

C     ----- RETURN TO COMPUTE LOAD OF ANOTHER ROOM UNTIL ALL ROOMS-----
C     ----- ARE COMPUTED. THEN PASS THRU TO THE LINE 810.         -----
      GO TO 180

  810 CONTINUE

C     ----- CLOSE FILE BUILDING.TMP, CONVER_.TMP AND COMPO.TMP.   -----
      CLOSE(7)
      CLOSE(8)

C     ----- CHECK WHETHER OR NOT THE COMPUTATION CONVERSES BY      -----
C     ----- CHECKING THE CONVERSING OF ROOM TEMPERATURES THOSE     -----
C     ----- WERE COLLECTED IN FILE CONVER_.TMP. (IN CASE THAT      -----
C     ----- IT IS THE FIRST INTERATION OF COMPUTATION, PASS THIS   -----
C     ----- MODULE).                                              -----
      IF( ITERATION.EQ.1 ) THEN
         ITERATION = ITERATION + 1
         GO TO 170
      ELSE
         OPEN(7,FILE='CONVER1.TMP')
         OPEN(8,FILE='CONVER2.TMP')
         DUMMY2 = 0.
         IDUMMY = 0
  820    READ(7,*,END=830) DUMMY
         READ(8,*) DUMMY1
         DUMMY2 = DUMMY2 + ABS(DUMMY-DUMMY1)
         IDUMMY = IDUMMY + 1

         IF( ABS(DUMMY1).GT.1000. .OR. ABS(DUMMY).GT.1000 ) GO TO 6400

         GO TO 820
  830    CLOSE(7)
         CLOSE(8)
         DUMMY = DUMMY2/FLOAT(IDUMMY)

         IF( DUMMY.GT.0.05 .AND. ITERATION.LT.50 ) THEN
            ITERATION = ITERATION + 1
            GO TO 170
         ELSE
            IF( ITERATION.LT.50 ) THEN
               DO 850 I=1,NUM_DAY
                  WRITE(3,*) DA(I),MO(I),YR(I)
                  DO 850 J=1,NUM_ROOM

                   OPEN(17,FILE='WALL_OUT.TMP',ACCESS='DIRECT',RECL=10)
                    IDUMMY= (I-1)*24 + (J-1)*NUM_DAY*24
                    DO 832 L=1,24
                       READ(17,REC=(IDUMMY+L)) DUMMY
  832                  WRITE(3,*) DUMMY
                   CLOSE(17)

                   OPEN(18,FILE='WIN_OUT.TMP',ACCESS='DIRECT',RECL=10)
                    IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                    DO 834 L=1,24
                       READ(18,REC=(IDUMMY+L*2-1)) DUMMY
  834                  WRITE(3,*) DUMMY
                    DO 835 L=1,24
                       READ(18,REC=(IDUMMY+L*2  )) DUMMY
  835                  WRITE(3,*) DUMMY
                   CLOSE(18)

                   OPEN(19,FILE='PEO_OUT.TMP',ACCESS='DIRECT',RECL=10)
                    IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                    DO 837 L=1,24
                       READ(19,REC=(IDUMMY+L*2-1)) DUMMY
  837                  WRITE(3,*) DUMMY
                    DO 838 L=1,24
                       READ(19,REC=(IDUMMY+L*2  )) DUMMY
  838                  WRITE(3,*) DUMMY
                   CLOSE(19)

                   OPEN(20,FILE='LIGH_OUT.TMP',ACCESS='DIRECT',RECL=10)
```

```
                    IDUMMY= (I-1)*24 + (J-1)*NUM_DAY*24
                    DO 840 L=1,24
                       READ(20,REC=(IDUMMY+L)) DUMMY
 840                   WRITE(3,*) DUMMY
                    CLOSE(20)

                    OPEN(21,FILE='VENT_OUT.TMP',ACCESS='DIRECT',RECL=10)
                     IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                     DO 841 L=1,24
                        READ(21,REC=(IDUMMY+L*2-1)) DUMMY
 841                    WRITE(3,*) DUMMY
                     DO 842 L=1,24
                        READ(21,REC=(IDUMMY+L*2  )) DUMMY
 842                    WRITE(3,*) DUMMY
                    CLOSE(21)

                    OPEN(22,FILE='INFI_OUT.TMP',ACCESS='DIRECT',RECL=10)
                     IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                     DO 843 L=1,24
                        READ(22,REC=(IDUMMY+L*2-1)) DUMMY
 843                    WRITE(3,*) DUMMY
                     DO 844 L=1,24
                        READ(22,REC=(IDUMMY+L*2  )) DUMMY
 844                    WRITE(3,*) DUMMY
                    CLOSE(22)

                    OPEN(23,FILE='EQUI_OUT.TMP',ACCESS='DIRECT',RECL=10)
                     IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                     DO 845 L=1,24
                        READ(23,REC=(IDUMMY+L*2-1)) DUMMY
 845                    WRITE(3,*) DUMMY
                     DO 846 L=1,24
                        READ(23,REC=(IDUMMY+L*2  )) DUMMY
 846                    WRITE(3,*) DUMMY
                    CLOSE(23)

                    OPEN(9,FILE='ER_OUT.TMP',ACCESS='DIRECT',RECL=10)
                     IDUMMY= (I-1)*48 + (J-1)*NUM_DAY*48
                     DO 848 L=1,24
                        READ(9,REC=(IDUMMY+L*2-1)) DUMMY
 848                    WRITE(3,*) DUMMY
                     DO 849 L=1,24
                        READ(9,REC=(IDUMMY+L*2  )) DUMMY
 849                    WRITE(3,*) DUMMY
                    CLOSE(9)
 850             CONTINUE
              ELSE
                 GO TO 6410
              END IF
           ENDIF
        ENDIF

C       ----- IF IT IS'N THE END OF THE PERIOD, GO ABOVE TO CONTINUE-----
C       ----- THE COMPUTATION.                                        -----
        NUM_DAY = 0
        IF( END_PER.NE.1 ) GO TO 130

        CLOSE(2)

        STOP

C       ------------------------ ERROR JUNCTION --------------------------
 6280 CALL ERROR(36,0,0,0)
 6281 CALL ERROR(37,0,0,0)
 6290 CALL ERROR(38,0,0,0)
 6291 CALL ERROR(39,0,0,0)
 6292 CALL ERROR(40,DA(1),MO(1),0)
 6293 CALL ERROR(41,0,0,0)
 6300 CALL ERROR(42,DA(NUM_DAY-1),MO(NUM_DAY-1),0)
 6310 CALL ERROR(43,DA(K),MO(K),ROOM_NUM)
 6311 CALL ERROR(44,DA(K),MO(K),ROOM_NUM)
 6312 CALL ERROR(45,DA(K),MO(K),ROOM_NUM)
 6320 CALL ERROR(46,I,ROOM_NUM,0)
 6321 CALL ERROR(47,J,I,ROOM_NUM)
 6330 CALL ERROR(48,I,ROOM_NUM,0)
 6340 CALL ERROR(49,0,0,0)
 6341 CALL ERROR(50,0,0,0)
 6342 CALL ERROR(51,I,ROOM_NUM,0)
 6343 CALL ERROR(52,J,I,ROOM_NUM)
 6350 CALL ERROR(53,DA(K),MO(K),ROOM_NUM)
 6351 CALL ERROR(54,DA(K),MO(K),ROOM_NUM)
 6352 CALL ERROR(55,DA(K),MO(K),ROOM_NUM)
 6360 CALL ERROR(56,DA(K),MO(K),ROOM_NUM)
 6361 CALL ERROR(57,DA(K),MO(K),ROOM_NUM)
 6362 CALL ERROR(57,DA(K),MO(K),ROOM_NUM)
 6370 CALL ERROR(59,DA(K),MO(K),ROOM_NUM)
 6371 CALL ERROR(60,DA(K),MO(K),ROOM_NUM)
 6372 CALL ERROR(61,DA(K),MO(K),ROOM_NUM)
 6380 CALL ERROR(62,DA(K),MO(K),ROOM_NUM)
 6381 CALL ERROR(63,DA(K),MO(K),ROOM_NUM)
 6382 CALL ERROR(64,DA(K),MO(K),ROOM_NUM)
 6390 CALL ERROR(65,DA(K),MO(K),ROOM_NUM)
 6391 CALL ERROR(66,DA(K),MO(K),ROOM_NUM)
 6392 CALL ERROR(67,DA(K),MO(K),ROOM_NUM)
 6400 CALL ERROR(68,0,0,0)
 6410 CALL ERROR(69,0,0,0)

        END

        SUBROUTINE RE_ARR(CP,KK,DEN,THICKNESS,NUM)
        REAL CP(*),KK(*),DEN(*),THICKNESS(*)
        INTEGER NUM
C       ================================================================
C       |                                                              |
```

```
C     |   SUBROUTINE TO REARRENGE THE ORDER OF LAYERS IN PARTITIONS   |
C     |   SO THAT WE WILL DETERMINE HOW HEAT FLUX TRANSFERS THROUGHT  |
C     |   THESE PARTITIONS IN OPPOSITE DIRECTION.                     |
C     |                                                               |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                         @
C     @         DUMMY'S          DUMMY VARIABLES                    @
C     @         I                NUMBER OF LAYER                    @
C     @         J                COUNTER                            @

      REAL    DUMMY(8)
      INTEGER I,J

      I = 0
   10 I = I + 1
      IF( I.GT.NUM ) GO TO 20
      IF( KK(I).NE.0.0 ) GO TO 10

   20 I = I - 1

      DO 30 J=1,I
   30 DUMMY(J) = CP(I-J+1)
      DO 40 J=1,I
   40 CP(J) = DUMMY(J)

      DO 50 J=1,I
   50 DUMMY(J) = KK(I-J+1)
      DO 60 J=1,I
   60 KK(J) = DUMMY(J)

      DO 70 J=1,I
   70 DUMMY(J) = DEN(I-J+1)
      DO 80 J=1,I
   80 DEN(J) = DUMMY(J)

      DO 90 J=1,I
   90 DUMMY(J) = THICKNESS(I-J+1)
      DO 100 J=1,I
  100 THICKNESS(J) = DUMMY(J)

      RETURN
      END

      FUNCTION WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA,MO)
      INTEGER SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA,MO
C     ================================================================
C     |                                                               |
C     |    FUNCTION FOR INDICATING IF THE DATE DA/MO IS WITHIN THE    |
C     |   PERIOD LIMITED BY SCH_STRD/SCH_STRM AND SCH_STPD/SCH_STPM.  |
C     |                                                               |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                         @
C     @         DATE             BRIEFLY CALCULATED DAY No. OF DA/MO  @
C     @         UPPER_L          BRIEFLY CALCULATED DAY No. OF SCH_STP @
C     @         LOWER_L          BRIEFLY CALCULATED DAY No. OF SCH_STR @

      INTEGER DATE,UPPER_L,LOWER_L

C     ----- BY ESTIMATING, 1 MONTH EQUALS 30 DAYS, WILL OBTAIN.   -----
      DATE    = DA       + MO*30
      UPPER_L = SCH_STPD + SCH_STPM*30
      LOWER_L = SCH_STRD + SCH_STRM*30

C     ----- WITHIN = -1,0 or 1, IF DATE IS LOWER, WITHIN or ABOVE -----
C     ----- THE CONSTRAINT, SPECIFIED BY UPPER_L & LOWER_L, RES. .-----
      IF( DATE.LT.LOWER_L ) THEN
         WITHIN = -1
      ELSE IF( DATE.GT.UPPER_L ) THEN
         WITHIN = 1
      ELSE
         WITHIN = 0
      END IF

      RETURN
      END

      SUBROUTINE WARNNING(IA,IB,IC,ID)
      INTEGER IA,IB,IC,ID
C     ================================================================
C     |                                                               |
C     |    THIS SUBROUTINE WILL WARN USERS ABOUT IMPROPER MATTERS     |
C     |    IN THE PROGRAM.                                            |
C     |                                                               |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                         @
C     @         INDENT           INDENT OF PRINTED LINES            @
C     @         LINE             THE LINE OF THE FIRST SENTENCE     @

      INTEGER LINE,INDENT

      LINE = 17
      INDENT = 8

      CALL TAT(0)
      CALL TCL(32,47)
      CALL XY(LINE,INDENT)

      IF( IA.EQ.1 ) THEN
         WRITE(6,*) 'Cannot read DATA of HOLIDAY in CARENDER FILE.'

      ELSE IF( IA.EQ.2 ) THEN
```

```
      WRITE(6,*) 'WEATHER DATA dosen''t cover the COMPUTING PERIOD.'

   ELSE IF( IA.EQ.3 ) THEN
      WRITE(6,*) 'CONVECTIVE HEAT TRANSFER COEFFICIENT of air film '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'is out of range. [AUTOMATICALLY EXTRAPOLATE]'

   ELSE IF( IA.EQ.4 ) THEN
      WRITE(6,*) 'GEOMETRY of WALL#',IB,' in ROOM#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'is invalid.'

   ELSE IF( IA.EQ.5 ) THEN
      WRITE(6,*) 'COMPUTING COOLING LOAD of WALL#',IB,' in ROOM#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'diverges.'

   ELSE IF( IA.EQ.6 ) THEN
      WRITE(6,*) 'GEOMETRY of WINDOW#',IB,' on WALL#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) ' in ROOM#',ID,'is invalid.'

   ELSE IF( IA.EQ.7 ) THEN
      WRITE(6,*) 'COMPUTING COOLING LOAD of WINDOW#',IB
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on WALL#',IC,'in ROOM#',ID,' diverges.'

   ELSE IF( IA.EQ.8 ) THEN
      WRITE(6,*) 'COMPUTING COOLING LOAD of OCCUPANCY in ROOM#',IB
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'diverges.'

   ELSE IF( IA.EQ.9 ) THEN
      WRITE(6,*) 'COMPUTING COOLING LOAD of LIGHTING in ROOM#',IB
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'diverges.'

   ELSE IF( IA.EQ.10 ) THEN
      WRITE(6,*) 'COMPUTING COOLING LOAD of EQUIPMENT in ROOM#',IB
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'diverges.'

   ELSE IF( IA.EQ.11 ) THEN
      WRITE(6,*) 'COMPUTING HEAT EXTRACTION of ROOM#',IB,' diverges.'

   END IF

   RETURN
   END

   SUBROUTINE ERROR(IA,IB,IC,ID)
   INTEGER IA,IB,IC,ID
C  ================================================================
C  |                                                              |
C  |       THIS SUBROUTINE WILL SHOW HOW ERROR OCCURS IN THE      |
C  |    PROGRAM THEN STOP THE PROGRAM FOR AVOIDING ANY SERIOUS    |
C  |    EFFECT, RESULTS FROM THAT ERROR.                          |
C  |                                                              |
C  ================================================================

C  @ ---- ASSIGN VARIABLES                                        @
C  @         INDENT          INDENT OF PRINTED LINES            @
C  @         LINE            THE LINE OF THE FIRST SENTENCE     @

   INTEGER LINE,INDENT

   LINE = 17
   INDENT = 8

   CALL TAT(0)
   CALL TCL(31,47)
   CALL XY(LINE,INDENT)

   IF( IA.EQ.1 ) THEN
      WRITE(6,*) 'Cannot read MODE of CALCULATION from the input',
  +              ' file.'

   ELSE IF( IA.EQ.2 ) THEN
      WRITE(6,*) 'The input file is empty.'

   ELSE IF( IA.EQ.3 ) THEN
      WRITE(6,*) 'Cannot read the starting date or the last date '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'of computing period.'

   ELSE IF( IA.EQ.4 ) THEN
      WRITE(6,*) 'Cannot read NUMBER of REFERENCE WALL from the ',
  +              'input file.'

   ELSE IF( IA.EQ.5 ) THEN
      WRITE(6,*) 'Cannot read COMPONENTS of REFERENCE WALL#', IB

   ELSE IF( IA.EQ.6 ) THEN
      WRITE(6,*) 'NO LAYER ::- in the REFERENCE WALL#', IB

   ELSE IF( IA.EQ.7 ) THEN
      WRITE(6,*) 'Cannot read NUMBER of REFERENCE WINDOW from the',
  +              'input file.'

   ELSE IF( IA.EQ.8 ) THEN
      WRITE(6,*) 'Cannot read CONPONENTs of REFERENCE WINDOW#', IB

   ELSE IF( IA.EQ.9 ) THEN
      WRITE(6,*) 'NO LAYER ::- in the REFERENCE WINDOW#', IB
```

```
      ELSE IF( IA.EQ.10 ) THEN
         WRITE(6,*) 'Cannot read NUMBER of REFERENCE PARTITION from ',
     +              'the input file.'

      ELSE IF( IA.EQ.11 ) THEN
         WRITE(6,*) 'Cannot read COMPONENTS of REFERENCE PARTITION#',IB

      ELSE IF( IA.EQ.12 ) THEN
         WRITE(6,*) 'NO LAYER ::- in the REFERENCE PARTITION#', IB

      ELSE IF( IA.EQ.13 ) THEN
         WRITE(6,*) 'Cannot read NUMBER of REFERENCE SCHEDULE from ',
     +              'the input file.'

      ELSE IF( IA.EQ.14 ) THEN
         WRITE(6,*) 'Cannot read DATA of REFERENCE SCHEDULE from the ',
     +              'input file.'

      ELSE IF( IA.EQ.15 ) THEN
         WRITE(6,*) 'Cannot read NUMBER of ROOM in this building.'

      ELSE IF( IA.EQ.16 ) THEN
         WRITE(6,*) 'Some problems with the NAME of ROOM#', IB

      ELSE IF( IA.EQ.17 ) THEN
         WRITE(6,*) 'Cannot read THERMAL PROPERTIES of ROOM#', IB

      ELSE IF( IA.EQ.18 ) THEN
         WRITE(6,*) 'Cannot read NUMBER of SCHEDULE, used in ROOM#', IB

      ELSE IF( IA.EQ.19 ) THEN
         WRITE(6,*) 'Cannot read DATA of SCHEDULE, used in ROOM#', IB

      ELSE IF( IA.EQ.20 ) THEN
         WRITE(6,*) 'Cannot read NUMBER of WALL in ROOM#', IB

      ELSE IF( IA.EQ.21 ) THEN
         WRITE(6,*) 'Cannot read PROPERTIES of WALL#',IB,' in ROOM#',IC

      ELSE IF( IA.EQ.22 ) THEN
         WRITE(6,*) 'Either HEIGHT or WIDTH of WALL#',IB
         CALL XY(LINE+1,INDENT)
         WRITE(6,*) 'in ROOM#',IC,' equals to zero.'

      ELSE IF( IA.EQ.23 ) THEN
         WRITE(6,*) 'WALL#',IB,' in ROOM#',IC,' is adjacent to'
         CALL XY(LINE+1,INDENT)
         WRITE(6,*) 'a room that has reference No. less than 1.'

      ELSE IF( IA.EQ.24 ) THEN
         WRITE(6,*) 'WALL#',IB,' in ROOM#',IC,' is adjacent to'
         CALL XY(LINE+1,INDENT)
         WRITE(6,*) 'a room that is previously not specified.'

      ELSE IF( IA.EQ.25 ) THEN
         WRITE(6,*) 'Cannot read PROPERTIES of WINDOW#',IB
         CALL XY(LINE+1,INDENT)
         WRITE(6,*) 'on WALL#',IC,' in ROOM#',ID

      ELSE IF( IA.EQ.26 ) THEN
         WRITE(6,*) 'Cannot read DATA of OCCUPANT LOAD in ROOM#',IB

      ELSE IF( IA.EQ.27 ) THEN
         WRITE(6,*) 'Cannot read SCHEDULE of OCCUPANT LOAD in ROOM#',IB

      ELSE IF( IA.EQ.28 ) THEN
         WRITE(6,*) 'Cannot read DATA of LIGHTING LOAD in ROOM#',IB

      ELSE IF( IA.EQ.29 ) THEN
         WRITE(6,*) 'Cannot read SCHEDULE of LIGHTING LOAD in ROOM#',IB

      ELSE IF( IA.EQ.30 ) THEN
         WRITE(6,*) 'Cannot read DATA of VENTILATION LOAD in ROOM#',IB

      ELSE IF( IA.EQ.31 ) THEN
         WRITE(6,*) 'Cannot read SCHEDULE of VENTILATION LOAD in ROOM#'
     +             ,IB

      ELSE IF( IA.EQ.32 ) THEN
         WRITE(6,*) 'Cannot read DATA of INFILTRATION LOAD in ROOM#',IB

      ELSE IF( IA.EQ.33 ) THEN
         WRITE(6,*) 'Cannot read SCHEDULE of INFILTRATION LOAD in ROOM#'
     +             ,IB

      ELSE IF( IA.EQ.34 ) THEN
         WRITE(6,*) 'Cannot read DATA of EQUIPMENT LOAD in ROOM#',IB

      ELSE IF( IA.EQ.35 ) THEN
         WRITE(6,*) 'Cannot read SCHEDULE of EQUIPMENT LOAD in ROOM#',IB

      ELSE IF( IA.EQ.36 ) THEN
         WRITE(6,*) 'Cannot read DAY TYPE (sun,mon,...) of the first '
         CALL XY(LINE+1,INDENT)
         WRITE(6,*) 'day in CANLENDER FILE.'

      ELSE IF( IA.EQ.37 ) THEN
         WRITE(6,*) 'CANLENDER FILE is empty.'

      ELSE IF( IA.EQ.38 ) THEN
         WRITE(6,*) 'Cannot read LOCATION of the WEATHER DATA FILE.'

      ELSE IF( IA.EQ.39 ) THEN
```

```
      WRITE(6,*) 'WEATHER DATA FILE is empty.'

    ELSE IF( IA.EQ.40 ) THEN
      WRITE(6,*) 'Cannot read WEATHER DATA after the date [dd/mm]'
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) '              ',IB,IC

    ELSE IF( IA.EQ.41 ) THEN
      WRITE(6,*) 'WEATHER DATA FILE is empty.'

    ELSE IF( IA.EQ.42 ) THEN
      WRITE(6,*) 'Cannot read WEATHER DATA after the date [dd/mm]'
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) '              ',IB,'/',IC

    ELSE IF( IA.EQ.43 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of SYSTEM CONTROL in ROOM#'
     +          ,ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.44 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified SYSTEM CONTROL SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of SYSTEM CONTROL SCHEDULE is improper.'

    ELSE IF( IA.EQ.45 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified SYSTEM CONTROL SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.46 ) THEN
      WRITE(6,*) 'WALL#',IB,' in ROOM#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'inquires REFERENCE WALL No. out of range.'

    ELSE IF( IA.EQ.47 ) THEN
      WRITE(6,*) 'WINDOW#',IB,' on WALL#',IC,' in ROOM#',ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'inquires REFERENCE WALL No. out of range.'

    ELSE IF( IA.EQ.48 ) THEN
      WRITE(6,*) 'WALL#',IB,' in ROOM#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'inquires WALL TYPE out of range.'

    ELSE IF( IA.EQ.49 ) THEN
      WRITE(6,*) 'FILE of DETACHED SHADDING DEVICES dosen''t exist.'

    ELSE IF( IA.EQ.50 ) THEN
      WRITE(6,*) 'Cannot read data from DETACHED SHADDING DEVICES.'

    ELSE IF( IA.EQ.51 ) THEN
      WRITE(6,*) 'GEOMETRY of WALL#',IB,' in ROOM#',IC,' is wrong.'

    ELSE IF( IA.EQ.52 ) THEN
      WRITE(6,*) 'GEOMETRY of WINDOW#',IB,' on WALL#',IC
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'in ROOM#',ID,' is wrong.'

    ELSE IF( IA.EQ.53 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of OCCUPANCY in ROOM#',ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.54 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified OCCUPANCY SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of OCCUPANCY SCHEDULE is improper.'

    ELSE IF( IA.EQ.55 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified OCCUPANCY SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.56 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of LIGHTING in ROOM#',ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.57 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified LIGHTING SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of LIGHTING SCHEDULE is improper.'

    ELSE IF( IA.EQ.58 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified LIGHTING SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.59 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of VENTILATION in ROOM#',ID
```

```
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.60 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified VENTILATION SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of VENTILATION SCHEDULE is improper.'

    ELSE IF( IA.EQ.61 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      WRITE(6,*) 'specified VENTILATION SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.62 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of INFILTRATION in ROOM#',ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.63 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified INFILTRATION SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of INFILTRATION SCHEDULE is improper.'

    ELSE IF( IA.EQ.64 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified INFILTRATION SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.65 ) THEN
      WRITE(6,*) 'Wrong inquiry:SCHEDULE of EQUIPMENT in ROOM#'
     +           ,ID
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'on date [dd/mm]',IB,'/',IC
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'Your requirement is out of range.'

    ELSE IF( IA.EQ.66 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is under '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified EQUIPMENT SCHEDULE in ROOM#',ID
      CALL XY(LINE+2,INDENT)
      WRITE(6,*) 'or SERIES of EQUIPMENT SCHEDULE is improper.'

    ELSE IF( IA.EQ.67 ) THEN
      WRITE(6,*) 'The date [dd/mm]',IB,'/',IC,' is over '
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) 'specified EQUIPMENT SCHEDULE in ROOM#',ID

    ELSE IF( IA.EQ.68 ) THEN
      WRITE(6,*) '***** ERROR :: COMPUTATION DIVERGES'
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) '-[ CAPACITY OF SYSTEM TENDS TO BE IMPROPER. ]-'

    ELSE IF( IA.EQ.69 ) THEN
      WRITE(6,*) '***** ERROR :: COMPUTATION DIVERGES'
      CALL XY(LINE+1,INDENT)
      WRITE(6,*) '-[ CAPACITY OF SOME SYSTEMS ARE PRETTY ',
     +           'LITTLE TO BE PROPER. ]-'

    ELSE IF( IA.EQ.70 ) THEN
      WRITE(6,*) '***** ERROR :: Cannot read THE ORIENTATION',
     +           'OF THE BUILDING.'

    ELSE IF( IA.EQ.71 ) THEN
      WRITE(6,*) '***** ERROR :: COMMENT LINES DISAPPEAR'

    ELSE IF( IA.EQ.72 ) THEN
      WRITE(6,*) '***** ERROR :: CANNOT DETECT END OF COMMENT'

    END IF

    CALL XY(LINE+3,35)
    WRITE(6,*) 'PRESS ENTER'
    READ(5,*)
    CALL TAT(0)
    CALL TAT(1)
    CALL TCL(36,40)

    STOP
    END

    FUNCTION H0(TEMP,VEL,L,ERR)
    REAL H0
    REAL TEMP,VEL,L,ERR
C   ================================================================
C   |                                                              |
C   |    FUNCTION FOR CALCULATING FORCED CONVECTIVE COEFFICIENT     |
C   |    WITH ASSUMING Pr=.71 (for atmospheric condition). THIS     |
C   |    FUNCTION IS LIMITED WITH Re < 8*10^8 (ERR = 1).            |
C   |                                                              |
C   ================================================================

C   @ ---- ASSIGN VARIABLES                                        @
C   @         DUMMY           DUMMY VARIABLE                     @
C   @         ERR             ERR INDICATOR                      @
C   @         K               CONDUCTIVITY OF AIR (W/m.K)        @
C   @         L               LENGHT OF CONVECTED SURFACE (m)    @
```

```
C     @         RE              Reynolds No.                              @
C     @         TEMP            TEMPERATURE OF AIR (C)                    @
C     @         V               KINEMATIC VISCOSITY (m^2/s)               @
C     @         VEL             VELOCITY OF WIND (m/s)                    @

      REAL RE,K,V
      REAL DUMMY

C     ----- ASSIGN ERROR INDICATOR TO BE 0 (no error appears)      -----
      ERR = 0

C     ----- IF VELOCITY OF WIND = 0, H0 WILL BE 0 TOO SINCE WE     -----
C     ----- HAVE ASSUMED THAT FREE CONVECTIVE HEAT TRANSFER IS     -----
C     ----- EQUAL TO RADIATIVE HEAT TRANSFER.                      -----
      IF( VEL.EQ.0.0 ) THEN
         H0 = 17.
         RETURN
      END IF

C     ----- CALCULATE KINEMATIC VISCOSITY, CONDUCTIVITY & Re       -----
      DUMMY = TEMP + 273.15
      V =-8.E-14/3.*DUMMY**3+1.4E-10*DUMMY**2+5.42E-8/3.*DUMMY-1.41E-6
      K =-4.E-10/3.*DUMMY**3+6.E-8*DUMMY**2+23.2E-5/3.*DUMMY+1.3E-3
      RE = VEL*L/V

C     ----- CLASSIFY WHICH CASE WE HAVE TO CALCULATE :-            -----
C     -----    RE => 10^8              OUT OF RANGE    (ERR = 1)   -----
C     -----  5*10^5 < RE < 10^8        TURBULENT FLOW              -----
C     -----    RE =< 5*10^5            LAMINAR FLOW                -----
      IF( RE.GE.1.E8 ) ERR = 1
      IF( RE.GT.5.E5 ) THEN
         H0 = K/L*(0.033*RE**.8 - 777)
      ELSE
         H0 = K/L*0.592*RE**.5
      END IF

C     ----- FOR THIS THESIS WE USE CONSTANT H0 = 17.0               -----
      H0 = 17.0

      RETURN
      END

      SUBROUTINE ANGLE(LLT,SAZ,TILT,LSM,LON,DATE,ALT,AZM,COSINE)
      REAL    LLT,SAZ,TILT,LSM,LON,DATE
      REAL    ALT(24),AZM(24),COSINE(24)
C     ================================================================
C     |                                                              |
C     |      SUBPROGRAM TO COMPUTE COSINE OF INCIDENT ANGLE OF       |
C     |      SOLAR BEAM WITH VECTOR OF THE SURFACE THAT WE ARE       |
C     |      INTERESTED, ALTITUDE AND AZIMUTH OF SOLAR BEAM.         |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         AST             APPARENT SOLAR TIME                 @
C     @         DEG             DEGREE TO RADIAL CONVERTING VALUE   @
C     @         DECL            DECLINATION                         @
C     @         ET              EQUATION OF TIME                    @
C     @         GAMMA           ANGLE BETWEEN AZIMUTH OF SOLAR AND  @
C     @                         THE SURFACE                         @

C     ----- ASSIGN VALUE OF DEG                                     -----
      DEG = ATAN(1.0)/45.0

C     ----- ASSIGN VALUE OF ET AND DECL MONTHLY                     -----
      ET = EQUA_TIM(DATE)
      DECL = -0.0595+23.4272*SIN(6.1877*DATE-1.3575)

C     ----- COMPUTE COSINE, THE INCIDENT ANGLE, HOURLY              -----
      DO 10 T=1,24

C     ----- CALCULATE APPARENT SOLAR TIME (AST)                     -----
         AST = FLOAT(T)*60. + ET + 4.*(LSM-LON)

C     ----- CALCULATE SOLAR ALTITUDE & SOLAR AZIMUTH                -----
         ALT(T) = ASIN(COS(LLT*DEG)*COS(DECL*DEG)
     +        *COS(0.25*(12.*60.-AST)*DEG) + SIN(LLT*DEG)*SIN(DECL*DEG))
         AZM(T) = ACOS((SIN(ALT(T))*SIN(LLT*DEG)-SIN(DECL*DEG))
     +        /(COS(ALT(T))*COS(LLT*DEG)))

C     ----- CALCULATE COSINE OF ANGLE BETWEEN SOLAR BEAM AND
C                                          THE VECTOR OF SURFACE -----
         IF( AST.GT.720 .AND. AST.LT.1440 ) AZM(T) = -AZM(T)
         GAMMA = AZM(T) - (180.0-SAZ)*DEG                          *new*
         COSINE(T) = COS(ALT(T))*COS(GAMMA)*SIN(TILT*DEG)
     +             + SIN(ALT(T))*COS(TILT*DEG)

   10 CONTINUE

      RETURN
      END

      FUNCTION EQUA_TIM(DATE)
      REAL DATE
C     ================================================================
C     |                                                              |
C     |    FUNCTION FOR COMPUTING EQUATION OF TIME BY USING LINEAR   |
C     |    INTERPOLATION.                                            |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         ET's            EQUATION OF TIME ON STANDARD DATE   @
C     @         I               COUNTER                             @
```

```
C     @          STDDATE          STANDARD DATE ( eg. FEB'2 = (31+2)/   @
C     @                           365 or 366 )                          @

      REAL ET(12),STDDATE(12)
      INTEGER I

C     ----- ASSIGN INITIAL VALUES INTO VARIABLES                  -----
      DATA ET    / -11.2, -13.9, -7.5, 1.1, 3.3, -1.4, -6.2, -2.4, 7.5,
     +             15.4, 13.8, 1.6 /
      DATA STDDATE / 0.058, 0.142, 0.219, 0.304, 0.386, 0.471, 0.553,
     +             0.638, 0.723, 0.805, 0.89, 0.973 /

C     ----- CHECK RANGE OF DOMAIN (DATE must be within 0 TO 1)    -----
      IF( DATE.GT.1. .OR. DATE.LT.0. ) THEN
         EQUA_TIM = 0
         RETURN
      END IF

C     ----- COMPUTING VALUE OF EQUATION OF TIME BY LINEAR         -----
C     ----- INTERPOLATION.                                        -----
      IF( DATE.LE.STDDATE(1) ) THEN
         EQUA_TIM = ET(12) + (ET(1)-ET(12))/(1.+STDDATE(1)-STDDATE(12))
     +              *(1.+DATE-STDDATE(12))
         RETURN
      END IF

      IF( DATE.GE.STDDATE(12) ) THEN
         EQUA_TIM = ET(12) + (ET(1)-ET(12))/(1.+STDDATE(1)-STDDATE(12))
     +              *(DATE-STDDATE(12))
         RETURN
      END IF

      DO 10 I=1,12
         IF( DATE.LT.STDDATE(I) ) THEN
            EQUA_TIM = ET(I-1)+(ET(I)-ET(I-1))/(STDDATE(I)-STDDATE(I-1))
     +                 *(DATE-STDDATE(I-1))
            RETURN
         END IF
   10 CONTINUE

      END

      SUBROUTINE SOL_AIR(DATE,DBT,IDN,ID,HO,ALT,COSINE,ALPHA,TILT,SHGF,
     +                   SOLTEMP)
      REAL    DBT(24),IDN(24),ID(24),HO(24),ALT(24),COSINE(24)
      REAL    ALPHA,TILT,DATE
      REAL    SHGF(24),SOLTEMP(24)

C     ================================================================
C     |                                                              |
C     |      SUBPROGRAM TO COMPUTE SOL-AIR TEMP AND SHGF FOR A DAY   |
C     |      BY USING WEATHER DATA OF THAT DAY.                      |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                       @
C     @         A              APPARENT SOLAR CONTANT (W/m^2)      @
C     @         ADDING         THE THIRD TERM OF EQ(3)             @
C     @         B              ATMOSPHERIC EXTINCTION COEFFICIENT  @
C     @         C              SKY DIFFUSE FACTOR                  @
C     @         DEG            DEGREE TO RADIUS CONVERTING VALUE   @
C     @         HO             OUTDOOR CONVECTIVE AND RADIATIVE    @
C     @                        HEAT TRANSFER COEFFICIENT           @
C     @         T              TIME                                @
C     @         Y              RATIO OF VERT/HOR SKY DIFFUSE       @

      REAL    ADDING
      REAL    DEG
      REAL    A,B,C,Y
      INTEGER T

C     ----- ASSIGN VALUE OF HO,ADDING & DEG OF VERTICAL SURFACE   -----
      ADDING = -3.9
      DEG = ATAN(1.0)/45.0

      A = 79.0*COS(DATE*6.283185308)+1160.0
      B = -.033*COS((DATE-10./365.)*6.283185308)+0.175
      C = -.040*COS((DATE- 5./365.)*6.283185308)+0.095

      DO 10 T=1,24
C     ----- CALCULATE TOTAL SOLAR INTENSITY                       -----
         IF( SIN(ALT(T)).LE.0.0 ) THEN
            DUMMY = 0.0
         ELSE
            DUMMY = ID(T)/C
         END IF
         Y = 0.55 + 0.437*COSINE(T) + 0.313*COSINE(T)
         SHGF(T) = C*DUMMY*(COS(TILT*DEG)+Y*(1.0-COS(TILT*DEG)))
     +             +DUMMY*(C*ALT(T))*0.1*(1-COS(TILT*DEG))

         IF( COSINE(T).GT.0.0 ) SHGF(T) = SHGF(T) + IDN(T)*COSINE(T)

C     ----- CALCULATE SOL-AIR TEMP AT TIME_T                      -----
         SOLTEMP(T) = DBT(T)+ALPHA*SHGF(T)/HO(T)+ADDING*COS(TILT*DEG)
   10 CONTINUE

      RETURN
      END

      SUBROUTINE HEATGAIN(B,NUM,SOLTEMP,A,U,TRC,HG)
      REAL B(24)
      REAL SOLTEMP(24),HG(24)
      REAL A,U,TRC
      INTEGER NUM
```

```
C     ================================================================
C     |                                                              |
C     |   HEAT GAIN WITHIN 24-HOUR PERIOD WILL BE CALCULATED IN THIS |
C     |   SUBROUTINE BY HOUR-BY-HOUR METHOD.                         |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         I              COUNTER                             @
C     @         J              COUNTER                             @
C     @         K              COUNTER                             @

      INTEGER I,J,K

C     ----- INITIALIZE VARIABLES                                    -----
      DO 10 I=1,24
         HG(I) = 0.0
   10 CONTINUE

C     ----- HOUR-BY-HOUR CALCULATION                                -----
   20 DO 40 I=1,24
         DO 30 J=1,NUM
            K = I - J + 1
   25       IF( K.LE.0 ) THEN
               K = K + 24
               GO TO 25
            END IF
            HG(I) = HG(I) + B(J)*SOLTEMP(K)
   30    CONTINUE
         HG(I) = ( HG(I) - U*TRC )*A
   40 CONTINUE

      RETURN
      END

      SUBROUTINE COEFF(CP,KT,DEN,THICKNESS,HO,BN,U,NUM,ERR)
      REAL CP(*),KT(*),DEN(*),THICKNESS(*)
      REAL BN(24)
      REAL HO,U
      REAL ERR
      INTEGER NUM
C     ================================================================
C     |                                                              |
C     | THIS SUBPROGRAM COMPUTES COEFFICIENTS FOR HEAT GAIN COMPUTING |
C     | BY USING FINITE ELEMENT METHOD.     THIS SUBPROGRAM LIMIT THE |
C     | LAYER OF STRUCTURES TO BE 8 ( or less ).                     |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         A's            MATRIX OF SIMUTANEOUS EQUATIONS     @
C     @         C's            Cp*THICKNESS*DENSITY                @
C     @         I              COUNTER                             @
C     @         K's            Kt/THICKNESS                        @
C     @         NLAYER         NUMBER OF LAYER                     @
C     @         Q's            VECTOR OF SIMUTANEOUS EQUATIONS     @
C     @         T's            TEMPERATURE AT NODES                @

      INTEGER I
      INTEGER NLAYER
      REAL A(2,19)
      REAL T(19),Q(19)
      REAL C(18),K(18)

C     ----- COMPUTE NUMBER OF LAYER                                 -----
      DO 10 I=8,1,-1
         IF( KT(I).NE.0.0 ) THEN
            NLAYER = I
            GO TO 20
         END IF
   10 CONTINUE

C     ----- IN CASE THAT THER ISN'T ANY LAYER WITHIN THE            -----
C     ----- CONSTRUCTION, THIS SUBROUTINE WILL RETURN AN ERROR.     -----
      ERR = 1
      RETURN

   20 CONTINUE
      ERR = 0

C     ----- ASSIGN INITIAL VALUE INTO VARIABLES                     -----
      DO 30 I=1,100
   30    T(I) = 0.0
      CALL SLLAYER(DEN,CP,KT,THICKNESS,NLAYER,C,K,U)

C     ----- ASSIGN C & K FOR OUTDOOR & INSIDE AIR FILMS,
C                                              RESPECTIVELY. -----
      C(1) = 0.0
      K(1) = HO
      C(NLAYER) = 0.0
      K(NLAYER) = 8.3

C     ----- ADD CONDUCTANCE OF BOTH AIR FILMS INTO U-VALUE OF
C                                              THE STRUCTURE. -----
      U = 1./( 1./U + 1./K(1) + 1./K(NLAYER) )

C     ----- SOLVE FOR Bn                                            -----
      CALL SOLVE(A,T,Q,C,K,U,NLAYER,BN,NUM)

      RETURN
      END

      SUBROUTINE SLLAYER(DEN,CP,KT,THICKNESS,NLAYER,C,K,U)
      REAL CP(*),KT(*),DEN(*),THICKNESS(*)
```

```
      REAL C(*),K(*)
      REAL U
      INTEGER NLAYER
C     ================================================================
C     |                                                              |
C     |     THIS SUBROUTINE IS USED TO SLIDE STRUCTURE COMPONENTS    |
C     |    FROM ABOUT  8 LAYERS INTO ABOUT  16 LAYERS. THEN COMPUTE  |
C     |    VALUE C AND K FOR EACH NEW LAYER AND U-VALUE OF STRUCTURE.|
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         DUMMY           DUMMY VARIABLE                     @
C     @         I               COUNTER                            @
C     @         J               COUNTER                            @
C     @         SPAN            APPROXIMATE THICKNESS OF EACH LAYER @

      INTEGER I,J
      REAL DUMMY,SPAN

C     ----- CALCULATING U-VALUE OF THE STRUCTURE                    -----
      DUMMY = 0.0
      DO 5 I=1,NLAYER
         IF( THICKNESS(I).NE.0.0 ) THEN
            DUMMY = DUMMY + THICKNESS(I)/KT(I)
         ELSE
            DUMMY = DUMMY + 1./KT(I)
         END IF
    5 CONTINUE
      U = 1/DUMMY

C     ----- COMPUTING APROXIMATE THICKNESS OF EACH LAYER            -----
      SPAN = 0.
      DO 10 I=1,NLAYER
         IF( THICKNESS(I).LT.0.0 ) THEN
            WRITE(6,100)
            STOP
         END IF
   10    SPAN = SPAN + THICKNESS(I)
      SPAN = SPAN/9.

C     ----- THE FIRST NLAYER IS RESERVED FOR OUTDOOR AIR FILM       -----
      J = NLAYER
      NLAYER = 1

C     ----- COMPUTING VALUE C AND K FOR EACH NEW LAYER              -----
      DO 30 I=1,J
         DUMMY = 0.
   20    NLAYER = NLAYER + 1
         DUMMY = DUMMY + SPAN
         IF( DUMMY.LT.THICKNESS(I) ) THEN
            C(NLAYER) = DEN(I)*CP(I)*SPAN
            K(NLAYER) = KT(I)/SPAN
            GO TO 20
         END IF
         DUMMY = SPAN - (DUMMY-THICKNESS(I))
         C(NLAYER) = DEN(I)*CP(I)*DUMMY
         IF( THICKNESS(I).EQ.0.0 ) THEN
            K(NLAYER) = KT(I)
         ELSE
            K(NLAYER) = KT(I)/DUMMY
         END IF
   30 CONTINUE

C     ----- ADD A UNIT INTO NLAYER FOR INTERNAL AIR FILM            -----
      NLAYER = NLAYER + 1

  100 FORMAT( ' ***** ERROR :: THERE IS AN INVALID THICKNESS OF',
     +        'CONSTRUCTION (THICKNESS < ZERO).' )

      RETURN
      END

      SUBROUTINE SOLVE(A,T,Q,C,K,U,NLAYER,BN,NUM)
      REAL A(2,*)
      REAL T(*),Q(*)
      REAL C(*),K(*)
      REAL BN(*)
      REAL U
      INTEGER NLAYER,NUM
C     ================================================================
C     |                                                              |
C     |  THIS SUBPROGRAM FINDS OUT TEMPERATURE DISTRIBUTION AND HEAT |
C     |  RATE AT BOTH THE SURFACES OF THE STRUCTURE. ADDITIONALLY,   |
C     |  IT WILL CHECK THE CONVERGENCE OF CALCULATING FOR THE BEST   |
C     |  RESULTS. FINALLY, IT CALCULATES A SET OF COEFFICIENT.       |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         CONST           CONSTANT OF OUTDOOR TEMP PROGESSION @
C     @         I               COUNTER                            @
C     @         N               NUMBER OF CALCULATED NODE          @
C     @         SLOPE           SLOPE OF OUTDOOR TEMP PROGESSION   @
C     @         TIME            TIME (IN SECOND UNIT)              @
C     @         TP's            STAND-BY OF T's                    @

      INTEGER I
      INTEGER N,INDEX
      REAL TP(19)
      REAL SLOPE,CONST

C     ----- ASSIGN INITIAL VALUE INTO VARIABLES                     -----
      N = NLAYER + 1
```

```
      SLOPE = 1./3600.
      CONST = 0.0
      NUM = 1
      INDEX = 1
      DT = 600.

      DO 5 I=1,24
         BN(I) = 0.0
    5 CONTINUE

C     ----- ASSIGN INITIAL VALUE INTO STRUCTURE TEMPERATURE AND
C           COMPUTE FOR THE TEMPERATURE DISTRIBUTION RESULTS AT
C           THE END OF 3600-SECOND INTERVAL.                    -----
      DO 10 I=1,N
   10    TP(I) = 0.0
      DO 20 TIME=DT,3600,DT
         CALL ASSEMBLY(A,TP,Q,C,K,DT,NLAYER)
         TP(1) = SLOPE*FLOAT(TIME) + CONST
         TP(N) = 0.0
         CALL GSS_E(A,TP,Q,N)
   20 CONTINUE

C     ----- RECOMPUTE THE TEMPERATURE DISTRIBUTION AT THE END OF
C           3600-SECOND INTERVAL WITH TIME STEP, DEVIED BY 2
C           SO THAT WE GET MORE ACCURACY.                       -----
   30 DT = DT/2.

      DO 40 I=1,N
   40   T(I) = 0.0
   45 CONTINUE
      DO 50 TIME=DT,3600,DT
         CALL ASSEMBLY(A,T,Q,C,K,DT,NLAYER)
         T(1) = SLOPE*FLOAT(TIME) + CONST
         T(N) = 0.0
         CALL GSS_E(A,T,Q,N)
   50 CONTINUE

C     ----- DETERMINE THE APPROPIATE TIME STEP BY COMPARISON TWO
C           ABOVE RESULTS ( TP[ with DT ] AND T[ with DT/2 ] ).
C           IF THEY ARE COMPARABLE, WE WILL GET A GOOD DT.      -----
      IF( NUM.EQ.1 ) THEN
         DUMMY = 0.0
         SUM = 0.0
         DO 60 I=1,N
            DUMMY = DUMMY + ABS(TP(I)-T(I))
            SUM = SUM + ABS(T(I))
            TP(I) = T(I)
   60    CONTINUE
         IF( (DUMMY/SUM).GT.0.01 .AND. DT.GT.0.01 ) GO TO 30
      END IF

C     ----- CALCULATE COEFFICIENT Bn's                          -----
         BN(INDEX) = -Q(N)/2./DT + BN(INDEX)

C     ----- CHECK WHETHER THE COMPUTING IS ENOUGH BY COMPARING
C           U-VALUE TO SUM OF Bn's.                             -----
      SUM = 0.0
      DO 70 I=1,24
   70    SUM = SUM + BN(I)/U

C     ----- IF WE MUST FURTHER THE COMPUTING, SKIP THE NUM TO THE
C           NEXT HOUR AND  CHANGE OUTDOOR TEMPERATURE CURVE TO BE
C           NEW ONE.                                            -----
      IF( (SUM.GT.1.01 .OR. SUM.LT..99) .AND. (NUM.LT.100) ) THEN
         NUM = NUM + 1
         INDEX = INDEX + 1
         IF( INDEX.GT.24 ) INDEX = INDEX - 24
         IF( NUM.EQ.2 ) THEN
            SLOPE = -1./3600.
            CONST =  1.
         ELSE
            SLOPE = 0.0
            CONST = 0.0
         END IF
         GO TO 45
      END IF

C     ----- ASJUST BN's SO THAT THE SUMMATION OF BN's WILL BE THE -----
C     ----- SAME VALUE AS U-VALUE (according to the assumtion).   -----
      DO 80 I=1,24
   80    BN(I) = BN(I)/SUM

C     ----- THE MAXIMUM NUMBER OF BN IS "24"                    -----
      IF( NUM.GT.24 ) NUM = 24

      RETURN
      END

      SUBROUTINE ASSEMBLY(A,T,Q,C,K,DT,NLAYER)
      REAL A(2,*)
      REAL T(*),Q(*)
      REAL C(*),K(*)
      REAL DT
      INTEGER NLAYER
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE WORKS AS A MATRIX ASSEMBLER FOR CRETING   |
C     |    A 2-HAFTBANDWITHD MATRIX [A] AND A VECTOR (Q) INTO FORM   |
C     |                                                              |
C     |                  [ A11  A21 ]       [ Q1 ]                   |
C     |                  [ A12  A22 ]       [ Q2 ]                   |
C     |                  [ A13   .  ]  AND  [ .  ]                   |
C     |                  [  .    .  ]       [ .  ]                   |
C     |                  [ Aln      ]       [ .  ]                   |
```

```
C     |                                                                     |
C     |    THIS MATRIX AND VECTOR WILL BE USED IN  COMPUTING TO             |
C     |    SOLVE FOR A SET OF ANSWER BY FINITE ELEMENT METHODS.             |
C     |                                                                     |
C     =======================================================================

C     @ ---- ASSIGN VARIABLES                                               @
C     @         I               COUNTER                                   @
C     @         J               COUNTER                                   @

      INTEGER I,J

C     ----- ASSIGN INITAIL VALUE INTO VARIABLES                           -----
      J = NLAYER + 1
      DO 10 I=1,J
         A(1,I) = 0.
         A(2,I) = 0.
         Q(I) = 0.
   10 CONTINUE

C     ----- ASSEMBLE FORM THE FIRST ROW TO THE LAST ROW                   -----
      DO 20 I=1,NLAYER
         A(1,I)   = A(1,I) + (C(I) + 2.*K(I)*DT)
         A(2,I)   = A(2,I) + (-2.*K(I)*DT)
         A(1,I+1) = A(1,I+1) + (C(I) + 2.*K(I)*DT)
         Q(I) = Q(I) + C(I)*T(I)
         Q(I+1) = Q(I+1) + C(I)*T(I+1)
   20 CONTINUE

      RETURN
      END

      SUBROUTINE GSS_E(A,T,Q,N)
      REAL A(2,*)
      REAL T(*)
      REAL Q(*)
      INTEGER N
C     =======================================================================
C     |                                                                     |
C     |  THIS PROGRAM IS USED FOR SOLVING A SET OF 2-HALFBANDWIDTH          |
C     |   EQUATION, WHICH THE FIRST AND LAST VARIABLES WERE KNOWN .         |
C     |                                                                     |
C     |             [ A11  A21 ] [ T1* ]     [ Q1  ]                      |
C     |             [ A12   .  ] [ T2  ]     [ Q2* ]                      |
C     |             [  .    .  ] [  .  ]  =  [  .* ]                      |
C     |             [ Aln      ] [ Tn* ]     [ Qn  ]                      |
C     |                                                                     |
C     |              note - asterisks mark the known value variables.       |
C     =======================================================================

C     @ ---- ASSIGN VARIABLES                                               @
C     @         I               COUNTER                                   @
C     @         J               COUNTER                                   @
C     @         SCALE           SCALE FOR ELIMINATING AN ELEMENT          @

      INTEGER I,J
      REAL SCALE

C     ----- ELIMINATING ALL ELEMENTS OF MATRIX A, RELATING TO B1
C           (EXCEPT THE ELEMENTS ON THE FIRST & LAST ROW).          -----
      Q(2)   = Q(2)   - A(2,1)*T(1)

C     ----- MAKING MATRIX A BEING AN UPPER DIAGONAL MATRIX BY
C           ELIMINATING ALL VALUE BELOW THE DIAGONAL ELEMENT OF
C           THE MATRIX, EXCLUDING ELEMENTS ON THE 1st & LAST ROW. -----
      DO 20 I=2,N-2
         J = I + 1
         SCALE = A(2,I)/A(1,I)
         A(1,J) = A(1,J) - SCALE*A(2,I)
         Q(J) = Q(J) - SCALE*Q(I)
   20 CONTINUE

C     ----- BACKWARD CALCULATION TO GET ALL VALUES                        -----
      DO 30 I=N-1,2,-1
         T(I) = (Q(I) - A(2,I)*T(I+1))/A(1,I)
   30 CONTINUE

      Q(1) = A(1,1)*T(1)     + A(2,1)*T(2)
      Q(N) = A(2,N-1)*T(N-1) + A(1,N)*T(N)

      RETURN
      END

      SUBROUTINE HEATLOAD(QL,HG,V,W,KT,ERR)
      REAL KT,ERR
      REAL QL(24),HG(24)
      REAL V(3),W(2)
C     =======================================================================
C     |                                                                     |
C     |     THIS SUBROUTINE HOURLY CALCULATES HEAT LOAD THAT FLOW           |
C     |     THROUGH WALLS OR ROOFS.          ALL METHOD REFER TO            |
C     |"METHODOLOGY AND EQUATIONS FOR  HOUR-BY-HOUR LOAD CALCULATIONS"|
C     |       EQ 44 CHAPTER 26 OF ASHRAE HANDBOOK, FUNDAMENTAL              |
C     |                                                                     |
C     =======================================================================

C     @ ---- ASSIGN VARIABLES                                               @
C     @         DUMMY           DUMMY VARIABLE                            @
C     @         FC              FACTOR OF COOLING LOAD                    @
C     @         I,J,N           COUNTER OR DUMMY VARIABLE                 @
C     @         QLP's           PREVIOUS QL's                             @

      REAL    QLP(24)
      REAL    FC,DUMMY
```

```
      INTEGER I,J,N

C     ----- INITIALLY SET ALL HEAT LOAD BE ZERO                    -----
      DO 10 I=1,24
   10    QL(I) = 0.

      ITER = 0
C     ----- DO LOOP OF HEAT LOAD FOR 1 DAY (HOURLY)                -----
   20 ITER = ITER + 1
      IF( ITER.GT.100 ) THEN
         ERR = 1
         RETURN
      END IF

      DO 40 I=1,24

C     ----- CALCULATE HEAT LOAD BY EQUATION 26                     -----
         QLP(I) = QL(I)
         QL(I) = 0.
         DO 25 N=1,2
            J = I - N
            IF( J.LE.0 ) J = J + 24
   25       QL(I) = QL(I)-W(N)*QL(J)
         DO 30 N=1,3
            J = I - N + 1
            IF( J.LE.0 ) J = J + 24
            FC = (1.0 - 0.0116*KT)
   30       QL(I) = QL(I) + FC*V(N)*HG(J)
   40 CONTINUE

C     ----- CHECK THE CONVERGENCE OF THE RESULTS                   -----
      DO 50 I=1,24
         DUMMY = ABS(QL(I)-QLP(I))
         IF( DUMMY.GT.1.E-3 ) GO TO 20
   50 CONTINUE

      ERR = 0

      RETURN
      END

      SUBROUTINE HEATEXTC(QL,SUM_UA,V,VI,TRTS,THTR,TRC,GG,P,FA,
     +                                    ERMX,ERMN,TRT,ER,NUM_DAY,ERR)
      REAL QL(*),V(*),VI(*),TRTS(*)
      REAL ERMX(*),ERMN(*),TRT(*),ER(*)
      REAL GG(3),P(2)
      REAL THTR,TRC,SUM_UA,FA,ERR
      INTEGER NUM_DAY
C     ================================================================
C     |                                                              |
C     |      PROGRAM FOR CALCULATION HEAT EXTRACTION OF A ROOM       |
C     |      BY HOUY-BY-HOUR METHOD.                                 |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                       @
C     @         A's             = G0/(S-G0)*W                      @
C     @         B's             = S/(S-G0)                         @
C     @         DIFF            DIFFERANCE VALUE OF TEMPERATURE FOR  @
C     @                         CHECKING CONVERGENCE OF THE COMPUTING @
C     @         G's             CONSTANT G(0,1,2) FROM TABLE 34    @
C     @         I               COUNTER                            @
C     @         IT              PARAMETER FOR COMPUTING VALUE ER(time)@
C     @         ITER            ITERATION OF COMPUTING             @
C     @         J               COUNTER                            @
C     @         K               COUNTER                            @
C     @         S's             PARAMETER S(time) OF SYSTEM        @
C     @         SUMG's          SUM OF G(0,1,2)                    @
C     @         TRTP            PREVIOUS ROOM TEMPERATURE OF TIME (t) @
C     @         W's             PARAMETER W(time) OF SYSTEM        @

      REAL G(3,168)
      REAL W(168),S(168),A(168),B(168),SUMG(168)
      REAL IT,TRTP,DIFF
      INTEGER I,J,K,ITER
      INTEGER NUM_HOUR

      NUM_HOUR = NUM_DAY*24
      ERR = 0

      DO 10 I=1,NUM_HOUR
         ER(I) = 0.0
         TRT(I) = 27.0

         G(1,I) = GG(1)*FA + P(1)*(SUM_UA + 1232.*(V(I) + VI(I)))
         J = I - 1
         IF( J.LE.0 ) J = J + NUM_HOUR
         G(2,I) = GG(2)*FA + P(2)*(SUM_UA + 1232.*(V(J) + VI(J)))
         G(3,I) = GG(3)*FA
         SUMG(I) = G(1,I)+G(2,I)+G(3,I)

         S(I) = (ERMX(I)-ERMN(I))/THTR
         W(I) = (ERMX(I)+ERMN(I))/2. - S(I)*TRTS(I)

         A(I) = G(1,I)/(S(I)+G(1,I))*W(I)
   10    B(I) = S(I)/(S(I)+G(1,I))

      ITER = 1
   20 DIFF = 0.0
      DO 30 I=1,NUM_HOUR
         J = I - 1
         IF( J.LE.0 ) J = J + NUM_HOUR
         K = I - 2
         IF( K.LE.0 ) K = K + NUM_HOUR
```

```
      IT = TRC*SUMG(I)
      IT = IT - G(2,I)*(TRT(J)) - G(3,I)*(TRT(K))
      IT = IT + P(1)*QL(I) + P(2)*QL(J) - P(2)*ER(J)

      ER(I) = A(I) + B(I)*IT

      TRTP = TRT(I)
      IF( ER(I).LT.ERMN(I) ) ER(I) = ERMN(I)
      IF( ER(I).GT.ERMX(I) ) ER(I) = ERMX(I)
      TRT(I) = (1/G(1,I))*(IT - ER(I))
      DIFF = DIFF + ABS(TRT(I)-TRTP)
   30 CONTINUE

      ITER = ITER + 1
      IF( DIFF.GT.0.1 .AND. ITER.LT.100 ) GO TO 20

      IF( DIFF.GT.0.1 .AND. ITER.GE.100 ) ERR = 1

      RETURN
      END

      SUBROUTINE SUNLITAREA(ALT,AZM,COSINE,PX,PY,PZ,W,H,SAZ,TILT,
     +                      RW,RH,PV,PH,FV,FH,RECESS,
     +                      DE_SHAD,UNIT_NUM,SUNLIT_A,ERR)
      REAL ALT(24),AZM(24),COSINE(24),SUNLIT_A(24)
      REAL PX,PY,PZ,W,H,SAZ,TILT
      REAL RW,RH,PV,PH,FV,FH,RECESS
      CHARACTER*40 DE_SHAD
      INTEGER UNIT_NUM
      REAL ERR
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE WILL FIND OUT THE NET SUNLIT AREA FOR     |
C     |    THE WINDOW THAT WE ARE INTERESTED IN. DE_SHAD = NAME      |
C     |    OF THE FILE THAT CONTAINS DETAIL OF DETACHED SHADING      |
C     |    DEVICES.                                                  |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         A,B,C,D          COEEFICIENTS OF WINDOW PLANE'S EQ.  @
C     @         DEG              CONVERTING VALUE FROM DEG. TO RAD.  @
C     @         DELTA            VERTICAL PROJECTION PROFILE ANGLE   @
C     @         DUMMY            DUMMY VARIABLE                      @
C     @         DUMMY1           DUMMY VARIABLE                      @
C     @         H1               ADJUSTED H IN EACH HOUR OF A DAY    @
C     @         I                COUNTER                             @
C     @         J                COUNTER                             @
C     @         K                COUNTER                             @
C     @         OMEGA            HORIZONTAL PROJECTION PROFILE ANGLE @
C     @         PART's           BREAK-OUT PARTS OF WINDOW           @
C     @         PX1              ADJUSTED PX IN EACH HOUR OF A DAY   @
C     @         PY1              ADJUSTED PY IN EACH HOUR OF A DAY   @
C     @         SH               LENGTH OF SHADOW FROM HOR. PROJECTION @
C     @         SHR              LENGTH OF SHADOW FROM HOR. RECESSION  @
C     @         SW               LENGTH OF SHADOW FROM VER. PROJECTION @
C     @         SWR              LENGTH OF SHADOW FROM VER. RECESSION  @
C     @         W1               ADJUSTED W IN EACH HOUR OF A DAY    @
C     @         X's              X-COORDINATE OF SHADOW CONNERS      @
C     @         X_W's            X-COORDINATE OF WINDOW CONNERS      @
C     @         XI               DUMMY OF X's                        @
C     @         Y's              Y-COORDINATE OF SHADOW CONNERS      @
C     @         Y_W's            Y-COORDINATE OF WINDOW CONNERS      @
C     @         YI               DUMMY OF Y's                        @
C     @         Z's              Z-COORDINATE OF SHADOW CONNERS      @
C     @         Z1               DUMMY OF Z's                        @

      REAL DELTA,OMEGA
      REAL SW,SH,SWR,SHR
      REAL DEG
      REAL PX1,PY1,W1,H1
      REAL X(4),Y(4),Z(4),X_W(4),Y_W(4),Z_W(4)
      INTEGER I,J,K
      REAL DUMMY,A,B,C,D
      REAL X1,Y1,Z1
      INTEGER PART(4,4)

C     ----- ASSING VALUE TO VARIABLE DEG                            -----
      DEG = ATAN(1.)/45.

C     ----- CHECK WETHER OR NOT THE AREA EQUALS ZERO                 -----
      IF( (W*H).EQ.0 ) THEN
         DO 10 I=1,24
   10       SUNLIT_A(I) = 0.
         RETURN
      END IF

C     ----- CHECK WHETHER THE GEOMETRY OF EXTERNAL SHADING DEVICES-----
C     ----- ARE VALID                                               -----
      ERR = 0
      IF((PV.EQ.0 .AND. FV.NE.0) .OR. (PH.EQ.0 .AND. FH.NE.0)) ERR = 1

      DO 100 I=1,24
C     ----- CHECK WETHER THE WINDOW'S ANY EXTERNAL SHADING DEVICES-----
C     ----- IF THERE DOSEN'T EXIST, SKIP THIS MODULE.               -----
         IF( PV.EQ.0 .AND. PH.EQ.0 .AND. FV.EQ.0 .AND. FH.EQ.0 .AND.
     +       RECESS.EQ.0 ) THEN
            IF( ALT(I).LT.0.0 .OR. COSINE(I).LE.0.0 ) THEN
               SUNLIT_A(I) = 0
               GO TO 100
            ELSE
               PX1 = PX
               PY1 = PY
               W1  = W
```

```
            H1  = H
            GO TO 20
         END IF
      END IF

C     ----- IF IT IS NIGHT TIME, SUNLIT AREA WILL BE ZERO AND      -----
C     ----- THIS ITERATION WILL BE SKIPPED, OR                     -----
C     ----- IF THE SUN ISN'T STAYING IN FRONT OF THE WINDOW,       -----
C     ----- SUNLIT_AREA WILL BE ZERO. OTHEREWISE, CALCULATION      -----
C     ----- MUST BE DONE.                                         -----
      IF( ALT(I).LT.0.0 .OR. COSINE(I).LE.0.0 ) THEN
         SUNLIT_A(I) = 0
         GO TO 100
      ELSE
         GAMMA = AZM(I) - SAZ*DEG
         DELTA = ATAN((SIN(GAMMA)*COS(ALT(I)))/COSINE(I))
         OMEGA = ATAN((SIN(ALT(I))*SIN(TILT*DEG)
     +           -COS(ALT(I))*COS(GAMMA)*COS(TILT*DEG))/COSINE(I))

         SW  = (PV+RECESS)*ABS(TAN(DELTA)) + FV - RW
         SH  = (PH+RECESS)*ABS(TAN(OMEGA)) + FH - RH
         SWR = RECESS*ABS(TAN(DELTA))
         SHR = RECESS*ABS(TAN(OMEGA))

C     ----- CHECK WHICH SHADOW IS THE MAIN EFFECT OF THE VERTICAL-----
         IF( ABS(SWR).GT.ABS(SW) ) SW = SWR

C     ----- CHECK WHICH SHADOW IS THE MAIN EFFECT OF THE HORIZON -----
         IF( SHR.GT.SH ) SH = SHR

C     ----- CALCULATE THE SUNLIT AREA OF THE WINDOW               -----
         H1 = H - SH

C     ----- IF GAMMA IS NEGATIVE, THE SHADOW IS TO THE RIGHT      -----
C     ----- SIDE OF THE WINDOW                                    -----
         IF( GAMMA.LT.0 ) THEN
            PX1 = PX + SW*COS(SAZ*DEG)
            PY1 = PY + SW*SIN(SAZ*DEG)
            W1 = W - SW
         ELSE
            W1 = W - SW
         END IF

      END IF

 20   CONTINUE

C     ----- IN CASE THAT EXTERNAL SHADING DEVICES ABSOLUTELY       -----
C     ----- SHAD THE WINDOW.                                      -----
      IF( H1.LE.0 .OR. W1.LE.0 ) THEN
         SUNLIT_A(I) = 0
         GO TO 100
      END IF

      SUNLIT_A(I) = H1*W1

C     ----- STARTING TO COMPUTE EFFECT OF SHADOWS FROM DETACHED    -----
C     ----- SHADING DEVICES.                                      -----

C     ----- OPEN THE DATAFILE OF DETACHED SHADING DEVICES TO       -----
C     ----- READ THE POSITIONS OF THEM.                           -----
      OPEN(UNIT_NUM,FILE=DE_SHAD,ERR=30,STATUS='OLD')
      GO TO 40

C     ----- IF THERE EXISTS AN ERROR ON OPENING THE DATAFILE, SKIP-----
 30   ERR = 2
      GO TO 99

 40   CONTINUE

C     ----- SET ALL PARTS OF THE WINDOW BEING SUNLIT.              -----
      DO 45 J=1,4
         DO 45 K=1,4
 45         PART(J,K) = 1

C     ----- RECEIVE POSITION OF DETACHED SHADING DEVICES           -----
 50   READ(UNIT_NUM,*,END=96,ERR=60) X,Y,Z
      GO TO 70

C     ----- IF THERE EXISTS AN ERROR ON OPENING THE DATAFILE, SKIP-----
 60   ERR = 3
      GO TO 99

 70   CONTINUE

C     ----- COMPUTE EQUATION OF WINDOW PLANE                       -----

      X_W(1) = PX
      Y_W(1) = PY
      Z_W(1) = PZ
      X_W(2) = PX + W*COS(SAZ*DEG)
      Y_W(2) = PY + W*SIN(SAZ*DEG)
      Z_W(2) = PZ
      X_W(3) = PX + W*COS(SAZ*DEG) +
     +         H*COS(TILT*DEG)*COS((SAZ+90.)*DEG)
      Y_W(3) = PY + W*SIN(SAZ*DEG) +
     +         H*COS(TILT*DEG)*SIN((SAZ+90.)*DEG)
      Z_W(3) = PZ + H*SIN(TILT*DEG)

      DUMMY = X_W(1)*Y_W(2)*Z_W(3)+X_W(3)*Y_W(1)*Z_W(2)
     +       +X_W(2)*Y_W(3)*Z_W(1)-X_W(1)*Y_W(3)*Z_W(2)
     +       -X_W(2)*Y_W(1)*Z_W(3)-X_W(3)*Y_W(2)*Z_W(1)

C     ----- WHEN THE GEOMETRY OF THE SHADING DEVICE IS VALID.      -----
```

```
      IF( DUMMY.NE.0 ) THEN
         A = (Y_W(2)*Z_W(3)+Y_W(1)*Z_W(2)+Y_W(3)*Z_W(1)
     +          -Y_W(3)*Z_W(2)-Y_W(1)*Z_W(3)-Y_W(2)*Z_W(1))/DUMMY
         B = (X_W(1)*Z_W(3)+X_W(3)*Z_W(2)+X_W(2)*Z_W(1)
     +          -X_W(1)*Z_W(2)-X_W(2)*Z_W(3)-X_W(3)*Z_W(1))/DUMMY
         C = (X_W(1)*Y_W(2)+X_W(3)*Y_W(1)+X_W(2)*Y_W(3)
     +          -X_W(1)*Y_W(3)-X_W(2)*Y_W(1)-X_W(3)*Y_W(2))/DUMMY
         D = 1.0

      ELSE
         DO 72 J=1,3
            X_W(J) = X_W(J) + 1
            Y_W(J) = Y_W(J) + 1
            Z_W(J) = Z_W(J) + 1
   72    CONTINUE
         DUMMY = X_W(1)*Y_W(2)*Z_W(3)+X_W(3)*Y_W(1)*Z_W(2)
     +           +X_W(2)*Y_W(3)*Z_W(1)-X_W(1)*Y_W(3)*Z_W(2)
     +           -X_W(2)*Y_W(1)*Z_W(3)-X_W(3)*Y_W(2)*Z_W(1)
         IF( DUMMY.NE.0 ) THEN
            A = (Y_W(2)*Z_W(3)+Y_W(1)*Z_W(2)+Y_W(3)*Z_W(1)
     +             -Y_W(3)*Z_W(2)-Y_W(1)*Z_W(3)-Y_W(2)*Z_W(1))/DUMMY
            B = (X_W(1)*Z_W(3)+X_W(3)*Z_W(2)+X_W(2)*Z_W(1)
     +             -X_W(1)*Z_W(2)-X_W(2)*Z_W(3)-X_W(3)*Z_W(1))/DUMMY
            C = (X_W(1)*Y_W(2)+X_W(3)*Y_W(1)+X_W(2)*Y_W(3)
     +             -X_W(1)*Y_W(3)-X_W(2)*Y_W(1)-X_W(3)*Y_W(2))/DUMMY
            D = 1.0 - A - B - C

            DO 73 J=1,3
               X_W(J) = X_W(J) - 1
               Y_W(J) = Y_W(J) - 1
               Z_W(J) = Z_W(J) - 1
   73       CONTINUE

         ELSE

C     ----- IF THE GEOMETRY IS INVALID, SKIP TO THE NEXT DEVICE.  -----
            ERR = 4
            DO 74 J=1,3
               X_W(J) = X_W(J) - 1
               Y_W(J) = Y_W(J) - 1
               Z_W(J) = Z_W(J) - 1
   74       CONTINUE
            GO TO 50
         END IF
      END IF

C     ----- SPECIFY POINTS OF ALL 4 CONNERS OF THE SHADOW BY       -----
C     ----- USING EQUATION OF WINDOW PLANE AND ANGLE OF THE SUN.   -----
      IF( COS(ALT(I)).EQ.0 ) THEN

C     ----- IF THE SUN IS EXACTLY ABOVE OUR HEAD (ALT = 90 deg),   -----
         IF( C.NE.0 ) THEN
C     ----- IN CASE THAT THIS WINDOW ISN'T VERTICAL.               -----
            DO 75 J=1,4
   75          Z(J) = (D-A*X(J)-B*Y(J))/C
         ELSE

C     ----- CONVERSELY, THERE IS NO SHADOW ON THE WINDOW           -----
               GO TO 50
         END IF

      ELSE

C     ----- IN CASE THAT THE SUN ISN'T ABOVE THE WALL              -----
         IF( ALT(I).EQ.0 ) THEN
            DUMMY = 1
         ELSE
            DUMMY = (A*SIN(AZM(I))-B*COS(AZM(I)))/TAN(ALT(I))+C
         END IF

         DO 80 J=1,4
            IF( ALT(I).EQ.0 ) THEN
               DUMMY1 = Z(J)
            ELSE
               DUMMY1 = (D+(DUMMY-C)*Z(J)-A*X(J)-B*Y(J))
            END IF

            Z1 = DUMMY1/DUMMY

C     ----- CHECK IF THIS DEVICE IS IN FRONT OF THE WINDOW         -----
            IF( Z1.LE.Z(J) ) THEN
               IF( ALT(I).EQ.0 ) THEN

C     ----- SPECIAL CASE :: THE SUN IS EXACTLY HORIZONTAL          -----

                  IF( TAN(AZM(I)).EQ.0.0 ) THEN
C     ----- >>> AND IT (THE SUN) IS AT EITHER SOUTH OR NORTH       -----
                     IF( B.EQ.0 ) GO TO 99
                     Y1 = (D-C*Z1-A*X(J))/B
                     X1 = X(J)
                     GO TO 76
                  END IF
                  IF( COS(AZM(I)).EQ.0.0 ) THEN
C     ----- >>> AND IT (THE SUN) IS AT EITHER EAST OR WEST         -----
                     IF( A.EQ.0 ) GO TO 99
                     X1 = (D-C*Z1-B*Y(J))/A
                     Y1 = Y(J)
                     GO TO 76
                  END IF
                  IF( (B-A*TAN(AZM(I))).EQ.0 ) THEN
C     ----- >>> AND IT (THE SUN) IS NORMAL TO THE WINDOW           -----
                     GO TO 99
                  END IF
```

```
                  Y1 = (D-C*Z1-A*Y(J)*TAN(AZM(I))-A*X(J))
     *                 /(B-A*TAN(AZM(I)))
                  X1 = (Y(J)-Y1)*TAN(AZM(I))+X(J)
  76              CONTINUE
                ELSE
                  X1 = (Z1-Z(J))/TAN(ALT(I))*SIN(AZM(I)) + X(J)
                  Y1 = (Z(J)-Z1)/TAN(ALT(I))*COS(AZM(I)) + Y(J)
                END IF
                X(J) = X1
                Y(J) = Y1
                Z(J) = Z1

C     ----- IN CASE THAT IT ISN'T IN FRONT OF THE WINDOW, SKIP TO -----
C     ----- THE NEXT DEVICE.                                      -----
              ELSE
                GO TO 50
              END IF
  80        CONTINUE

        END IF

C    ----- TRANSFER ALL POINTS INTO WINDOWS PLANE (PLANE OF 2-D)  -----
        X_W(1) = 0
        Y_W(1) = 0
        X_W(2) = W
        Y_W(2) = 0
        X_W(3) = W
        Y_W(3) = H
        X_W(4) = 0
        Y_W(4) = H

        DO 90 J=1,4
          DUMMY  = X(J)-PX1
          DUMMY1 = Y(J)-PY1
          X(J) = (DUMMY**2+DUMMY1**2)**.5
          IF( X(J).NE.0 ) THEN
             DUMMY = ACOS(DUMMY/X(J))
             IF( DUMMY1.LT.0 ) DUMMY = -DUMMY
             IF( SIN(DUMMY-(SAZ-90)*DEG).LT.0 ) X(J) = -X(J)
          END IF
          Y(J) = (Z(J)-PZ)*SIN(TILT*DEG)
  90    CONTINUE

C     ----- USING FUNCTION SH_REGION  TO CHECK WETHER THIS WINDOW -----
C     ----- IS WITHIN THE SHADOW AREA.                            -----
        DUMMY = SH_REGION(X,Y,X_W,Y_W,ERR)

C     ----- THE WINDOWS IS IN THE SHADOW REGION.                  -----
        IF( DUMMY.EQ.4 ) THEN
           SUNLIT_A(I) = 0
           GO TO 99

C     ----- THE WINDOW IS OUT OF THE SHADOW REGION.               -----
        ELSE IF( DUMMY.EQ.0 ) THEN
           GO TO 50
        END IF

C     ----- SOME PARTS OF THE WINDOW IS IN THE SHADOW REGION.     -----
C     ----- IN THIS CASE, WE WILL DIVIDE THE WINDOW INTO 16 PARTS -----
C     ----- AND COMPUTE SEPARATELY INDIVIDUAL PART.               -----
        DO 95 J=1,4
           DO 95 K=1,4
              X_W(1) = (J-1)*W/4.
              Y_W(1) = (K-1)*H/4.
              X_W(2) = J*W/4.
              Y_W(2) = (K-1)*H/4.
              X_W(3) = J*W/4.
              Y_W(3) = K*H/4.
              X_W(4) = (J-1)*W/4.
              Y_W(4) = K*H/4.

              DUMMY = SH_REGION(X,Y,X_W,Y_W,ERR)
              IF( ERR.EQ.0 .AND. DUMMY.EQ.4 ) THEN
                 PART(J,K) = 0
              END IF
  95    CONTINUE
        GO TO 50

  96    CONTINUE

C     ----- COMPUTE PERCENTAGE OF SUNLIT AREA TO ALL AREA.        -----
        DUMMY = 0
        DO 97 J=1,4
           DO 97 K=1,4
  97          DUMMY = DUMMY + PART(J,K)
        DUMMY = DUMMY/16.
        SUNLIT_A(I) = H1*W1*DUMMY

  99    CLOSE(UNIT_NUM)

 100 CONTINUE

      RETURN
      END

      FUNCTION SH_REGION(X,Y,PX,PY,ERR)
      REAL X(4),Y(4),PX(4),PY(4)
      REAL ERR
C     ======================================================================
C     |                                                                    |
C     |    THIS FUNTION WILL GIVE HOW MANY PIONTS (PX,PY) ARE THERE        |
C     |    IN THE REGION, COMPOUND OF POINTS (X,Y). THIS WILL USE          |
C     |    MAPPING THE REGION OF QUADILATERAL SHAPE TO BE SQUARE           |
C     |    SHAPE.                                                          |
```

```
C     |                                                                     |
C     =======================================================================

C     @ ---- ASSIGN VARIABLES                                               @
C     @         A,B,C           DUMMY                                       @
C     @         AX,BX,CX,DX     DUMMY                                       @
C     @         AY,BY,CY,DY     DUMMY                                       @
C     @         E               MAPPED PX                                   @
C     @         I               COUNTER                                     @
C     @         J               COUNTER                                     @
C     @         N               MAPPED PY                                   @
C     @         PX's            X-COORDINATE OF CHECKED POINTS              @
C     @         PY's            Y-COORDINATE OF CHACKED POINTS              @
C     @         X's             X-COORDINATE OF THE REGION                  @
C     @         Y's             Y-COORDINATE OF THE REGION                  @

      REAL AX,BX,CX,DX,AY,BY,CY,DY,A,B,C
      REAL N,E
      INTEGER I

C     ----- CHECK WETHER SHADOW SIZE EQUALS ZERO                            -----
      DO 5 I=1,4
         DO 5 J=(I+1),4
            IF(X(I).EQ.X(J) .AND. Y(I).EQ.Y(J)) THEN
               ERR = 1
               SH_REGION = 0
               RETURN
            END IF
    5 CONTINUE
      ERR = 0

C     ----- ASSIGN VALUES INTO DUMMY VARIABLES                              -----
      AX =  X(1) + X(2) + X(3) + X(4)
      BX = -X(1) + X(2) + X(3) - X(4)
      CX = -X(1) - X(2) + X(3) + X(4)
      DX =  X(1) - X(2) + X(3) - X(4)

      AY =  Y(1) + Y(2) + Y(3) + Y(4)
      BY = -Y(1) + Y(2) + Y(3) - Y(4)
      CY = -Y(1) - Y(2) + Y(3) + Y(4)
      DY =  Y(1) - Y(2) + Y(3) - Y(4)

C     ----- COMPUTING SH_REGION, IF A CHECKED POINT IS IN THE               -----
C     ----- REGION, SH_REGION WILL BE ADDED WITH 1                          -----
      SH_REGION = 0
      A = CY*DX - DY*CX

      DO 20 I=1,4
         B = -4.*PY(I)*DX + AY*DX - BY*CX + CY*BX + DY*4.*PX(I) - DY*AX
         C = -4.*PY(I)*BX + AY*BX + 4.*PX(I)*BY - AX*BY

         IF( A.EQ.0 ) THEN
            IF( B.NE.0 ) THEN
               N = -C/B
               IF( N.LT.-1 .OR. N.GT.1 ) GO TO 20

               IF( (BX+DX*N).NE.0 ) THEN
                  E = (4.*PX(I)-AX-CX*N)/(BX+DX*N)
                  IF( E.LT.-1 .OR. E.GT.1 ) GO TO 20
                  SH_REGION = SH_REGION+1
               ELSE
                  GO TO 20
               END IF
            ELSE
               GO TO 20
            END IF
         ELSE
            DO 10 J=1,2
               N = B**2-4*A*C
               IF( N.LT.0 ) GO TO 20
               N = (-B+(-1)**(J)*N**.5)/2./A
               IF( N.LT.-1 .OR. N.GT.1 ) GO TO 10

               IF( (BX+DX*N).NE.0 ) THEN
                  E = (4.*PX(I)-AX-CX*N)/(BX+DX*N)
                  IF( E.LT.-1 .OR. E.GT.1 ) GO TO 10
                  SH_REGION = SH_REGION + 1
                  GO TO 20
               ELSE
                  GO TO 10
               END IF
   10       CONTINUE
         END IF
   20 CONTINUE

      RETURN
      END

      SUBROUTINE PHYCO_CHT(DBT,WBT,HR,MODE)
      REAL DBT,WBT,HR
      INTEGER MODE
C     =======================================================================
C     |                                                                     |
C     |       SUBROUTINE TO COMPUTE PROPERTIES OF AIR ALONG TO              |
C     |       DATA ON PHYCOMETRIC CHART :-                                  |
C     |            MODE            GIVEN            DETERMINED              |
C     |              1           WBT / HR              DBT                  |
C     |              2           DBT / HR              WBT                  |
C     |              3           DBT / WBT             HR                   |
C     |                                                                     |
C     |                          [ SI UNIT]                                 |
C     |                                                                     |
C     =======================================================================
```

```
C     @ ---- ASSIGN VARIABLES                                          @
C     @         A1,B1,C1        COEFFICIENTS FOR MODE-1 COMPUTING     @
C     @         A2,B2,C2        COEFFICIENTS FOR MODE-2 COMPUTING     @
C     @         A3,B3,C3        COEFFICIENTS FOR MODE-3 COMPUTING     @

      REAL    A1(3),B1(3),C1(3)
      REAL    A2(3),B2(3),C2(3)
      REAL    A3(3),B3(3),C3(3)

      DATA A1 / -7.46200E+00,   -2.27503E+02,    5.24570E+03 /
      DATA B1 /  1.72690E+00,   -4.26215E+01,   -2.23457E+03 /
      DATA C1 /  3.58000E-02,    1.60720E+00,    9.24510E+00 /

      DATA A2 / -9.04230E+00,    3.04500E-01,   -3.93150E-03 /
      DATA B2 /  9.07775E+02,   -3.54841E+01,    6.84700E-01 /
      DATA C2 / -2.97765E+04,    1.77963E+03,   -6.31000E+00 /

      DATA A3 / -1.52880E-10,   -1.79940E-07,    4.63050E-06 /
      DATA B3 /  5.23090E-07,   -1.21140E-05,   -4.50170E-04 /
      DATA C3 /  1.35440E-05,    7.34790E-04,    4.04220E-03 /

      IF( MODE.EQ.1 ) THEN
         DBT =         (A1(1)*WBT**2+A1(2)*WBT+A1(3))*HR**2
         DBT = DBT + (B1(1)*WBT**2+B1(2)*WBT+B1(3))*HR
         DBT = DBT + (C1(1)*WBT**2+C1(2)*WBT+C1(3))
      ELSE IF( MODE.EQ.2 ) THEN
         WBT =         (A2(1)*HR **2+A2(2)*HR +A2(3))*DBT**2
         WBT = WBT + (B2(1)*HR **2+B2(2)*HR +B2(3))*DBT
         WBT = WBT + (C2(1)*HR **2+C2(2)*HR +C2(3))
      ELSE IF( MODE.EQ.3 ) THEN
         HR  =         (A3(1)*WBT**2+A3(2)*WBT+A3(3))*DBT**2
         HR  = HR  + (B3(1)*WBT**2+B3(2)*WBT+B3(3))*DBT
         HR  = HR  + (C3(1)*WBT**2+C3(2)*WBT+C3(3))
      ENDIF

      RETURN
      END

C     ========== BELOW THIS LINE, ALL EITHER SUBROUTINES OR ===========
C     ========== FUNCTIONS ARE FOR MANAGING SCREEN,     NOT ===========
C     ========== INVOLVING TO SYMULATION.                 ===========

      SUBROUTINE SCREEN()
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE WILL PERPARE THE APPROPREATE SCREEN FOR    |
C     |    RUNNING THE PROGRAM.                                      |
C     |                                                              |
C     ================================================================

      CALL TCL(37,40)
      CALL CSC()
      CALL BLOCK(12,2,78,22,37,44,47)
      CALL BLOCK(7,12,24,15,30,43,40)
      CALL TCL(33,40)
      CALL XY(9,14)
      WRITE(6,*) 'DD | MM'
      CALL TCL(36,40)

      RETURN
      END

      SUBROUTINE BLOCK(TOP,LEFT,RIGHT,BOTTOM,CL1,CL2,CL3)
      INTEGER TOP,LEFT,RIGHT,BOTTOM
      INTEGER CL1,CL2,CL3
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE WILL CREATE A BLOCK WINDOW IN THE SCREEN.  |
C     |                                                              |
C     ================================================================


      CALL TCL(CL1,CL2)
      CALL TAT(1)
      DO 10 I=0,(RIGHT-LEFT),2
         CALL XY(TOP,LEFT+I)
         WRITE(6,'(A1)') CHAR(0)
         CALL XY(BOTTOM,LEFT+I)
         WRITE(6,'(A1)') CHAR(0)
   10 CONTINUE
      DO 20 I=TOP,BOTTOM,2
         CALL XY(I,LEFT)
         WRITE(6,'(A1)') CHAR(0)
         CALL XY(I,RIGHT)
         WRITE(6,'(A1)') CHAR(0)
   20 CONTINUE
      DO 30 I=1,(RIGHT-LEFT),2
         CALL XY(TOP,LEFT+I)
         WRITE(6,'(A1)') CHAR(177)
         CALL XY(BOTTOM,LEFT+I)
         WRITE(6,'(A1)') CHAR(177)
   30 CONTINUE
      DO 40 I=TOP+1,BOTTOM,2
         CALL XY(I,LEFT)
         WRITE(6,'(A1)') CHAR(177)
         CALL XY(I,RIGHT)
         WRITE(6,'(A1)') CHAR(177)
   40 CONTINUE

      CALL TCL(CL2,CL3)
      DO 50 I=TOP+1,BOTTOM-1
         DO 50 J=LEFT+1,RIGHT-1
            CALL XY(I,J)
            WRITE(6,'(A1)') CHAR(0)
```

```
   50 CONTINUE

      RETURN
      END

      SUBROUTINE TAT(IC)
      INTEGER IC
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE HELP USER TO SELECT TEXT ATTRIBUTE MODE.  |
C     |                0       : STANDARD TEXT                       |
C     |                1       : BOLD TOGGLE                         |
C     |                5       : BLINK MODE                          |
C     |                7       : INVERSE TEXT                        |
C     |                8       : CONCEALED TEXT                      |
C     |                                                              |
C     ================================================================
      CHARACTER*5 S
      CHARACTER*2 C

      CALL STR(C,IC)
      S = CHAR(27)//'['//C//'m'
      WRITE(6,'(1X,A5)') S

      RETURN
      END

      SUBROUTINE TCL(TC,BC)
      INTEGER TC,BC
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE HELP USER TO SELECT TEXT COLOR MODE.      |
C     |                                                              |
C     |                           TEXT BACKGROUND                    |
C     |                BLACK       30         40                    |
C     |                RED         31         41                    |
C     |                GREEN       32         42                    |
C     |                YELLOW      33         43                    |
C     |                BLUE        34         44                    |
C     |                MAGENTA     35         45                    |
C     |                CYAN        36         46                    |
C     |                WHITE       37         47                    |
C     |                                                              |
C     ================================================================
      CHARACTER*8 S
      CHARACTER*2 T,B

      CALL STR(T,TC)
      CALL STR(B,BC)
      S = CHAR(27)//'['//T//';'//B//'m'
      WRITE(6,'(1X,A8)') S

      RETURN
      END

      SUBROUTINE CSC()
C     ================================================================
C     |                                                              |
C     |            SUBROUTINE FOR CLEARING TEXT SCREEN               |
C     |                                                              |
C     ================================================================
      CHARACTER*4 S

      S = CHAR(27)//'['//'2'//'J'
      WRITE(6,'(1X,A4)') S

      RETURN
      END

      SUBROUTINE CLN()
C     ================================================================
C     |                                                              |
C     |            SUBROUTINE FOR CLEARING A LINE                    |
C     |                                                              |
C     ================================================================
      CHARACTER*3 S

      S = CHAR(27)//'['//'K'
      WRITE(6,'(1X,A3)') S

      RETURN
      END

      SUBROUTINE XY(IR,IC)
      INTEGER IR,IC
C     ================================================================
C     |                                                              |
C     | THIS SUBROUTINE LOCATES SPECIFIED ROW & COLUMN ON THE MONITOR |
C     |                                                              |
C     ================================================================
      CHARACTER*8 S
      CHARACTER*2 R,C

      CALL STR(R,IR)
      CALL STR(C,IC)
      S = CHAR(27)//'['//R//';'//C//'H'

      WRITE(6,10) S
   10 FORMAT( 1X, A8\ )

      RETURN
      END

      SUBROUTINE STR(X,IX)
```

```
      INTEGER IX
      CHARACTER*2 X
C     ================================================================
C     |                                                              |
C     |   THIS SUBROUTINE WILL TRANSLATE TWO-DECIMA INTERGER NUMBER  |
C     |                    INTO TWO-BITE CHARACTER.                  |
C     |                                                              |
C     ================================================================
      CHARACTER*2 CX
      CHARACTER   C(2)
      EQUIVALENCE (CX,C)

      INTEGER DUMMY

      DUMMY = IX/10
      C(1) = CHAR(DUMMY+48)
      C(2) = CHAR(48+(IX-DUMMY*10))
      X = CX

      RETURN
      END
```

SOURCE CODE โปรแกรม SYSTEM

```
C     ================================================================
C     |                                                              |
C     |  PROGRAM FOR SIMULATING ENERGY CONSUMED BY  SYSTEMS & PLANTS |
C     |           BY USING RESULTS FOR PROGRAM "CAL.EXE".            |
C     |                        [SI Unit]                             |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         CALENDAR        A FILE CONTAINS SCHEDULE OF HOLIDAY  @
C     @         CAP_MAX         MAXIMUM CAPACITY OF SYSTEMS of PLANTS @
C     @         CFM_MAX         MAXIMUM CFM OF SYSTEMS               @
C     @         DAY_T           TYPE OF DATE [0=SUN, 1=MON,..., 7=HOL]@
C     @         DAY_TYPE        DUMMY OF DAY_T                       @
C     @         DA              DAY THAT PROGRAM IS INTERESTED       @
C     @         DA_HOLIDAY      DAY OF HOLIDAY                       @
C     @         DA_WEATHER      DAY OF WEATHER DATA THAT IS TAKEN    @
C     @         DBT's           DRY BULB TEMPERATURE [degC]          @
C     @         DELTA_P         PRESSURE DROP IN DUCT SYSTEM (Pa)    @
C     @         ENERGY's        CONSUMED ENERGY (W)                  @
C     @         HR's            HUMIDITY RATIO                       @
C     @         I               COUNTER                              @
C     @         ID's            DIFFUSED SOLAR RADIATION [W/m^2]     @
C     @         IDN's           DIRECT NOEMAL SOLAR RADIATION [W/m^2] @
C     @         INPUTFILE       INPUTFILE OF SYSTEMS AND PLANTS      @
C     @         J               COUNTER                              @
C     @         LOADFILE        FILE THAT CONTAINS BUILDING LOADS    @
C     @         MO              MONTH THAT PROGRAM IS INTERESTED     @
C     @         MO_HOLIDAY      MONTH OF HOLIDAY                     @
C     @         MO_WEATHER      MONTH OF WEATHER DATA THAT IS TAKEN  @
C     @         NAME            NAME OF ZONES, SYSTEMS or PLANTS     @
C     @         NUM_PNT         NUMBER OF PLANT                      @
C     @         NUM_SCH         NUMBER OF SCHEDULE                   @
C     @         NUM_SYS         NUMBER OF SYSTEM                     @
C     @         NUM_S_SYS       NUMBER OF SERVED SYSTEM BY A PLANT   @
C     @         NUM_S_ZN        NUMBER OF SERVED ZONE BY A SYSTEM    @
C     @         NUM_ZONE        NUMBER OF ZONE                       @
C     @         PNT_NUM         PLANT NUMBER                         @
C     @         P_OUTPUT        OUTPUT FILE OF PLANT                 @
C     @         SCH_NUM's       REFERENT NUMBER OF SCHEDULE          @
C     @         STP_DD          STOPPING DAY OF SCHEDULE             @
C     @         STP_MM          STOPPING MONTH OF SCHEDULE           @
C     @         STR_DD          STARTING DAY OF SCHEDULE             @
C     @         STR_MM          STARTING MONTH OF SCHEDULE           @
C     @         SYS_NUM         SYSTEM NUMBER                        @
C     @         S_OUTPUT        OUTPUT FILE OF SYSTEM                @
C     @         TYPE            TYPE OF SYSTEMS or PLANTS            @
C     @         WBT's           OUTDOOR WET BULB TEMPERATURE [degC]  @
C     @         WD's            WIND DIRECTION                       @
C     @         WEATHER         INPUTFILE OF WEATHER DATA            @
C     @         WSP's           WIND SPEED                           @
C     @         YR              YEAR THAT PROGRAM IS INTERESTED      @
C     @         YR_WEATHER      YEAR OF WEATHER DATA THAT IS TAKEN   @
C     @         ZONE_NUM        ZONE NUMBER                          @

      CHARACTER*80 LINE
      CHARACTER*40 INPUTFILE,LOADFILE,S_OUTPUT,P_OUTPUT,CALENDAR,WEATHER
      CHARACTER*20 NAME,TYPE
      INTEGER      NUM_SCH
      INTEGER      STR_DD,STR_MM,STP_DD,STP_MM
      INTEGER      SCH_NUM(8)
      INTEGER      DA,MO,YR
      INTEGER      DA_WEATHER,MO_WEATHER,YR_WEATHER
      INTEGER      DA_HOLIDAY,MO_HOLIDAY,DAY_T,DAY_TYPE
      INTEGER      NUM_ZONE,NUM_SYS,NUM_PNT
      INTEGER      ZONE_NUM,SYS_NUM,PNT_NUM
      INTEGER      NUM_S_ZN,NUM_S_SYS
      INTEGER      I,J
      REAL         SCHEDULE(24)
      REAL         DUM(24)
      REAL         CAP_MAX,CFM_MAX,DELTA_P
      REAL         DBT(24),WBT(24),IDN(24),ID(24),HR(24),WSP(24),WD(24)
      REAL         ENERGY(24)
      REAL         DUMMY

C     ----- READ INPUT FILENAME          ---> INPUTFILE             -----
C     -----      FILE OF BUILDING LOADS ---> LOADFILE              -----
C     -----      WEATHER DATA FILENAME  ---> WEATHER               -----
C     -----      CALENDAR FILENAME      ---> CALENDAR              -----
C     -----      OUTPUT FILENAMES       ---> S_OUTPUT, P_OUTPUT    -----
   10 WRITE(6,1000)
      READ(5,'(A)',ERR=20) INPUTFILE,LOADFILE,WEATHER,CALENDAR,
     +                     S_OUTPUT,P_OUTPUT
      GO TO 30
   20 WRITE(6,1001)
      GO TO 10
   30 CONTINUE

C     ----- PARSING INPUT DATA FILE INTO RETAIL FILES              -----
      CALL SCREEN()
      CALL PARSE(INPUTFILE)

C     ----- OPEN FILES TO BE READY FOR BEING UTILIZED.             -----
      OPEN(2,FILE=LOADFILE)
      OPEN(3,FILE=WEATHER)
      OPEN(4,FILE=CALENDAR)
      OPEN(12,FILE=S_OUTPUT)
```

```
      OPEN(13,FILE=P_OUTPUT)

C     ----- WRITE HEADING OF OUTPUT FILES INTO THOSE FILES        -----
      OPEN(14,FILE='HEADING.TMP')
      WRITE(12,'(A)') 'SYSTEM LOAD'
      READ(14,*) NUM_SYS
      WRITE(12,*) NUM_SYS
      DO 32 I=1,NUM_SYS
         READ(14,'(A)') NAME
         WRITE(12,'(A)') NAME
   32 CONTINUE

      WRITE(13,'(A)') 'PLANT LOAD'
      READ(14,*) NUM_PNT
      WRITE(13,*) NUM_PNT
      DO 38 I=1,NUM_PNT
         READ(14,'(A)') NAME
         WRITE(13,'(A)') NAME
   38 CONTINUE
      CLOSE(14)

C     ----- READ HEADING DATA OF LOADFILE                         -----
      READ(2,'(A)') LINE
      IF( LINE.NE.'BUILDING LOADS' ) GO TO 6010

      READ(2,*) NUM_ZONE
      DO 40 I=1,NUM_ZONE
   40    READ(2,'(A)') NAME
      DO 50 I=1,NUM_ZONE
   50    READ(2,*) DUMMY

C     ----- READ HEADING DATA OF WEATHER DATA and CALENDAR FILE   -----
      READ(3,'(A)') LINE
      READ(4,*) DAY_T
      DAY_T = DAY_T-1
      READ(4,*,END=55,ERR=56) DA_HOLIDAY,MO_HOLIDAY
   54 GO TO 57
   55 MO_HOLIDAY = 13
      GO TO 57
   56 CALL WARNNING(1,0,0,0)
   57 CONTINUE

C     ----- READ DATA OF LOADS, THEN KEEP ONLY HEATEXTRACTION RATE-----
C     ----- INTO A FILE, CALLED ZONE_L.TMP, UNTIL END OF LOADFILE -----
   60 OPEN(10,FILE='ZONE_L.TMP',ACCESS='DIRECT',RECL=10)
      READ(2,*,ERR=6020,END=290) DA,MO,YR
      WRITE(12,*) DA,MO,YR
      WRITE(13,*) DA,MO,YR

      DO 100 I=1,NUM_ZONE
         DO 70 J=1,14
   70       READ(2,*,ERR=6020,END=6020) DUM
         IDUMMY = (I-1)*24
         DO 80 J=1,24
   80       WRITE(10,REC=(IDUMMY+J)) DUM(J)
  100 CONTINUE
      CLOSE(10)

C     ----- SHOW THE CURRENT DATE THAT THE PROGRAM'S SYMULATING.  -----
      CALL TAT(5)
      CALL TCL(33,40)
      CALL XY(11,15)
      WRITE(6,'(I2,A3,I2)') DA,' | ',MO
      CALL TAT(0)
      CALL TAT(1)
      CALL TCL(36,40)

C     ----- READ WEATHER DATA and CALENDAR FILE, COINCIDENT TO THE-----
C     ----- DATE, READ ABOVE.                                     -----
  110 READ(3,*,ERR=6020,END=6030) DA_WEATHER,MO_WEATHER,YR_WEATHER
      READ(3,*,ERR=6020,END=6030) DBT,WBT,IDN,ID,HR,WSP,WD
      DAY_T = DAY_T+1
      IF( DAY_T.GE.7 ) DAY_T = 0
      DAY_TYPE = DAY_T
      IF( DA.NE.DA_WEATHER .OR. MO.NE.MO_WEATHER ) GO TO 110

  120 IF( MO_HOLIDAY.LT.MO ) THEN
         READ(4,*,END=135,ERR=140) DA_HOLIDAY,MO_HOLIDAY
         GO TO 120
      ELSE IF( MO_HOLIDAY.EQ.MO ) THEN
         IF( DA_HOLIDAY.LT.DA ) THEN
            READ(4,*,END=135,ERR=140) DA_HOLIDAY,MO_HOLIDAY
            GO TO 120
         ELSE IF( DA_HOLIDAY.EQ.DA ) THEN
            DAY_TYPE = 7
         END IF
      END IF
  130 GO TO 150
  135 MO_HOLIDAY = 13
      GO TO 150
  140 CALL WARNNING(1,0,0,0)
  150 CONTINUE

C     ----- COMPUTING SYSTEM LOAD                                 -----
      SYS_NUM = 0
      OPEN(8,FILE='SYSTEM.TMP')

  155 SYS_NUM = SYS_NUM+1
      READ(8,'(A)') TYPE
      READ(8,*) CAP_MAX,CFM_MAX,DELTA_P

      READ(8,*) NUM_SCH
      OPEN(7,FILE='SCHEDULE.TMP',ACCESS='DIRECT',RECL=10)
      DO 170 I=1,NUM_SCH
```

```
      READ(8,*) STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
      IF(WITHIN(STR_DD,STR_MM,STP_DD,STP_MM,DA,MO).EQ.0) THEN
         IDUMMY = (SCH_NUM(DAY_TYPE+1)-1)*24
         DO 160 J=1,24
160         READ(7,REC=(IDUMMY+J),ERR=6040,END=6040) SCHEDULE(J)
      END IF
170 CONTINUE
    CLOSE(7)

    DO 180 I=1,24
180    DUM(I) = 0.0

    READ(8,*) NUM_S_ZN
    OPEN(10,FILE='ZONE_L.TMP',ACCESS='DIRECT',RECL=10)
    DO 200 I=1,NUM_S_ZN
       READ(8,*) ZONE_NUM
       IDUMMY = (ZONE_NUM-1)*24
       DO 190 J=1,24
          READ(10,REC=(IDUMMY+J),END=6050) DUMMY
          DUM(J) = DUM(J) + DUMMY
190    CONTINUE
200 CONTINUE
    CLOSE(10)

    OPEN(11,FILE='SYSTEM_L.TMP',ACCESS='DIRECT',RECL=10)
    IDUMMY = (SYS_NUM-1)*24
    DO 210 I=1,24
       IF( DUM(I).LE.(SCHEDULE(I)*CAP_MAX) ) THEN
          WRITE(11,REC=(IDUMMY+I)) DUM(I)
       ELSE
          GO TO 6060
       END IF
210 CONTINUE
    CLOSE(11)

    CALL SYSTEM(TYPE,CAP_MAX,CFM_MAX,DELTA_P,SCHEDULE,DUM,DBT,WBT,
   +            ENERGY)
    WRITE(12,*) DUM,ENERGY

    IF( SYS_NUM.LT.NUM_SYS ) GO TO 155
    CLOSE(8)

C   ----- COMPUTING PLANT LOAD                                    -----
    PNT_NUM = 0
    OPEN(9,FILE='PLANT.TMP')

220 PNT_NUM = PNT_NUM+1
    READ(9,'(A)') TYPE
    READ(9,*) CAP_MAX

    READ(9,*) NUM_SCH
    OPEN(7,FILE='SCHEDULE.TMP',ACCESS='DIRECT',RECL=10)
    DO 240 I=1,NUM_SCH
       READ(9,*) STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
       IF(WITHIN(STR_DD,STR_MM,STP_DD,STP_MM,DA,MO).EQ.0) THEN
          IDUMMY = (SCH_NUM(DAY_TYPE+1)-1)*24
          DO 230 J=1,24
230          READ(7,REC=(IDUMMY+J),ERR=6040,END=6040) SCHEDULE(J)
       END IF
240 CONTINUE
    CLOSE(7)

    DO 250 I=1,24
250    DUM(I) = 0.0

    READ(9,*) NUM_S_SYS
    OPEN(11,FILE='SYSTEM_L.TMP',ACCESS='DIRECT',RECL=10)
    DO 270 I=1,NUM_S_SYS
       READ(9,*) SYS_NUM
       IDUMMY = (SYS_NUM-1)*24
       DO 260 J=1,24
          READ(11,REC=(IDUMMY+J),END=6050) DUMMY
          DUM(J) = DUM(J) + DUMMY
260    CONTINUE
270 CONTINUE
    CLOSE(11)

    DO 280 I=1,24
       IF( DUM(I).GT.(SCHEDULE(I)*CAP_MAX) ) GO TO 6070
280 CONTINUE

    CALL PLANT(TYPE,CAP_MAX,SCHEDULE,DUM,DBT,WBT,ENERGY)
    WRITE(13,*) DUM,ENERGY

    IF( PNT_NUM.LT.NUM_PNT ) GO TO 220
    CLOSE(9)

C   ----- RETURN TO THE NEXT DATE                                 -----
    GO TO 60

C   ----- FINISH COMPUTATION                                      -----
290 CONTINUE
    CLOSE(3)
    CLOSE(4)
    CLOSE(12)
    CLOSE(13)

    CALL TAT(0)
    CALL TAT(1)
    CALL TCL(36,40)
    STOP

1000 FORMAT(1X,  'ENTER RELATIVELY FILENAME OF >', /,
   +       T20, '1. INPUT FILE', /,
```

```
     +       T20, '2. FILE OF BUILDING LOADS', /,
     +       T20, '3. WEATHER DATA', /,
     +       T20, '4. CALENDAR FILE', /,
     +       T20, '5. SYSTEM OUTPUT FILE', /,
     +       T20, '6. PLANT OUTPUT FILE')
 1001 FORMAT(1X, 'MESSAGE ERRORS :: PLEASE RE-ENTER')

 6000 CALL ERROR(1,0,0,0)
 6010 CALL ERROR(2,0,0,0)
 6020 CALL ERROR(3,0,0,0)
 6030 CALL ERROR(4,0,0,0)
 6040 CALL ERROR(5,0,0,0)
 6050 CALL ERROR(6,SYS_NUM,ZONE_NUM,0)
 6060 CALL ERROR(7,SYS_NUM,0,0)
 6070 CALL ERROR(8,PNT_NUM,0,0)

      END

      FUNCTION WITHIN(SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA,MO)
      INTEGER SCH_STRD,SCH_STRM,SCH_STPD,SCH_STPM,DA,MO
C     ================================================================
C     |                                                              |
C     |    FUNCTION FOR INDICATING IF THE DATE DA/MO IS WITHIN THE   |
C     |   PERIOD LIMITED BY SCH_STRD/SCH_STRM AND SCH_STPD/SCH_STPM. |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         DATE            BRIEFLY CALCULATED DAY No. OF DA/MO  @
C     @         UPPER_L         BRIEFLY CALCULATED DAY No. OF SCH_STP @
C     @         LOWER_L         BRIEFLY CALCULATED DAY No. OF SCH_STR @

      INTEGER DATE,UPPER_L,LOWER_L

C     ----- BY ESTIMATING, 1 MONTH EQUALS 30 DAYS, WILL OBTAIN.   -----
      DATE    = DA       + MO*30
      UPPER_L = SCH_STPD + SCH_STPM*30
      LOWER_L = SCH_STRD + SCH_STRM*30

C     ----- WITHIN = -1,0 or 1, IF DATE IS LOWER, WITHIN or ABOVE -----
C     ----- THE CONSTRAINT, SPECIFIED BY UPPER_L & LOWER_L, RES. .-----
      IF( DATE.LT.LOWER_L ) THEN
         WITHIN = -1
      ELSE IF( DATE.GT.UPPER_L ) THEN
         WITHIN = 1
      ELSE
         WITHIN = 0
      END IF

      RETURN
      END

      SUBROUTINE PARSE(INPUTFILE)
      CHARACTER*40 INPUTFILE
C     ================================================================
C     |                                                              |
C     |          PARSING DATA IN INPUTFILE INTO RETAIL FILES         |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         CAP_MAX         MAXIMUM CAPACITY OF SYSTEMS of PLANTS @
C     @         CFM_MAX         MAXIMUM CFM OF SYSTEMS               @
C     @         I               COUNTER                              @
C     @         J               COUNTER                              @
C     @         NAME            NAME OF ZONES, SYSTEMS or PLANTS     @
C     @         NUM_PNT         NUMBER OF PLANT                      @
C     @         NUM_SCH         NUMBER OF SCHEDULE                   @
C     @         NUM_SYS         NUMBER OF SYSTEM                     @
C     @         NUM_S_SYS       NUMBER OF SERVED SYSTEM BY A PLANT   @
C     @         NUM_S_ZN        NUMBER OF SERVED ZONE BY A SYSTEM    @
C     @         SCH_NUM's       REFERENT NUMBER OF SCHDULE           @
C     @         STP_DD          STOPPING DAY OF SCHEDULE             @
C     @         STP_MM          STOPPING MONTH OF SCHEDULE           @
C     @         STR_DD          STARTING DAY OF SCHEDULE             @
C     @         STR_MM          STARTING MONTH OF SCHEDULE           @
C     @         SYS_NUM         SYSTEM NUMBER                        @
C     @         TYPE            TYPE OF SYSTEMS or PLANTS            @
C     @         ZONE_NUM        ZONE NUMBER                          @

      CHARACTER*20 NAME,TYPE
      INTEGER      NUM_SCH
      INTEGER      STR_DD,STR_MM,STP_DD,STP_MM
      INTEGER      SCH_NUM (8)
      INTEGER      NUM_SYS,NUM_PNT
      INTEGER      ZONE_NUM,SYS_NUM
      INTEGER      NUM_S_ZN,NUM_S_SYS
      INTEGER      I,J
      REAL         SCHEDULE(24)
      REAL         CAP_MAX,CFM_MAX

C     ----- OPEN FILE INPUTFILE                                    -----
      OPEN(1,FILE=INPUTFILE)
      CALL READ_COMM()                                                *new*

C     ----- PARSING REFERENT SCHEDULES INTO FILE SCHEDULE.TMP      -----
      OPEN(7,FILE='SCHEDULE.TMP',ACCESS='DIRECT',RECL=10)
      CALL READ_COMM()                                                *new*
      READ(1,*,ERR=6000,END=6000) NUM_SCH
      DO 10 I=1,NUM_SCH
         READ(1,*,ERR=6000,END=6000) SCHEDULE
         IDUMMY = (I-1)*24
         DO 10 J=1,24
            WRITE(7,REC=(IDUMMY+J)) SCHEDULE(J)
```

```
   10 CONTINUE
      CLOSE(7)

C     ----- PARSING SYSTEM DATA INTO TEMP-FILE HEADING and SYSTEM -----
      OPEN(14,FILE='HEADING.TMP')
      OPEN(8,FILE='SYSTEM.TMP')

      CALL READ_COMM()                                                  *new*
      READ(1,*,ERR=6010,END=6010) NUM_SYS
      WRITE(14,*) NUM_SYS
      DO 20 I=1,NUM_SYS
         READ(1,'(A)',ERR=6010,END=6010) NAME
         WRITE(14,'(A)') NAME
   20 CONTINUE

      CALL READ_COMM
      DO 50 I=1,NUM_SYS                                                 *new*
   25    READ(1,'(A)',ERR=6020,END=6020) TYPE
         IF( TYPE.EQ.'                    ' ) GO TO 25
         WRITE(8,'(A)') TYPE
         READ(1,*,ERR=6020,END=6020) CAP_MAX,CFM_MAX,DELTA_P
         WRITE(8,*) CAP_MAX,CFM_MAX,DELTA_P

         READ(1,*,ERR=6020,END=6020) NUM_SCH
         WRITE(8,*) NUM_SCH
         DO 30 J=1,NUM_SCH
            READ(1,*,ERR=6020,END=6020)
     +              STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
            WRITE(8,*) STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
   30    CONTINUE

         READ(1,*,ERR=6020,END=6020) NUM_S_ZN
         WRITE(8,*) NUM_S_ZN
         DO 40 J=1,NUM_S_ZN
            READ(1,*,ERR=6020,END=6020) ZONE_NUM
            WRITE(8,*) ZONE_NUM
   40    CONTINUE
   50 CONTINUE
      CLOSE(8)

C     ----- PARSING PLANT DATA INTO TEMP-FILE HEADING and PLANT   -----
      OPEN(9,FILE='PLANT.TMP')

      CALL READ_COMM()                                                  *new*
      READ(1,*,ERR=6030,END=6030) NUM_PNT
      WRITE(14,*) NUM_PNT
      DO 60 I=1,NUM_PNT
         READ(1,'(A)',ERR=6030,END=6030) NAME
         WRITE(14,'(A)') NAME
   60 CONTINUE

      CALL READ_COMM()                                                  *new*
      DO 90 I=1,NUM_PNT
   65    READ(1,'(A)',ERR=6040,END=6040) TYPE
         IF( TYPE.EQ.'                    ' ) GO TO 65
         WRITE(9,'(A)') TYPE
         READ(1,*,ERR=6040,END=6040) CAP_MAX
         WRITE(9,*) CAP_MAX

         READ(1,*,ERR=6040,END=6040) NUM_SCH
         WRITE(9,*) NUM_SCH
         DO 70 J=1,NUM_SCH
            READ(1,*,ERR=6040,END=6040)
     +              STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
            WRITE(9,*) STR_DD,STR_MM,STP_DD,STP_MM,SCH_NUM
   70    CONTINUE

         READ(1,*,ERR=6040,END=6040) NUM_S_SYS
         WRITE(9,*) NUM_S_SYS
         DO 80 J=1,NUM_S_SYS
            READ(1,*,ERR=6040,END=6040) SYS_NUM
            WRITE(9,*) SYS_NUM
   80    CONTINUE
   90 CONTINUE
      CLOSE(9)
      CLOSE(14)

      RETURN

 6000 CALL ERROR(9,0,0,0)
 6010 CALL ERROR(10,0,0,0)
 6020 CALL ERROR(11,I,0,0)
 6030 CALL ERROR(12,0,0,0)
 6040 CALL ERROR(13,I,0,0)
      END

      SUBROUTINE SYSTEM(TYPE,CAP_MAX,CFM_MAX,DELTA_P,SCHEDULE,DUM,DBT,
     +                  WBT,ENERGY)
      CHARACTER*20 TYPE
      REAL         CAP_MAX,CFM_MAX,DELTA_P
      REAL         SCHEDULE(24),DUM(24),DBT(24),WBT(24),ENERGY(24)
C     ================================================================
C     |                                                              |
C     |        THIS WILL COMPUTE ENERGY CONSUMPTION OF SYSTEMS       |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         APPROACH        COOLED WATER TEMP - WBT [degC]     @
C     @         BHP             BREAK HORSE POWER [HP]             @
C     @         CAP_MAX1        CORRECTED MAXIMUM CAPACITY [W]     @
C     @         CDS_TEMP        CONDENSING TEMPERATURE [degC]      @
C     @         CFM             CFM OF SUPPLY AIR                  @
C     @         COP             COP OF AIR CONDITIONER             @
```

```
C     @         EFF             EFFICIENT OF BROWER                    @
C     @         I               COUNTER                                @
C     @         MAX_FAC         MAX FACTOR AT INDIVIDUAL CDS_TEMP      @
C     @         NORM_CAP        NORMINAL CAPACITY [W]                  @
C     @         PERCENT         PERCENT LOAD                           @
C     @         RANGE           CONDENSING TEMP - COOLED WATER TEMP    @
C     @         SP              STATIC PRESSURE [INCH-WATER]           @

      INTEGER       I
      REAL          CDS_TEMP,APPROACH,MAX_FAC
      REAL          NORM_CAP
      REAL          RANGE,CAP_MAX1
      REAL          CFM,COP,PERCENT,EFF,SP

C     ----- COMPUTING ENERGY CONSUMPTION OF SPECIFIED TYPE OF     -----
C     ----- SYSTEM.                                               -----

C     ::::::::::              SPLIT TYPE                  ::::::::::
      IF( TYPE.EQ.'SPLIT TYPE' .OR. TYPE.EQ.'Split type' .OR.
     +    TYPE.EQ.'Split Type' .OR. TYPE.EQ.'split type' ) THEN

         DO 10 I=1,24
            CFM = CFM_MAX*SCHEDULE(I)
            ENERGY(I) = (6.4381E-8*CFM*DELTA_P+1.22322)*CFM*DELTA_P     [fan]
            ENERGY(I) = ENERGY(I) + 1.7463E-8*CFM**2 + 5.8594E-5*CFM    [fan]
            EFF       = 3.432E-10*ENERGY(I)**2+1.089E-5*ENERGY(I)+0.784 [fan]
            IF( EFF.GT.0.87 ) EFF = 0.87                                [fan]
            ENERGY(I) = ENERGY(I)/EFF                                   [fan]

            IF( DUM(I).EQ.0.0 ) THEN                                    [cds]
               ENERGY(I) = 0.0                                          [cds]
            ELSE                                                        [cds]
               PERCENT   =  DUM(I)/(CAP_MAX*SCHEDULE(I))                [cds]
               COP       =  (-0.1213235*DBT(I) + 8.9338235)*PERCENT     [cds]
     +                    + ( 0.0331985*DBT(I) - 2.0900735)            [cds]
               IF( COP.GT.4.2 ) COP = 4.2                               [cds]
               ENERGY(I) = ENERGY(I) + DUM(I)/COP                       [cds]
            END IF                                                      [cds]
   10    CONTINUE
         RETURN
      END IF
C     ================================================================

C     ::::::::::                FAN COIL                  ::::::::::
      IF( TYPE.EQ.'FAN COIL' .OR. TYPE.EQ.'Fan coil' .OR.
     +    TYPE.EQ.'Fan Coil' .OR. TYPE.EQ.'fan coil' ) THEN

         DO 20 I=1,24
            CFM = CFM_MAX*SCHEDULE(I)
            ENERGY(I) = -5.661E-10*CFM**4 + 1.568E-6*CFM**3
            ENERGY(I) = ENERGY(I) - 1.266E-3*CFM**2 + 5.551E-1
   20    CONTINUE
         RETURN
      END IF
C     ================================================================

C     ::::::::::      AIR HANDLING UNIT (LOW PRESSURE)      ::::::::::
      IF(TYPE.EQ.'AHU LOW PRESSURE'.OR.TYPE.EQ.'AHU Low pressure' .OR.
     +   TYPE.EQ.'AHU Low Pressure'.OR.TYPE.EQ.'AHU low pressure') THEN

         SP = DELTA_P/9.81/998/0.305*12.0
         DO 30 I=1,24
            CFM = CFM_MAX*SCHEDULE(I)
            ENERGY(I) = 2.781e-6*CFM**1.527*SP**4.357E-1
            ENERGY(I) = ENERGY(I)*746.0
   30    CONTINUE
         RETURN
      END IF
C     ================================================================

C     ::::::::::      AIR HANDLING UNIT (HIGH PRESSURE)     ::::::::::
      IF(TYPE.EQ.'AHU HIGH PRESSURE'.OR.TYPE.EQ.'AHU High pressure' .OR.
     +  TYPE.EQ.'AHU High Pressure'.OR.TYPE.EQ.'AHU high pressure') THEN

         SP = DELTA_P/9.81/998/0.305*12.0
         DO 40 I=1,24
            CFM = CFM_MAX*SCHEDULE(I)
            ENERGY(I) = 8.350E-6*CFM**1.416*SP**6.471E-1
            ENERGY(I) = ENERGY(I)*746.0
   40    CONTINUE
         RETURN
      END IF
C     ================================================================

C     ::::::::::        AIR HANDLING UNIT (AIR FOIL)        ::::::::::
      IF(TYPE.EQ.'AHU AIR FOIL' .OR. TYPE.EQ.'AHU Air foil' .OR.
     +   TYPE.EQ.'AHU Air Foil' .OR. TYPE.EQ.'AHU air foil') THEN

         SP = DELTA_P/9.81/998/0.305*12.0
         DO 50 I=1,24
            CFM = CFM_MAX*SCHEDULE(I)
            ENERGY(I) = 7.698E-9*CFM**2.166*SP**6.062E-1
            ENERGY(I) = ENERGY(I)*746.0
   50    CONTINUE
         RETURN
      END IF
C     ================================================================

C     ::::::::::            WATER COOLED PACKAGE          ::::::::::
      IF( TYPE.EQ.'WATER COOLED PACKAGE' .OR.
     +    TYPE.EQ.'Water cooled package' .OR.
     +    TYPE.EQ.'Water Cooled Package' .OR.
     +    TYPE.EQ.'water cooled package' ) THEN
```

```
      SP = DELTA_P/9.81/998/0.305*12.0
      DO 70 I=1,24

         APPROACH = 29.44 - WBT(I)                                    [cds]
         MAX_FAC = -0.0342*(WBT(I)/5*9+32)+4.2758                     [cds]
         CDS_TEMP = 33.0                                              [cds]
60       RANGE = (CDS_TEMP-29.44)                                     [cds]
         FACTOR = 4.166E-6*APPROACH**2-1.1E-4*APPROACH+9.357E-4       [cds]
         FACTOR = FACTOR*((RANGE*9/5)**2-1.306E2*RANGE+1.837E3)       [cds]
         FACTOR = FACTOR*MAX_FAC                                      [cds]
         NORM_CAP = CAP_MAX*SCHEDULE(I)*RANGE*9/50                    [cds]
         IF( (NORM_CAP/FACTOR).LT.DUM(I) ) THEN                       [cds]
            CDS_TEMP = CDS_TEMP + 1.0                                 [cds]
            GO TO 60                                                  [cds]
         END IF                                                       [cds]

         CDS_TEMP = CDS_TEMP*9.0/5.0+32.0                             [cds]
         CAP_MAX1 = (-0.00365*CDS_TEMP+1.3287)*CAP_MAX*SCHEDULE(I)    [cds]

         PERCENT = DUM(I)/CAP_MAX1                                    [cds]
         IF( PERCENT.LT.0.85 ) PERCENT = 0.85                         [cds]
         COP      = (1.909e-4*PERCENT+3.011E-5)                       [cds]
         COP      = COP*(CDS_TEMP**2-4.665E2*CDS_TEMP+5.516E4)        [cds]
         ENERGY(I)= DUM(I)/COP                                        [cds]

         CFM = CFM_MAX*SCHEDULE(I)                                    [fan]
         BHP      = 1.707E-9*CFM**2.1332*(SP+2.5567)**1.3477          [fan]
         IF( BHP.LT.134 ) THEN                                        [fan]
            BHP = BHP*1.2                                             [fan]
         ELSE                                                         [fan]
            BHP = BHP*1.15                                            [fan]
         END IF                                                       [fan]
         ENERGY(I) = ENERGY(I) + BHP*746.0                            [fan]

70    CONTINUE
      RETURN
   END IF
C  ================================================================
   CALL ERROR(14,0,0,0)

   END

   SUBROUTINE PLANT(TYPE,CAP_MAX,SCHEDULE,DUM,DBT,WBT,ENERGY)
   CHARACTER*20 TYPE
   REAL         CAP_MAX
   REAL         SCHEDULE(24),DUM(24),DBT(24),WBT(24),ENERGY(24)
C  ================================================================
C  |                                                              |
C  |         THIS WILL COMPUTE ENERGY CONSUMPTION OF PLANT        |
C  |                                                              |
C  ================================================================

C  @ ---- ASSIGN VARIABLES                                        @
C  @         CDS_TEMP        CONDENSING TEMPERATURE [degC]      @
C  @         APPROACH        COOLED WATER TEMP - WBT [degC]     @
C  @         MAX_FAC         MAX FACTOR AT INDIVIDUAL CDS_TEMP  @
C  @         I               COUNTER                            @
C  @         PERCENT_LOAD    PERCENT LOAD                       @
C  @         COP             COP                                @
C  @         NORM_CAP        NORMINAL CAPACITY [W]              @
C  @         RANGE           CONDENSING TEMP - COOLED WATER TEMP @
C  @         CAP_MAX1        CORRECTED MAXIMUM CAPACITY [W]     @

   INTEGER      I
   REAL         CDS_TEMP,APPROACH,MAX_FAC,PERCENT_LOAD,COP
   REAL         NORM_CAP
   REAL         RANGE,CAP_MAX1

C  ----- COMPUTING ENERGY CONSUMPTION OF SPECIFIED TYPE OF     -----
C  ----- SYSTEM.                                              -----

C  :::::::::::   WATER COOLED CHILLER WITH COOLING TOWER   :::::::::::
   IF( TYPE.EQ.'WATER COOLED CHILLER' .OR.
  +    TYPE.EQ.'Water cooled chiller' .OR.
  +    TYPE.EQ.'Water Cooled Chiller' .OR.
  +    TYPE.EQ.'water cooled chiller' ) THEN

      DO 20 I=1,24

         APPROACH = 29.44 - WBT(I)
         MAX_FAC = -0.0342*(WBT(I)/5*9+32)+4.2758
         CDS_TEMP = 33.0
10       RANGE = (CDS_TEMP-29.44)
         FACTOR = 4.166E-6*APPROACH**2-1.1E-4*APPROACH+9.357E-4
         FACTOR = FACTOR*((RANGE*9/5)**2-1.306E2*RANGE+1.837E3)
         FACTOR = FACTOR*MAX_FAC
         NORM_CAP = CAP_MAX*SCHEDULE(I)*RANGE*9/50
         IF( (NORM_CAP/FACTOR).LT.DUM(I) ) THEN
            CDS_TEMP = CDS_TEMP + 1.0
            GO TO 10
         END IF

         CDS_TEMP = CDS_TEMP*9.0/5.0+32.0
         CAP_MAX1 = (-0.0051*CDS_TEMP+1.4353)*CAP_MAX*SCHEDULE(I)

         PERCENT_LOAD = DUM(I)/CAP_MAX1
         IF( PERCENT_LOAD.LT.0.83 ) PERCENT_LOAD = 0.83
         COP      = (5.099E-4*PERCENT_LOAD+1.456E-4)
         COP      = COP*(CDS_TEMP**2-2.92E2*CDS_TEMP+2.606E4)
         ENERGY(I)= DUM(I)/COP

20    CONTINUE
      RETURN
```

```
      END IF
C     ================================================================
      CALL ERROR(15,0,0,0)

      END

      SUBROUTINE READ_COMM()
C     ================================================================
C     |                                                              |
C     | THIS SUBROUTINE WILL ELIMINATE THE COMMENT ON THE INPUT FILE. |
C     | IT READ "<" AS A BEGINING SIGN OF COMMENT AND ">" AS THE END. |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         FIRST           THE FIRST CHARACTER OF LINE        @

      CHARACTER    FIRST

    5 READ(1,'(A)',END=6000) FIRST
      IF( FIRST.EQ.' ' ) GO TO 5
      IF( FIRST.NE.'<' ) GO TO 6000

   10 READ(1,'(A)',END=6010) FIRST
      IF( FIRST.NE.'>' ) GO TO 10

      RETURN

 6000 CALL ERROR(71,0,0,0)
 6010 CALL ERROR(72,0,0,0)
      END


      SUBROUTINE WARNNING(IA,IB,IC,ID)
      INTEGER IA,IB,IC,ID
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE WILL WARN USERS ABOUT IMPROPER MATTERS     |
C     |    IN THE PROGRAM.                                           |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         INDENT          INDENT OF PRINTED LINES            @
C     @         LINE            THE LINE OF THE FIRST SENTENCE     @

      INTEGER LINE,INDENT

      LINE = 17
      INDENT = 8

      CALL TAT(0)
      CALL TCL(32,47)
      CALL XY(LINE,INDENT)

      IF( IA.EQ.1 ) THEN
         WRITE(6,*) 'CALENDAR FILE :: Illegal FORMAT.'
      END IF

      RETURN
      END

      SUBROUTINE ERROR(IA,IB,IC,ID)
      INTEGER IA,IB,IC,ID
C     ================================================================
C     |                                                              |
C     |       THIS SUBROUTINE WILL SHOW HOW ERROR OCCURS IN THE      |
C     |    PROGRAM THEN STOP THE PROGRAM FOR AVOIDING ANY SERIOUS    |
C     |    EFFECT, RESULTS FROM THAT ERROR.                          |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         INDENT          INDENT OF PRINTED LINES            @
C     @         LINE            THE LINE OF THE FIRST SENTENCE     @

      INTEGER LINE,INDENT

      LINE = 17
      INDENT = 8

      CALL TAT(0)
      CALL TCL(31,47)
      CALL XY(LINE,INDENT)

      IF( IA.EQ.1 ) THEN
         WRITE(6,*) 'BUILDING-LOAD FILE is WRONG in FORMAT.'

      ELSE IF( IA.EQ.2 ) THEN
         WRITE(6,*) 'READING ERROR IN BUILDING-LOAD FILE'

      ELSE IF( IA.EQ.3 ) THEN
         WRITE(6,*) 'READING ERROR IN WEATHER DATA FILE'

      ELSE IF( IA.EQ.4 ) THEN
         WRITE(6,*) 'DATE in WEATHER DATA :: NOT MATCH TO LOAD FILE'

      ELSE IF( IA.EQ.5 ) THEN
         WRITE(6,*) 'SPECIFIED WRONG SCHEDULE No. :: OUT OF FILE'

      ELSE IF( IA.EQ.6 ) THEN
         WRITE(6,*) 'SPECIFIED WRONG ZONE No. :: ',IC,' AT SYSTEM ',IB
```

```
      ELSE IF( IA.EQ.7 ) THEN
         WRITE(6,*) 'LOAD IS EXCEED SYSTEM CAPACITY AT SYSTEM ',IB

      ELSE IF( IA.EQ.8 ) THEN
         WRITE(6,*) 'LOAD IS EXCEED PLANT CAPACITY AT PLANT ',IB

      ELSE IF( IA.EQ.9 ) THEN
         WRITE(6,*) 'ERROR READING INPUT FILE [REFERENT SCHEDULE]'

      ELSE IF( IA.EQ.10 ) THEN
         WRITE(6,*) 'ERROR READING SYSTEM NAME in INPUT FILE'

      ELSE IF( IA.EQ.11 ) THEN
         WRITE(6,*) 'ERROR READING SYSTEM DATA AT SYSTEM ',IB

      ELSE IF( IA.EQ.12 ) THEN
         WRITE(6,*) 'ERROR READING PLANT NAME in INPUT FILE'

      ELSE IF( IA.EQ.13 ) THEN
         WRITE(6,*) 'ERROR READING PLANT DATA AT PLANT ',IB

      ELSE IF( IA.EQ.14 ) THEN
         WRITE(6,*) 'USER SPECIFIED SYSTEM TYPE OUT OF LIBRALY'

      ELSE IF( IA.EQ.15 ) THEN
         WRITE(6,*) 'USER SPECIFIED PLANT TYPE OUT OF LIBRALY'

      ELSE IF( IA.EQ.71 ) THEN
         WRITE(6,*) '***** ERROR :: COMMENT LINES DISAPPEAR'

      ELSE IF( IA.EQ.72 ) THEN
         WRITE(6,*) '***** ERROR :: CANNOT DETECT END OF COMMENT'

      END IF

      CALL XY(LINE+3,35)
      WRITE(6,*) 'PRESS ENTER'
      READ(5,*)
      CALL TAT(0)
      CALL TAT(1)
      CALL TCL(36,40)

      STOP
      END

C     ========== BELOW THIS LINE, ALL EITHER SUBROUTINES OR ===========
C     ========== FUNCTIONS ARE FOR MANAGING SCREEN,     NOT ===========
C     ========== INVOLVING TO SYMULATION.                ===========

      SUBROUTINE SCREEN()
C     ====================================================================
C     |                                                                  |
C     |    THIS SUBROUTINE WILL PERPARE THE APPROPREATE SCREEN FOR       |
C     |    RUNNING THE PROGRAM.                                          |
C     |                                                                  |
C     ====================================================================

      CALL TCL(37,40)
      CALL CSC()
      CALL BLOCK(12,2,78,22,37,44,47)
      CALL BLOCK(7,12,24,15,30,43,40)
      CALL TCL(33,40)
      CALL XY(9,14)
      WRITE(6,*) 'DD ) MM'
      CALL TCL(36,40)

      RETURN
      END

      SUBROUTINE BLOCK(TOP,LEFT,RIGHT,BOTTOM,CL1,CL2,CL3)
      INTEGER TOP,LEFT,RIGHT,BOTTOM
      INTEGER CL1,CL2,CL3
C     ====================================================================
C     |                                                                  |
C     |    THIS SUBROUTINE WILL CREATE A BLOCK WINDOW IN THE SCREEN.     |
C     |                                                                  |
C     ====================================================================


      CALL TCL(CL1,CL2)
      CALL TAT(1)
      DO 10 I=0,(RIGHT-LEFT),2
         CALL XY(TOP,LEFT+I)
         WRITE(6,'(A1)') CHAR(0)
         CALL XY(BOTTOM,LEFT+I)
         WRITE(6,'(A1)') CHAR(0)
   10 CONTINUE
      DO 20 I=TOP,BOTTOM,2
         CALL XY(I,LEFT)
         WRITE(6,'(A1)') CHAR(0)
         CALL XY(I,RIGHT)
         WRITE(6,'(A1)') CHAR(0)
   20 CONTINUE
      DO 30 I=1,(RIGHT-LEFT),2
         CALL XY(TOP,LEFT+I)
         WRITE(6,'(A1)') CHAR(177)
         CALL XY(BOTTOM,LEFT+I)
         WRITE(6,'(A1)') CHAR(177)
   30 CONTINUE
      DO 40 I=TOP+1,BOTTOM,2
         CALL XY(I,LEFT)
         WRITE(6,'(A1)') CHAR(177)
         CALL XY(I,RIGHT)
         WRITE(6,'(A1)') CHAR(177)
```

```
   40 CONTINUE

      CALL TCL(CL2,CL3)
      DO 50 I=TOP+1,BOTTOM-1
         DO 50 J=LEFT+1,RIGHT-1
            CALL XY(I,J)
            WRITE(6,'(A1)') CHAR(0)
   50 CONTINUE

      RETURN
      END

      SUBROUTINE TAT(IC)
      INTEGER IC
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE HELP USER TO SELECT TEXT ATTRIBUTE MODE.  |
C     |                  0       : STANDARD TEXT                     |
C     |                  1       : BOLD TOGGLE                       |
C     |                  5       : BLINK MODE                        |
C     |                  7       : INVERSE TEXT                      |
C     |                  8       : CONCEALED TEXT                    |
C     |                                                              |
C     ================================================================
      CHARACTER*5 S
      CHARACTER*2 C

      CALL STR(C,IC)
      S = CHAR(27)//'['//C//'m'
      WRITE(6,'(1X,A5)') S

      RETURN
      END

      SUBROUTINE TCL(TC,BC)
      INTEGER TC,BC
C     ================================================================
C     |                                                              |
C     |    THIS SUBROUTINE HELP USER TO SELECT TEXT COLOR MODE.      |
C     |                                                              |
C     |                          TEXT BACKGROUND                     |
C     |                BLACK      30         40                     |
C     |                RED        31         41                     |
C     |                GREEN      32         42                     |
C     |                YELLOW     33         43                     |
C     |                BLUE       34         44                     |
C     |                MAGENTA    35         45                     |
C     |                CYAN       36         46                     |
C     |                WHITE      37         47                     |
C     |                                                              |
C     ================================================================
      CHARACTER*8 S
      CHARACTER*2 T,B

      CALL STR(T,TC)
      CALL STR(B,BC)
      S = CHAR(27)//'['//T//';'//B//'m'
      WRITE(6,'(1X,A8)') S

      RETURN
      END

      SUBROUTINE CSC()
C     ================================================================
C     |                                                              |
C     |            SUBROUTINE FOR CLEARING TEXT SCREEN               |
C     |                                                              |
C     ================================================================
      CHARACTER*4 S

      S = CHAR(27)//'['//'2'//'J'
      WRITE(6,'(1X,A4)') S

      RETURN
      END

      SUBROUTINE CLN()
C     ================================================================
C     |                                                              |
C     |              SUBROUTINE FOR CLEARING A LINE                  |
C     |                                                              |
C     ================================================================
      CHARACTER*3 S

      S = CHAR(27)//'['//'K'
      WRITE(6,'(1X,A3)') S

      RETURN
      END

      SUBROUTINE XY(IR,IC)
      INTEGER IR,IC
C     ================================================================
C     |                                                              |
C     | THIS SUBROUTINE LOCATES SPECIFIED ROW & COLUMN ON THE MONITOR |
C     |                                                              |
C     ================================================================
      CHARACTER*8 S
      CHARACTER*2 R,C

      CALL STR(R,IR)
      CALL STR(C,IC)
      S = CHAR(27)//'['//R//';'//C//'H'
```

```
      WRITE(6,10) S
   10 FORMAT( 1X, A8\ )

      RETURN
      END

      SUBROUTINE STR(X,IX)
      INTEGER IX
      CHARACTER*2 X
C     ======================================================================
C     |                                                                    |
C     |   THIS SUBROUTINE WILL TRANSLATE TWO-DECIMA INTERGER NUMBER        |
C     |                  INTO TWO-BITE CHARACTER.                          |
C     |                                                                    |
C     ======================================================================
      CHARACTER*2 CX
      CHARACTER   C(2)
      EQUIVALENCE (CX,C)

      INTEGER DUMMY

      DUMMY = IX/10
      C(1) = CHAR(DUMMY+48)
      C(2) = CHAR(48+(IX-DUMMY*10))
      X = CX

      RETURN
      END
```

SOURCE CODE โปรแกรม REPORT

```
C     ================================================================
C     |                                                              |
C     |                  PROGRAM FOR REPORTING DATA                  |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @          DATAFILE         DATA FILENAME                     @
C     @          LINE             80-COLUMN CHARACTER               @

      CHARACTER*40 LINE
      CHARACTER*20 DATAFILE

      WRITE(6,1000)
      READ(5,'(A)') DATAFILE

      OPEN(1,FILE=DATAFILE)
      READ(1,'(A)') LINE
      CLOSE(1)
      IF( LINE.EQ.'BUILDING LOADS' ) CALL BLDG_LOAD(DATAFILE)
      IF( LINE.EQ.'SYSTEM LOAD'    ) CALL SYS_LOAD(DATAFILE)
      IF( LINE.EQ.'PLANT LOAD'     ) CALL PNT_LOAD(DATAFILE)

 1000 FORMAT(1X,'ENTER THE FILENAME OF DATA FILE ::> ')
      END

      SUBROUTINE BLDG_LOAD(DATAFILE)
      CHARACTER*40 DATAFILE
C     ================================================================
C     |                                                              |
C     |   REPORT BUILDING LOAD DATA IN THE FORM THAT USER SPECIFIED  |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @          DA               DAY                               @
C     @          DUM's            ARRAY OF DUMMY VARIABLE           @
C     @          END_COLUMN       THE LAST COLUMN OF LOAD SUM       @
C     @          FA               FLOOR AREA (m^2)                  @
C     @          I                COUNTER                           @
C     @          IMODE            MODE OF REPORT THAT USER SPECIFIED @
C     @          J                COUNTER                           @
C     @          K                COUNTER                           @
C     @          MO               MONTH                             @
C     @          NAME             NAME OF ZONE                      @
C     @          NUM_ROOM         NUMBER OF ROOM IN THE BUILDING    @
C     @          START_COLUMN     THE FIRST COLUMN OF LOAD SUM      @
C     @          SUM_LOAD's       SUM OF LOAD                       @
C     @          YY               YEAR                              @

      REAL    DUM(24,14),DUM2(24,14)
      REAL    SUM_LOAD(24)
      INTEGER NUM_ROOM
      REAL    FA
      CHARACTER*20 NAME
      CHARACTER*40 LINE
      INTEGER DA,MO,YY
      INTEGER IMODE,START_COLUMN,END_COLUMN
      INTEGER I,J,K

      OPEN(1,FILE= DATAFILE )
      OPEN(2,FILE='REPORT.OUT')
      OPEN(3,FILE='NAME.TMP',ACCESS='DIRECT',RECL=20)

    5 WRITE(6,1000)
      READ(5,*,ERR=5) IMODE
      IF( IMODE.GT.8 .OR. IMODE.LE.0 ) GO TO 5

      IF( IMODE.EQ.1 .OR. IMODE.EQ.2 ) WRITE(2,2000)
      IF( IMODE.EQ.3 .OR. IMODE.EQ.4 ) THEN
         WRITE(2,2001)
         START_COLUMN = 1
         END_COLUMN   = 3
      END IF
      IF( IMODE.EQ.5 ) THEN
         WRITE(2,2002)
         START_COLUMN = 1
         END_COLUMN   = 2
      END IF
      IF( IMODE.EQ.6 ) THEN
         WRITE(2,2003)
         START_COLUMN = 3
         END_COLUMN   = 3
      END IF
      IF( IMODE.EQ.7 .OR. IMODE.EQ.8 ) THEN
         WRITE(2,2004)
         START_COLUMN = 1
         END_COLUMN   = 12
      END IF

      READ(1,'(A)') LINE
      READ(1,*) NUM_ROOM
      DO 10 I=1,NUM_ROOM
         READ(1,'(A)') NAME
   10    WRITE(3,REC=I) NAME
      DO 11 I=1,NUM_ROOM
         READ(1,*) FA
```

```
11    WRITE(3,REC=(NUM_ROOM+I)) FA

15 I = 0
   DO 16 J=1,24
16    SUM_LOAD(J) = 0.0
   DO 17 J=1,24
      DO 17 K=1,14
17       DUM2(J,K) = 0.0

   READ(1,*,ERR=100,END=90) DA,MO,YY
   IF(IMODE.EQ.1.or.IMODE.EQ.2.or.IMODE.EQ.3.or.IMODE.EQ.7)
  +                                          WRITE(2,2010) DA,MO,YY

20 READ(1,*,ERR=100,END=90) DUM
   I = I+1

   IF(IMODE.EQ.3 .or. IMODE.EQ.7) THEN
      DO 25 J=1,24
25       SUM_LOAD(J) = 0.0
   END IF

   IF(IMODE.EQ.3.or.IMODE.EQ.4.or.
  +   IMODE.EQ.5.or.IMODE.EQ.6.or.
  +   IMODE.EQ.7.or.IMODE.EQ.8) THEN
      DO 30 J=START_COLUMN,END_COLUMN
         DO 30 K=1,24
            SUM_LOAD(K) = SUM_LOAD(K)+DUM(K,J)
30    CONTINUE
   END IF

   IF( IMODE.EQ.1 .OR. IMODE.EQ.3 .OR. IMODE.EQ.7 ) THEN
      READ(3,REC=I) NAME
      READ(3,REC=(NUM_ROOM+I)) FA
   END IF

   IF( IMODE.EQ.1 ) THEN
      WRITE(2,2020) NAME,FA
      DO 35 J=1,24
35       WRITE(2,2030) (DUM(J,K),K=1,14)
   END IF

   IF( IMODE.EQ.2 ) THEN
      DO 40 J=1,24
         DO 40 K=1,14
40          DUM2(J,K) = DUM2(J,K) + DUM(J,K)
   END IF

   IF( I.LT.NUM_ROOM ) THEN
      IF( IMODE.EQ.3 .OR. IMODE.EQ.7 )
  +                                  WRITE(2,2040) NAME,SUM_LOAD
      GO TO 20
   ELSE
      IF( IMODE.EQ.2 ) THEN
         DO 70 J=1,24
70          DUM2(J,13) = DUM2(J,13)/NUM_ROOM
         DO 80 J=1,24
80          WRITE(2,2030) (DUM2(J,K),K=1,14)
      END IF
      IF( IMODE.EQ.4 .OR. IMODE.EQ.5  .OR.
  +       IMODE.EQ.6 .OR. IMODE.EQ.8 )
  +       WRITE(2,2050) DA,MO,SUM_LOAD
      GO TO 15
   END IF
90 CONTINUE

100 CLOSE(1)
    CLOSE(2)
    CLOSE(3)
    STOP

1000 FORMAT(T30,'BUILDING LOAD REPORT',/,/,
    +       1X,'ENTER MODE OF REPORT :: ',/,
    +       T10,'1. ALL COMPONENTS (HOURLY,ZONES)',/,
    +       T10,'2. ALL COMPONENTS (HOURLY,WHOLE THE BUILDING)',/,
    +       T10,'3. SUM OF EXTERNAL LOAD (DAILY,ZONES)',/,
    +T10,'4. SUM OF EXTERNAL LOAD (DAILY, WHOLE THE BUILDING)',/,
    +T10,'5. SUM OF CONDUCTION LOADS (DAILAY, WHOLE THE BUILDING)',/,
    +T10,'6. SUM OF SOLAR LOAD (DAILY, WHOLE THE BUILDING)',/,
    +       T10,'7. SUMMARY (DAILY,ZONES)',/,
    +       T10,'8. SUMMARY (DAILY,WHOLE THE BUILDING)',/,
    +       /,' NUMBER ')
2000 FORMAT(1X,'| WALL<cond> | WIN <cond> | WIN <sola> | MAN  <sen> ',
    +          '| MAN  <lat> |  LIGHTING  | VENT <sen> | VENT <lat> ',
    +          '| INFI <sen> | INFI <lat> | MISC <sen> | MISC <lat> ',
    +          '|  DRY BULB  |     ER     |'
    +   , /,1X,'|     W      |     W      |     W      |     W      ',
    +          '|     W      |     W      |     W      |     W      ',
    +          '|     W      |     W      |     W      |     W      ',
    +          '|   deg C    |     W      |')
2001 FORMAT(1X, T10, 'SUM OF EXTERNAL COOLING LOADS', /,
    + 1X, 20X, '|     01     |     02     |     03     |     04     ',
    +          '|     05     |     06     |     07     |     08     ',
    +          '|     09     |     10     |     11     |     12     ',
    +          '|     13     |     14     |     15     |     16     ',
    +          '|     17     |     18     |     19     |     20     ',
    +          '|     21     |     22     |     23     |     24     |')
2002 FORMAT(1X, T10, 'SUM OF CONDUCTION LOADS', /,
    + 1X, 20X, '|     01     |     02     |     03     |     04     ',
    +          '|     05     |     06     |     07     |     08     ',
    +          '|     09     |     10     |     11     |     12     ',
    +          '|     13     |     14     |     15     |     16     ',
    +          '|     17     |     18     |     19     |     20     ',
    +          '|     21     |     22     |     23     |     24     |')
2003 FORMAT(1X, T10, 'SUM OF SOLAR LOADS', /,
```

```
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
 2004 FORMAT(1X, T10, 'SUMMARY', /,
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
 2010 FORMAT(1X, '## DATE (dd-mm-yyyy) --> ', I2, '-', I2, '-', I4)
 2020 FORMAT(1X, '## ROOM NAME         --> ', A20, ' ## ',
     +       'FLOOR AREA (sq.m) :: ',F10.2)
 2030 FORMAT(1X, '| ', 12(F10.0,' | '), (F10.1,' | '), (F10.0,' | '))
 2040 FORMAT(1X, A20, '| ',24(F10.0,' | '))
 2050 FORMAT(1X, I2,'/',I2, 15X, '| ',24(F10.0,' | '))
      END

      SUBROUTINE SYS_LOAD(DATAFILE)
      CHARACTER*40 DATAFILE
C     ================================================================
C     |                                                              |
C     |   REPORT   SYSTEM LOAD DATA IN THE FORM THAT USER SPECIFIED  |
C     |                                                              |
C     ================================================================

C     @ ---- ASSIGN VARIABLES                                        @
C     @         DA               DAY                                @
C     @         ENERGY's         CONSUMED SYSTEM ENERGY             @
C     @         I                COUNTER                            @
C     @         IMODE            MODE OF REPORT THAT USER SPECIFIED @
C     @         J                COUNTER                            @
C     @         LINE             80-COLUMN CHARACTER                @
C     @         LOAD's           SYSTEM LOAD                        @
C     @         MO               MONTH                              @
C     @         NAME             NAME OF SYSTEM                     @
C     @         NUM_SYS          NUMBER OF SYSTEM IN THE BUILDING   @
C     @         SUM's            SUM OF LOAD or USED ENERGY         @
C     @         YR               YEAR                               @

      CHARACTER*40 LINE
      CHARACTER*20 NAME
      INTEGER      IMODE,I,J
      INTEGER      NUM_SYS
      INTEGER      DA,MO,YR
      REAL         LOAD(24),ENERGY(24),SUM(24)

      OPEN(1,FILE= DATAFILE )
      OPEN(2,FILE='REPORT.OUT')
      OPEN(3,FILE='NAME.TMP',ACCESS='DIRECT',RECL=20)

   10 WRITE(6,1000)
      READ(5,*,ERR=10) IMODE
      IF( IMODE.LE.0 .OR. IMODE.GT.3 ) GO TO 10
      IF( IMODE.EQ.1 ) WRITE(2,2000)
      IF( IMODE.EQ.2 ) WRITE(2,2001)
      IF( IMODE.EQ.3 ) WRITE(2,2002)

      READ(1,'(A)') LINE
      READ(1,*) NUM_SYS
      DO 20 I=1,NUM_SYS
         READ(1,'(A)') NAME
         WRITE(3,REC=I) NAME
   20 CONTINUE

   30 READ(1,*,ERR=90,END=90) DA,MO,YR
      DO 40 I=1,24
   40    SUM(I) = 0.0
      IF( IMODE.EQ.1 ) WRITE(2,2010) DA,MO,YR
      DO 80 I=1,NUM_SYS
         READ(1,*) LOAD,ENERGY
         IF( IMODE.EQ.1 ) THEN
            READ(3,REC=I) NAME
            WRITE(2,2020) NAME
            DO 50 J=1,24
   50          WRITE(2,2030) LOAD(J),ENERGY(J)
         ELSE IF( IMODE.EQ.2 ) THEN
            DO 60 J=1,24
   60          SUM(J) = SUM(J)+LOAD(J)
         ELSE IF( IMODE.EQ.3 ) THEN
            DO 70 J=1,24
   70          SUM(J) = SUM(J)+ENERGY(J)
         END IF
   80 CONTINUE
      IF( IMODE.EQ.2 .OR. IMODE.EQ.3 ) WRITE(2,2050) DA,MO,SUM
      GO TO 30

   90 CONTINUE

      RETURN

 1000 FORMAT(T31,'SYSTEM LOAD REPORT',/,/,
     +       1X,'ENTER MODE OF REPORT :: ',/,
     +       T10,'1. LOAD & CONSUMED ENERGY (HOURLY,ZONES)',/,
     +       T10,'2. LOAD (HOURLY,WHOLE THE BUILDING)',/,
     +       T10,'3. CONSUMED ENERGY (HOURLY,WHOLE THE BUILDING)',/,
     +       /,' NUMBER ')
 2000 FORMAT(T50,'|    LOAD    |   ENERGY   |'
     +   , /,T50,'|     W      |     W      |')
 2001 FORMAT(1X, T10, 'SUM OF SYSTEM LOAD', /,
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
```

```
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
 2002 FORMAT(1X, T10, 'SUM OF CONSUMED SYSTEM ENERGY', /,
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
 2010 FORMAT(1X, '## DATE (dd-mm-yyyy) --> ', I2, '-', I2, '-', I4)
 2020 FORMAT(1X, '## SYSTEM NAME       --> ', A20, ' ## ')
 2030 FORMAT(T50, '| ', 2(F10.0,' | '))
 2040 FORMAT(1X, A20, '| ',24(F10.0,' | '))
 2050 FORMAT(1X, I2,'/',I2, 15X, '| ',24(F10.0,' | '))
      END

      SUBROUTINE PNT_LOAD(DATAFILE)
      CHARACTER*40 DATAFILE
C     ------------------------------------------------------------------
C     |                                                                |
C     |   REPORT    PLANT LOAD DATA IN THE FORM THAT USER SPECIFIED    |
C     |                                                                |
C     ------------------------------------------------------------------

C     @ ---- ASSIGN VARIABLES                                          @
C     @          DA              DAY                                  @
C     @          ENERGY's        CONSUMED PLANT  ENERGY               @
C     @          I               COUNTER                              @
C     @          IMODE           MODE OF REPORT THAT USER SPECIFIED   @
C     @          J               COUNTER                              @
C     @          LINE            80-COLUMN CHARACTER                  @
C     @          LOAD's          SYSTEM LOAD                          @
C     @          MO              MONTH                                @
C     @          NAME            NAME OF PLANT                        @
C     @          NUM_PNT         NUMBER OF PLANT  IN THE BUILDING     @
C     @          SUM's           SUM OF LOAD or USED ENERGY           @
C     @          YR              YEAR                                 @

      CHARACTER*40 LINE
      CHARACTER*20 NAME
      INTEGER      IMODE,I,J
      INTEGER      NUM_PNT
      INTEGER      DA,MO,YR
      REAL         LOAD(24),ENERGY(24),SUM(24)

      OPEN(1,FILE= DATAFILE )
      OPEN(2,FILE='REPORT.OUT')
      OPEN(3,FILE='NAME.TMP',ACCESS='DIRECT',RECL=20)

   10 WRITE(6,1000)
      READ(5,*,ERR=10) IMODE
      IF( IMODE.LE.0 .OR. IMODE.GT.3 ) GO TO 10
      IF( IMODE.EQ.1 ) WRITE(2,2000)
      IF( IMODE.EQ.2 ) WRITE(2,2001)
      IF( IMODE.EQ.3 ) WRITE(2,2002)

      READ(1,'(A)') LINE
      READ(1,*) NUM_PNT
      DO 20 I=1,NUM_PNT
         READ(1,'(A)') NAME
         WRITE(3,REC=I) NAME
   20 CONTINUE

   30 READ(1,*,ERR=90,END=90) DA,MO,YR
      DO 40 I=1,24
   40    SUM(I) = 0.0
      IF( IMODE.EQ.1 ) WRITE(2,2010) DA,MO,YR
      DO 80 I=1,NUM_PNT
         READ(1,*) LOAD,ENERGY
         IF( IMODE.EQ.1 ) THEN
            READ(3,REC=I) NAME
            WRITE(2,2020) NAME
            DO 50 J=1,24
   50          WRITE(2,2030) LOAD(J),ENERGY(J)
         ELSE IF( IMODE.EQ.2 ) THEN
            DO 60 J=1,24
   60          SUM(J) = SUM(J)+LOAD(J)
         ELSE IF( IMODE.EQ.3 ) THEN
            DO 70 J=1,24
   70          SUM(J) = SUM(J)+ENERGY(J)
         END IF
   80 CONTINUE
      IF( IMODE.EQ.2 .OR. IMODE.EQ.3 ) WRITE(2,2050) DA,MO,SUM
      GO TO 30

   90 CONTINUE

      RETURN

 1000 FORMAT(T31,'PLANT LOAD  REPORT',/,/,
     +        1X,'ENTER MODE OF REPORT :: ',/,
     +       T10,'1. LOAD & CONSUMED ENERGY (HOURLY,ZONES)',/,
     +       T10,'2. LOAD (HOURLY,WHOLE THE BUILDING)',/,
     +       T10,'3. CONSUMED ENERGY (HOURLY,WHOLE THE BUILDING)',/,
     +       /,' NUMBER ')
 2000 FORMAT(T50,'|    LOAD    |   ENERGY   |'
     +   , /,T50,'|     W      |     W      |')
 2001 FORMAT(1X, T10, 'SUM OF PLANT LOAD', /,
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
```

```
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
2002 FORMAT(1X, T10, 'SUM OF CONSUMED PLANT ENERGY', /,
     + 1X, 20X, '|     01     |     02     |     03     |     04     ',
     +          '|     05     |     06     |     07     |     08     ',
     +          '|     09     |     10     |     11     |     12     ',
     +          '|     13     |     14     |     15     |     16     ',
     +          '|     17     |     18     |     19     |     20     ',
     +          '|     21     |     22     |     23     |     24     |')
2010 FORMAT(1X, '## DATE (dd-mm-yyyy) --> ', I2, '-', I2, '-', I4)
2020 FORMAT(1X, '## PLANT NAME        --> ', A20, ' ## ')
2030 FORMAT(T50, '| ', 2(F10.0,' | '))
2040 FORMAT(1X, A20, '| ',24(F10.0,' | '))
2050 FORMAT(1X, I2,'/',I2, 15X, '| ',24(F10.0,' | '))
     END
```

ภาคผนวก ค

INPUT FILE (อาคารสำนักงานใหญ่การปิโตรเลียมแห่งประเทศไทย)

INPUT FILE สำหรับโปรแกรม CAL.EXE

```
<                          Building's geometry data file

        Project          : PTT building
        Input by         : Mr. Boonchai Lertnuwat
        Date             : 17 March 1997
        Comment          : This input file is used as an example in thesis,
                           named Development of Energy Simulation Program
                           for Building in Thailand.
>
< Input MODE of computation (1,2,3)
        Choose 1 : for computation on a day          user must specify the date of computation on the below line
        Choose 2 : for computation on a period       user must specify the period of computation on the next two
lines
        Choose 3 : for computation in whole a year   on more detail line is needed
        format
                   MODE
                   DD MM (in case of either MODE=1 or MODE=2)
                   DD MM (in case of MODE=2 )
>
2
18 12
24 12
< List of Temporary wall library :
        NUMBER OF WALL (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
3
879.0  647. 879.0 000.0 000.0 000.0 000.0 000.0
0.721 31.98 0.721 000.0 000.0 000.0 000.0 000.0
1858. 5049. 1858. 000.0 000.0 000.0 000.0 000.0
 .013 .600  0.013 000.0 000.0 000.0 000.0 000.0
0.90
 840.  647. 879.0 000.0 000.0 000.0 000.0 000.0
2.924 31.98 0.721 000.0 000.0 000.0 000.0 000.0
 264. 5049. 1858. 000.0 000.0 000.0 000.0 000.0
0.010 .600  0.013 000.0 000.0 000.0 000.0 000.0
0.90
 840.  840. 000.0 000.0 000.0 000.0 000.0 000.0
0.040 1.731 000.0 000.0 000.0 000.0 000.0 000.0
1119. 2243. 000.0 000.0 000.0 000.0 000.0 000.0
.0032 .200  000.0 000.0 000.0 000.0 000.0 000.0
0.90
< List of Temporary window library :
        NUMBER OF WINDOW (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        SHADDING_FACTOR     ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        SHADDING_FACTOR     ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
2
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
125.0 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0.85 0.00
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
125.0 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0000. 0000. 0000. 000.0 000.0 000.0 000.0 000.0
0.85 0.00
< List of Temporary partition library :
        NUMBER OF PARTITION (specified in this library)
        Cp1          Cp2          Cp3          Cp4          ... Cp8          )  the
        K1           K2           K3           K4           ... K8           )  first
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     )  wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        ABSORBSITIVITY

        Cp1          Cp2          Cp3          Cp4          ... Cp8          ) the
        K1           K2           K3           K4           ... K8           ) second
        DENSITY1     DENSITY2     DENSITY3     DENSITY4     ... DENSITY8     ) wall
        THICKNESS1 THICKNESS2 THICKNESS3 THICKNESS4 ... THICKNESS8  )
        ABSORBSITIVITY

        (note that LAYERS of material are arranged from OUTSIDE to INSIDE)
>
```

```
2
2850. 000.0 2850. 000.0 000.0 000.0 000.0 000.0
0.622 5.847 0.622 000.0 000.0 000.0 000.0 000.0
 880. 000.0  880. 000.0 000.0 000.0 000.0 000.0
0.012 000.0 0.012 000.0 000.0 000.0 000.0 000.0
0.90
000.0  840. 000.0 000.0 000.0 000.0 000.0 000.0
4.608 1.731 000.0 000.0 000.0 000.0 000.0 000.0
000.0 2243. 000.0 000.0 000.0 000.0 000.0 000.0
000.0 .150  000.0 000.0 000.0 000.0 000.0 000.0
0.90
< List of SCHEDULE library :
        NUMBER OF SCHEDULE (specified in this library)
        A1     A2     A3     A4     A5          ...         A24
        B1     B2     B3     B4     B5          ...         B24
    .   .                                                  .
        .                                                  .
        .                                                  .
>
12
2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5
2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5
00000. 00000. 00000. 00000. 00000. 0.0000 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5
2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 00000. 00000.
00000. 00000. 00000. 00000. 00000. 0.0000 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5 2.32E5
00000. 00000. 00000. 00000. 00000. 00000. 00000. 00000.
0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
0.0    0.0    0.0    0.0    0.0    0.0    0.0
25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0
25.0   25.0   25.0   25.0   25.0   25.0   25.0   25.0
0.0    0.0    0.0    0.0    0.0    1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
0.0    0.0    0.0    0.0    0.0    0.50   0.50   0.50   0.50   0.50   0.50   0.50   0.50   0.50   0.50   0.50   0.50
0.50   0.00   0.00   0.00   0.00   0.00   0.00   0.00
1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
0.0    0.0    0.0    0.0    0.0    0.00   0.25   1.00   1.00   1.00   1.00   0.20   1.00   1.00   1.00   1.00   1.00
0.15   0.10   0.10   0.00   0.00   0.00   0.00   0.00
0.0    0.0    0.0    0.0    0.0    0.00   0.13   0.50   0.50   0.50   0.50   0.10   0.10   0.10   0.10   0.10   0.10
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
0.0    0.0    0.0    0.0    0.0    0.00   0.35   1.00   1.00   1.00   1.00   0.20   1.00   1.00   1.00   1.00   1.00
0.35   0.20   0.20   0.00   0.00   0.00   0.00   0.00
0.0    0.0    0.0    0.0    0.0    0.00   0.18   0.50   0.50   0.50   0.50   0.10   0.10   0.10   0.10   0.10   0.10
0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
< ORENTATION of the building (in degree refer to North) :
```


```
>
180.00
< NUMBER OF ZONE (specified in this simulation)
  NAME OF ZONE1 (20-character limitation)
  NAME OF ZONE2              ,,
  NAME OF ZONE3              ,,
>
25
Ground floor
Ground floor (lift)
2nd floor (lift)
3rd floor
4th floor
5th floor
6nd floor
7rd floor
8th floor
9th floor
10th floor
11th floor
12th floor
13th floor
14th floor
15th floor
16th floor
17th floor
18th floor
19th floor
20th floor
21th floor
22th floor
23th floor
24th floor
< Zone 1
        Thermo properties of zone :-

        V_SOLAR1          V_SOLAR2          V_SOLAR3
        V_CONDUCTION1     V_CONDUCTION2     V_CONDUCTION3
        V_LIGHTING1       V_LIGHTING2       V_LIGHTING3
        W1                W2
        KT                PERIMETER
        G1                G2                G3              P1              P2
        FLOOR_AREA
        CONSTANT_ROOM_TEMP.
        THROTTLING_RANGE
>
0.187 -.152 0.0
```

```
0.676 -.641 0.0
0.65  -.615 0.0 -.965 0.0
-1.00 146.4
10.50 -11.07 0.57 1.0 -.93
1285.56
25.0 1.0
< NUMBER_OF_SCHEDULE
  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

>
1
01 01  31 12  4  4  4  4  4  4  4  4
               4  4  4  4  4  4  4  4
               5  5  5  5  5  5  5  5
< NUMBER_OF_WALL (in this zone)
  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
  CO-ORINATE_X        CO-ORDINATE_Y        CO-ORDINATE_Z    WIDTH    HEIGHT
  WALL_TYPE (1=external wall   2=unsunlit wall   3=partition)
  ADJACENT_ROOM (in case of WALL_TYPE=3, or else ADJACENT_ROOM=0)
  NUMBER_OF_WINDOW (on this wall)
     AZIMUTH_OF_WINDOW_1   TILT_OF_WINDOW_1
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

     AZIMUTH_OF_WINDOW_2   TILT_OF_WINDOW_2
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
        .
        .
        .

>
9
315. 000.
2 1285.56
00.0 00.0 10.0 36.6 36.6
 3
 4
 0

315. 090.
2 246.0
00.0 00.0 00.0 36.6 10.0
 1
 0
 24
1 5.0
0.42 -.42 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
 1.6 -1.6 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
02.7 -2.7 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
03.8 -3.8 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
04.9 -4.9 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
06.1 -6.1 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
07.2 -7.2 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
08.3 -8.3 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
09.5 -9.5 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
011. -11. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
012. -12. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
013. -13. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
014. -14. 01.0 01.0 05.0
```

```
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
015. -15. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
016. -16. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
017. -17. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
019. -19. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
020. -20. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
021. -21. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
022. -22. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
023. -23. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
024. -24. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
025. -25. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
026. -26. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5

045. 090.
2 246.0
25.9 -26. 00.0 36.6 10.0
 1
 0
 24
1 5.0
026. -25. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
 27. -24. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
029. -23. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
030. -22. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
031. -21. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
032. -20. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
033. -19. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
034. -18. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
035. -16. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
036. -15. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
038. -14. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
039. -13. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
040. -12. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
041. -11. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
042. -9.6 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
043. -8.5 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
044. -7.4 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
046. -6.2 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
047. -5.1 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
048. -4.0 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
050. -2.8 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
```

```
1 5.0
050. -1.7 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
051. -.57 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
052. 0.57 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5

135. 090.
2 246.0
51.8 00.0 00.0 36.6 10.0
 1
 0
 24
1 5.0
051.  .39 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
050.  1.4 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
049.  2.5 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
048.  3.6 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
047.  4.6 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
046.  5.7 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
045.  6.7 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
044.  7.8 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
043.  8.9 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
042.  9.9 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
041.  11. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
040.  12. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
039.  13. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
038.  14. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
037.  15. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
035.  16. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
034.  17. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
033.  18. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
032.  19. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
031.  21. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
030.  22. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
029.  23. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
028.  24. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
027. 025. 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5

225. 090.
2 246.0
25.9 25.9 00.0 36.6 10.0
 1
 0
 24
1 5.0
25.5 25.5 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
24.4 24.4 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
23.4 23.4 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
```

```
1 5.0
22.3 22.3 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
21.2 21.2 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
20.2 20.2 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
19.1 19.1 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
18.1 18.1 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
17.0 17.0 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
15.9 15.9 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
14.9 14.9 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
13.8 13.8 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
12.8 12.8 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
11.7 11.7 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
10.6 10.6 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
9.58 9.58 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
8.52 8.52 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
7.46 7.46 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
6.40 6.40 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
5.34 5.34 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
4.28 4.28 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
3.22 3.22 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
2.16 2.16 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 5.0
1.10 1.10 01.0 01.0 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.5

315. 090.
-1 150.0
20.1 4.79 00.0 15.0 10.0
 3
 2
 0

045. 090.
-1 00.00
30.7 -5.8 00.0 03.6 10.0
 3
 2
 1
2 36.0
30.7 -5.8 00.0 03.6 10.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

135. 090.
-1 150.0
33.2 -3.3 00.0 15.0 10.0
 3
 2
 0

225. 090.
-1 00.00
22.6 7.34 00.0 03.6 10.0
 3
 2
 1
2 36.0
22.6 7.34 00.0 03.6 10.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

< NUMBER_OF_PEOPLE   SENSIBLE_FACTOR   RADIATION_FACTOR   LATENT_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
000 065. 0.33 055. 1
01 01  31 12  4 4 4 4 4 4 4 4
```

```
< MAX_KILOWATT_OF_LIGHTING   RADIATION_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
00.0 0.50 1
01 01  31 12  4 4 4 4 4 4 4 4

< MAX_cubic_METER_per_SECOND_OF_VENTILATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.000 1
01 01  31 12  4 4 4 4 4 4 4 4

< MAX_cubic_METER_per_SECOND_OF_INFILTRATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.000 1
01 01  31 12  6 6 6 6 6 6 6 6

< SENSIBLE_WATT_OF_APPLIANCES   RADIATION_FACTOR   LATENT_WATT_OF_APPLIANCES
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
000. 0.33 000. 1
01 01  31 12  4 4 4 4 4 4 4 4

< Zone 2
        Thermo properties of zone :-

        V_SOLAR1          V_SOLAR2          V_SOLAR3
        V_CONDUCTION1     V_CONDUCTION2     V_CONDUCTION3
        V_LIGHTING1       V_LIGHTING2       V_LIGHTING3
        W1                W2
        KT                PERIMETER
        G1                G2                G3                P1            P2
        FLOOR_AREA
        CONSTANT_ROOM_TEMP.
        THROTTLING_RANGE
>
0.187 -.152 0.0
0.676 -.641 0.0
0.65  -.615 0.0 -.965 0.0
-1.00 111.6
10.50 -11.07 0.57 1.0 -.93
54.00
25.0 1.0
< NUMBER_OF_SCHEDULE
  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

>
1
01 01  31 12  1  1  1  1  1  1  1  1
               4  4  4  4  4  4  4  4
               5  5  5  5  5  5  5  5
< NUMBER_OF_WALL (in this zone)
  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
  CO-ORINATE_X        CO-ORDINATE_Y        CO-ORDINATE_Z    WIDTH    HEIGHT
  WALL_TYPE (1=external wall   2=unsunlit wall   3=partition)
  ADJACENT_ROOM (in case of WALL_TYPE=3, or else ADJACENT_ROOM=0)
  NUMBER_OF_WINDOW (on this wall)
     AZIMUTH_OF_WINDOW_1   TILT_OF_WINDOW_1
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

     AZIMUTH_OF_WINDOW_2   TILT_OF_WINDOW_2
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
        .
        .
        .

>
5
315. 000.
2 54.00
20.1 4.79 05.0 15.0 03.6
 3
 3
 0

315. 090.
```

```
1 075.0
20.1 4.79 00.0 15.0 05.0
 3
 1
 0

045. 090.
1 00.00
30.7 -5.8 00.0 03.6 05.0
 3
 1
 1
2 18.0
30.7 -5.8 00.0 03.6 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

135. 090.
1 075.0
33.2 -3.3 00.0 15.0 05.0
 3
 1
 0

225. 090.
1 00.00
22.6 7.34 00.0 03.6 05.0
 3
 1
 1
2 18.0
22.6 7.34 00.0 03.6 05.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

< NUMBER_OF_PEOPLE   SENSIBLE_FACTOR   RADIATION_FACTOR   LATENT_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
010 065. 0.33 055. 1
01 01  31 12  4 6 6 6 6 6 7 4

< MAX_KILOWATT_OF_LIGHTING   RADIATION_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
03.3 0.50 1
01 01  31 12  4 8 8 8 8 8 8 4

< MAX_cubic_METER_per_SECOND_OF_VENTILATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.000 1
01 01  31 12  4 4 4 4 4 4 4 4

< MAX_cubic_METER_per_SECOND_OF_INFILTRATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.000 1
01 01  31 12  4 4 4 4 4 4 4 4

< SENSIBLE_WATT_OF_APPLIANCES   RADIATION_FACTOR   LATENT_WATT_OF_APPLIANCES
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
000. 0.33 000. 1
01 01  31 12  4 4 4 4 4 4 4 4

< Zone 3
      Thermo properties of zone :-

      V_SOLAR1          V_SOLAR2          V_SOLAR3
      V_CONDUCTION1     V_CONDUCTION2     V_CONDUCTION3
      V_LIGHTING1       V_LIGHTING2       V_LIGHTING3
      W1                W2
      KT                PERIMETER
      G1                G2                G3                P1                P2
      FLOOR_AREA
      CONSTANT_ROOM_TEMP.
      THROTTLING_RANGE
>
0.187 -.152 0.0
0.676 -.641 0.0
0.65  -.615 0.0 -.965 0.0
-1.00 111.6
10.50 -11.07 0.57 1.0 -.93
54.00
25.0 1.0
< NUMBER_OF_SCHEDULE
  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (thermostat)

  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL   (thermostat)

>
1
01 01  31 12  1  1  1  1  1  1  1  1
               4  4  4  4  4  4  4  4
               5  5  5  5  5  5  5  5
```

```
< NUMBER_OF_WALL (in this zone)
  AZIMUTH_OF_WALL_1    TILT_OF_WALL_1
  WALL_REF_No.         AREA
  CO-ORINATE_X         CO-ORDINATE_Y        CO-ORDINATE_Z     WIDTH     HEIGHT
  WALL_TYPE (1=external wall   2=unsunlit wall   3=partition)
  ADJACENT_ROOM (in case of WALL_TYPE=3, or else ADJACENT_ROOM=0)
  NUMBER_OF_WINDOW (on this wall)
     AZIMUTH_OF_WINDOW_1    TILT_OF_WINDOW_1
     WINDOW_REF_No.         AREA
     CO-ORINATE_X           CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN      OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG            OVERHANG_PROLONG
     FIN_FLANGE             OVERHANG_FLANGE
     RECESS_OF_WINDOW

     AZIMUTH_OF_WINDOW_2    TILT_OF_WINDOW_2
     WINDOW_REF_No.         AREA
     CO-ORINATE_X           CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN      OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG            OVERHANG_PROLONG
     FIN_FLANGE             OVERHANG_FLANGE
     RECESS_OF_WINDOW

  AZIMUTH_OF_WALL_1    TILT_OF_WALL_1
  WALL_REF_No.         AREA
        .
        .
        .

>
6
315. 000.
2 54.00
20.1 4.79 10.0 15.0 03.6
 3
 4
 0

315. 000.
-2 54.00
20.1 4.79 05.0 15.0 03.6
 3
 2
 0

315. 090.
1 075.0
20.1 4.79 05.0 15.0 05.0
 3
 1
 0

045. 090.
1 00.00
30.7 -5.8 05.0 03.6 05.0
 3
 1
 1
2 16.0
30.7 -5.8 00.0 03.6 10.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

135. 090.
1 075.0
33.2 -3.3 05.0 15.0 05.0
 3
 1
 0

225. 090.
1 00.00
22.6 7.34 05.0 03.6 05.0
 3
 1
 1
2 16.0
22.6 7.34 00.0 03.6 10.0
00.0 00.0 00.0 00.0 00.0 00.0 00.0

< NUMBER_OF_PEOPLE   SENSIBLE_FACTOR   RADIATION_FACTOR   LATENT_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
010 065. 0.33 055. 1
01 01  31 12  4 6 6 6 6 6 7 4

< MAX_KILOWATT_OF_LIGHTING   RADIATION_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
3.15 0.50 1
01 01  31 12  4 8 8 8 8 8 8 4

< MAX_cubic_METER_per_SECOND_OF_VENTILATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.142 1
01 01  31 12  4 8 8 8 8 8 8 4

< MAX_cubic_METER_per_SECOND_OF_INFILTRATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
```

```
.000 1
01 01  31 12  4 4 4 4 4 4 4 4

< SENSIBLE_WATT_OF_APPLIANCES   RADIATION_FACTOR   LATENT_WATT_OF_APPLIANCES
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
000. 0.33 000. 1
01 01  31 12  4 4 4 4 4 4 4 4

< Zone 4
        Thermo properties of zone :-

        V_SOLAR1          V_SOLAR2          V_SOLAR3
        V_CONDUCTION1     V_CONDUCTION2     V_CONDUCTION3
        V_LIGHTING1       V_LIGHTING2       V_LIGHTING3
        W1                W2
        KT                PERIMETER
        G1                G2                G3              P1              P2
        FLOOR_AREA
        CONSTANT_ROOM_TEMP.
        THROTTLING_RANGE
>
0.187 -.152 0.0
0.676 -.641 0.0
0.65  -.615 0.0 -.965 0.0
-1.00 146.4
10.50 -11.07 0.57 1.0 -.93
1340.00
25.0 1.0
< NUMBER_OF_SCHEDULE
  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

>
1
01 01  31 12  4  2  2  2  2  2  3  4
               4  4  4  4  4  4  4  4
               5  5  5  5  5  5  5  5
< NUMBER_OF_WALL (in this zone)
  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
  CO-ORINATE_X        CO-ORDINATE_Y       CO-ORDINATE_Z     WIDTH     HEIGHT
  WALL_TYPE (1=external wall   2=unsunlit wall   3=partition)
  ADJACENT_ROOM (in case of WALL_TYPE=3, or else ADJACENT_ROOM=0)
  NUMBER_OF_WINDOW (on this wall)
     AZIMUTH_OF_WINDOW_1   TILT_OF_WINDOW_1
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

     AZIMUTH_OF_WINDOW_2   TILT_OF_WINDOW_2
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
        .
        .
        .

>
6
315. 000.
2 1340.00
00.0 00.0 14.0 36.6 36.6
 3
 5
 0

315. 000.
-2 1340.00
00.0 00.0 10.0 36.6 36.6
 3
 1
 0 .

315. 090.
1 86.40
00.0 00.0 10.0 36.6 04.0
 1
 0
 24
1 2.5
0.42 -.42 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
 1.6 -1.6 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
```

```
02.7 -2.7 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
03.8 -3.8 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
04.9 -4.9 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
06.1 -6.1 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
07.2 -7.2 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
08.3 -8.3 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
09.5 -9.5 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
011. -11. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
012. -12. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
013. -13. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
014. -14. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
015. -15. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
016. -16. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
017. -17. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
019. -19. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
020. -20. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
021. -21. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
022. -22. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
023. -23. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
024. -24. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
025. -25. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
026. -26. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

045. 090.
1 86.40
25.9 -26. 10.0 36.6 04.0
 1
 0
 24
1 2.5
026. -25. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
 27. -24. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
029. -23. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
030. -22. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
031. -21. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
032. -20. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
033. -19. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
034. -18. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
035. -16. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
036. -15. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
038. -14. 11.0 01.0 02.5
```

```
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
039. -13. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
040. -12. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
041. -11. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
042. -9.6 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
043. -8.5 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
044. -7.4 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
046. -6.2 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
047. -5.1 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
048. -4.0 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050. -2.8 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050. -1.7 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
051. -.57 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
052. 0.57 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

135. 090.
1 86.4
51.8 00.0 10.0 36.6 04.0
 1
 0
 24
1 2.5
051.  .39 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050.  1.4 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
049.  2.5 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
048.  3.6 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
047.  4.6 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
046.  5.7 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
045.  6.7 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
044.  7.8 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
043.  8.9 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
042.  9.9 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
041.  11. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
040.  12. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
039.  13. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
038.  14. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
037.  15. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
035.  16. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
034.  17. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
033.  18. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
032.  19. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
```

```
1 2.5
031.  21. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
030.  22. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
029.  23. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
028.  24. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
027. 025. 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

225. 090.
1 86.40
25.9 25.9 10.0 36.6 04.0
 1
 0
 24
1 2.5
25.5 25.5 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
24.4 24.4 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
23.4 23.4 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
22.3 22.3 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
21.2 21.2 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
20.2 20.2 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
19.1 19.1 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
18.1 18.1 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
17.0 17.0 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
15.9 15.9 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
14.9 14.9 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
13.8 13.8 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
12.8 12.8 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
11.7 11.7 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
10.6 10.6 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
9.58 9.58 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
8.52 8.52 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
7.46 7.46 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
6.40 6.40 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
5.34 5.34 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
4.28 4.28 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
3.22 3.22 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
2.16 2.16 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
1.10 1.10 11.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

< NUMBER_OF_PEOPLE   SENSIBLE_FACTOR   RADIATION_FACTOR   LATENT_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
054 065. 0.33 055. 1
01 01  31 12   4  9  9  9  9  9 10  4

< MAX_KILOWATT_OF_LIGHTING   RADIATION_FACTOR
  NUMBER_OF_SCHEDULE
```

```
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
14.0 0.50 1
01 01  31 12   4 11 11 11 11 11 12  4

< MAX_cubic_METER_per_SECOND_OF_VENTILATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.757 1
01 01  31 12   4 11 11 11 11 11 12  4

< MAX_cubic_METER_per_SECOND_OF_INFILTRATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.000 1
01 01  31 12   4  4  4  4  4  4  4  4

< SENSIBLE_WATT_OF_APPLIANCES   RADIATION_FACTOR   LATENT_WATT_OF_APPLIANCES
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
000. 0.33 000. 1
01 01  31 12   4  4  4  4  4  4  4  4

< Zone 5
< Zone 6
< Zone 7
    .
    .
    .
< Zone 24
< Zone 25
       Thermo properties of zone :-

       V_SOLAR1          V_SOLAR2          V_SOLAR3
       V_CONDUCTION1     V_CONDUCTION2     V_CONDUCTION3
       V_LIGHTING1       V_LIGHTING2       V_LIGHTING3
       W1                W2
       KT                PERIMETER
       G1                G2                G3              P1              P2
       FLOOR_AREA
       CONSTANT_ROOM_TEMP.
       THROTTLING_RANGE
>
0.187 -.152 0.0
0.676 -.641 0.0
0.65  -.615 0.0 -.965 0.0
-1.00 146.4
10.50 -11.07 0.57 1.0 -.93
1340.00
25.0 1.0
< NUMBER_OF_SCHEDULE
  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

  STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER max)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (ER min)
  REF_SCHE_No._of_SUN  REF_SCHE_No._of_MON   ...  REF_SCHE_No._of_SAT  REF_SCHE_No._of_HOL    (thermostat)

>
1
01 01  31 12  4  2  2  2  2  2  3  4
               4  4  4  4  4  4  4  4
               5  5  5  5  5  5  5  5
< NUMBER_OF_WALL (in this zone)
  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
  CO-ORINATE_X        CO-ORDINATE_Y        CO-ORDINATE_Z     WIDTH     HEIGHT
  WALL_TYPE (1=external wall   2=unsunlit wall   3=partition)
  ADJACENT_ROOM (in case of WALL_TYPE=3, or else ADJACENT_ROOM=0)
  NUMBER_OF_WINDOW (on this wall)
     AZIMUTH_OF_WINDOW_1   TILT_OF_WINDOW_1
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

     AZIMUTH_OF_WINDOW_2   TILT_OF_WINDOW_2
     WINDOW_REF_No.        AREA
     CO-ORINATE_X          CO-ORDINATE_Y          CO-ORDINATE_Z    WIDTH   HEIGHT
     FIN_OFFSET_TO_WIN     OVERHANG_OFFSET_TO_WIN
     FIN_PROLONG           OVERHANG_PROLONG
     FIN_FLANGE            OVERHANG_FLANGE
     RECESS_OF_WINDOW

  AZIMUTH_OF_WALL_1   TILT_OF_WALL_1
  WALL_REF_No.        AREA
        .
        .
        .

>
6
315. 000.
3 1340.00
00.0 00.0 98.0 36.6 36.6
```

รูปแบบเดียวกันกับโซนที่ 4 แต่ปรับเปลี่ยนจำนวนคน และการใช้พลังงานแสงสว่าง ไปตามแต่ละชั้น

```
 1
 0
 0

315. 000.
-2 1340.00
00.0 00.0 94.0 36.6 36.6
 3
 24
 0

315. 090.
1 86.40
00.0 00.0 94.0 36.6 04.0
 1
 0
 24
1 2.5
0.42 -.42 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
 1.6 -1.6 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
02.7 -2.7 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
03.8 -3.8 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
04.9 -4.9 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
06.1 -6.1 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
07.2 -7.2 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
08.3 -8.3 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
09.5 -9.5 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
011. -11. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
012. -12. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
013. -13. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
014. -14. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
015. -15. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
016. -16. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
017. -17. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
019. -19. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
020. -20. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
021. -21. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
022. -22. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
023. -23. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
024. -24. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
025. -25. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
026. -26. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

045. 090.
1 86.40
25.9 -26. 94.0 36.6 04.0
 1
 0
 24
1 2.5
026. -25. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
 27. -24. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
029. -23. 95.0 01.0 02.5
```

```
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
030. -22. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
031. -21. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
032. -20. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
033. -19. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
034. -18. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
035. -16. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
036. -15. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
038. -14. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
039. -13. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
040. -12. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
041. -11. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
042. -9.6 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
043. -8.5 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
044. -7.4 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
046. -6.2 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
047. -5.1 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
048. -4.0 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050. -2.8 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050. -1.7 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
051. -.57 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
052. 0.57 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

135. 090.
1 86.40
51.8 00.0 94.0 36.6 04.0
 1
 0
 24
1 2.5
051.  .39 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
050.  1.4 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
049.  2.5 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
048.  3.6 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
047.  4.6 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
046.  5.7 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
045.  6.7 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
044.  7.8 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
043.  8.9 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
042.  9.9 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
041.  11. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
```

```
1 2.5
040.  12. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
039.  13. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
038.  14. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
037.  15. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
035.  16. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
034.  17. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
033.  18. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
032.  19. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
031.  21. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
030.  22. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
029.  23. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
028.  24. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
027. 025. 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

225. 090.
1 86.40
25.9 25.9 94.0 36.6 04.0
 1
 0
 24
1 2.5
25.5 25.5 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
24.4 24.4 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
23.4 23.4 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
22.3 22.3 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
21.2 21.2 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
20.2 20.2 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
19.1 19.1 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
18.1 18.1 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
17.0 17.0 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
15.9 15.9 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
14.9 14.9 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
13.8 13.8 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
12.8 12.8 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
11.7 11.7 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
10.6 10.6 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
9.58 9.58 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
8.52 8.52 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
7.46 7.46 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
6.40 6.40 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
```

```
5.34 5.34 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
4.28 4.28 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
3.22 3.22 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
2.16 2.16 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5
1 2.5
1.10 1.10 95.0 01.0 02.5
00.0 00.0 00.0 00.0 00.0 00.0 00.5

< NUMBER_OF_PEOPLE   SENSIBLE_FACTOR   RADIATION_FACTOR   LATENT_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
010 065. 0.33 055. 1
01 01  31 12   4  9  9  9  9  9 10  4

< MAX_KILOWATT_OF_LIGHTING   RADIATION_FACTOR
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
15.75 0.50 1
01 01  31 12   4 11 11 11 11 11 12  4

< MAX_cubic_METER_per_SECOND_OF_VENTILATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.757 1
01 01  31 12   4 11 11 11 11 11 12  4

< MAX_cubic_METER_per_SECOND_OF_INFILTRATION
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
.105 1
01 01  31 12   6  6  6  6  6  6  6  6

< SENSIBLE_WATT_OF_APPLIANCES   RADIATION_FACTOR   LATENT_WATT_OF_APPLIANCES
  NUMBER_OF_SCHEDULE
  REF._SCHE_No._of_SUN   REF._SCHE_No._of_MON ... REF._SCHE_No._of_SAT   REF._SCHE_No._of_HOL
>
12.3e3 0.33 000. 1
01 01  31 12   4 11 11 11 11 11 12 4
```

INPUT FILE สำหรับโปรแกรม SYSTEM.EXE

```
<                          SYSTEM & PLANT DATA FILE

        Project  : PTT Building
        Input by : Mr. Boonchai Lertnuwat
        Date     : 19 March 1997
        Comment  :

>
< Input list of library schedule :
  NUMBER_OF_SCHEDULE (specified in this file)
        HOUR1   HOUR2   HOUR3   ... HOUR24    (schedule No.1)
        HOUR1   HOUR2   HOUR3   ... HOUR24    (schedule No.2)
          .
          .
>
4
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
0.00 0.00 0.00 0.00 0.00 0.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 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 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 0.00 0.00 0.00 0.00 0.00

< NUMBER_OF_SYSTEM
  NAME_OF_SYSTEM_1
  NAME_OF_SYSTEM_2
          .
          .
          .

>
24
AHU-01
AHU-02
AHU-03
AHU-04
AHU-05
AHU-06
AHU-07
AHU-08
AHU-09
AHU-10
AHU-11
AHU-12
AHU-13
AHU-14
AHU-15
AHU-16
AHU-17
AHU-18
AHU-19
AHU-20
AHU-21
AHU-22
AHU-23
AHU-24

< Input details of systems :
  TYPE_OF_SYSTEM
  MAX_CAPACITY   MAX_CFM   PRESSURE_DROP_OF_AIRFLOW_IN_SYSTEM
  NUMBER_SCHDULE
     STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH   SCHED_OF_SUN   SCHED_OF_MON ... SCHED_OF_HOL
          .
          .
          .
  NUMBER_OF_SERVED_ZONE
     SERVED_ZONE_NUMBER
          .
          .
          .

>
AHU LOW PRESSURE
 19800.0  2000.00 12.5
1
01 01 31 12 1 1 1 1 1 1 1 1
1
2

AHU LOW PRESSURE
 19800.0  2000.00 12.5
1
01 01 31 12 1 1 1 1 1 1 1 1
1
3

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
4

AHU HIGH PRESSURE
232000 22700.00 125.0
```

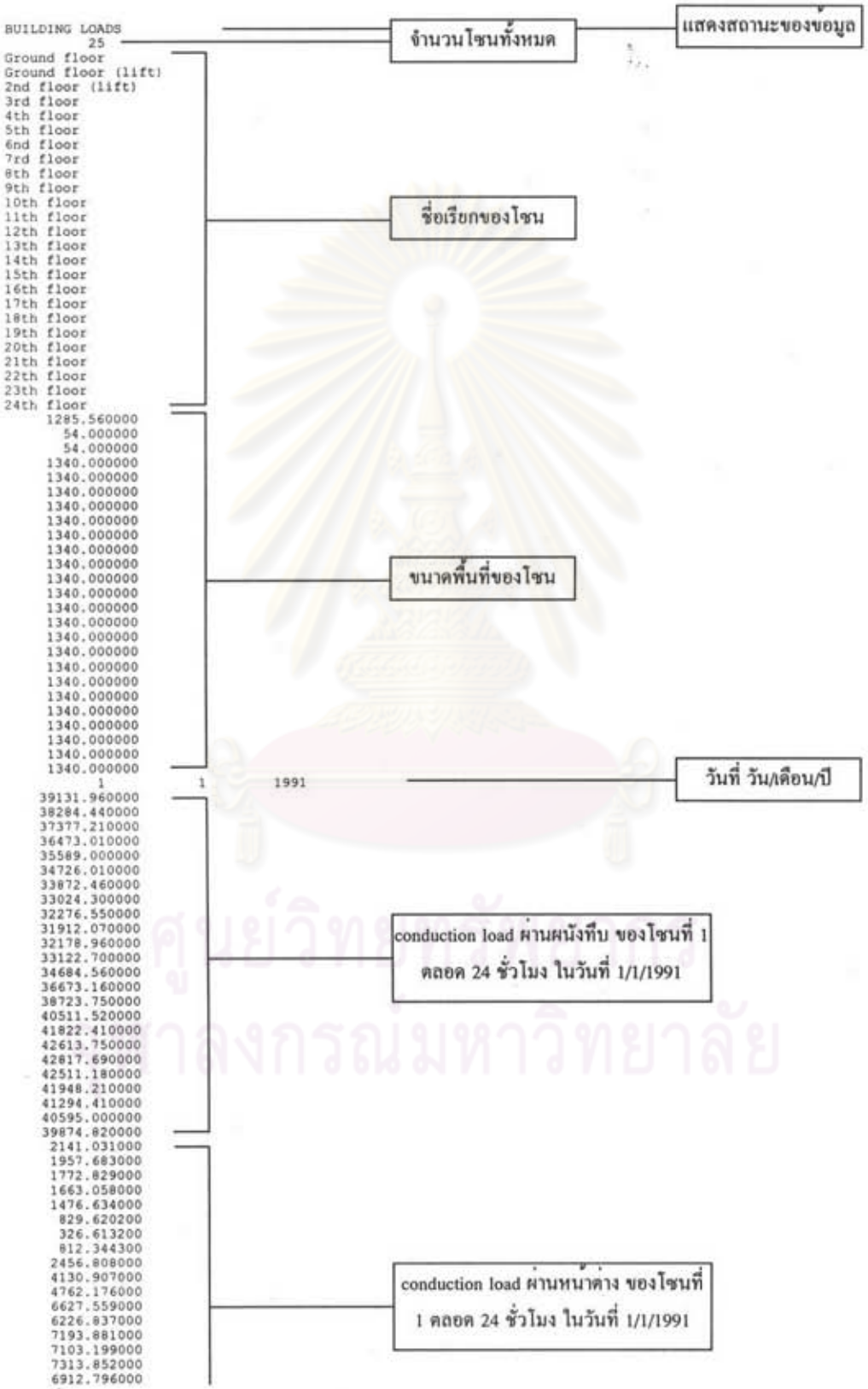
```
1
01 01 31 12 4 2 2 2 2 2 3 4
1
5

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
6

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
7

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
8

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
9

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
10

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
11

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
12

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
13

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
14

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
15

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
16

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
17

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
18

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
19

AHU HIGH PRESSURE
```

```
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
20

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
21

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
22

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
23

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
24

AHU HIGH PRESSURE
232000 22700.00 125.0
1
01 01 31 12 4 2 2 2 2 2 3 4
1
25

< NUMBER_OF_PLANT
  NAME_OF_PLANT_1
  NAME_OF_PLANT_2
        .
        .
        .

>
1
CHILLER PLANT

< Input details of plant :
  TYPE_OF_PLANT
  MAX_CAPACITY
  NUMBER_SCHDULE
     STARTING_DAY   STARTING_MONTH   ENDDING_DAY   ENDDING_MONTH   SCHED_OF_SUN   SCHED_OF_MON ... SCHED_OF_HOL
           .
           .
           .
  NUMBER_OF_SERVED_SYSTEM
     SERVED_SYSTEM_NUMBER
              .
              .
              .

>
WATER COOLED CHILLER
6540000.00
1
01 01 31 12 1 1 1 1 1 1 1 1
24
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
```

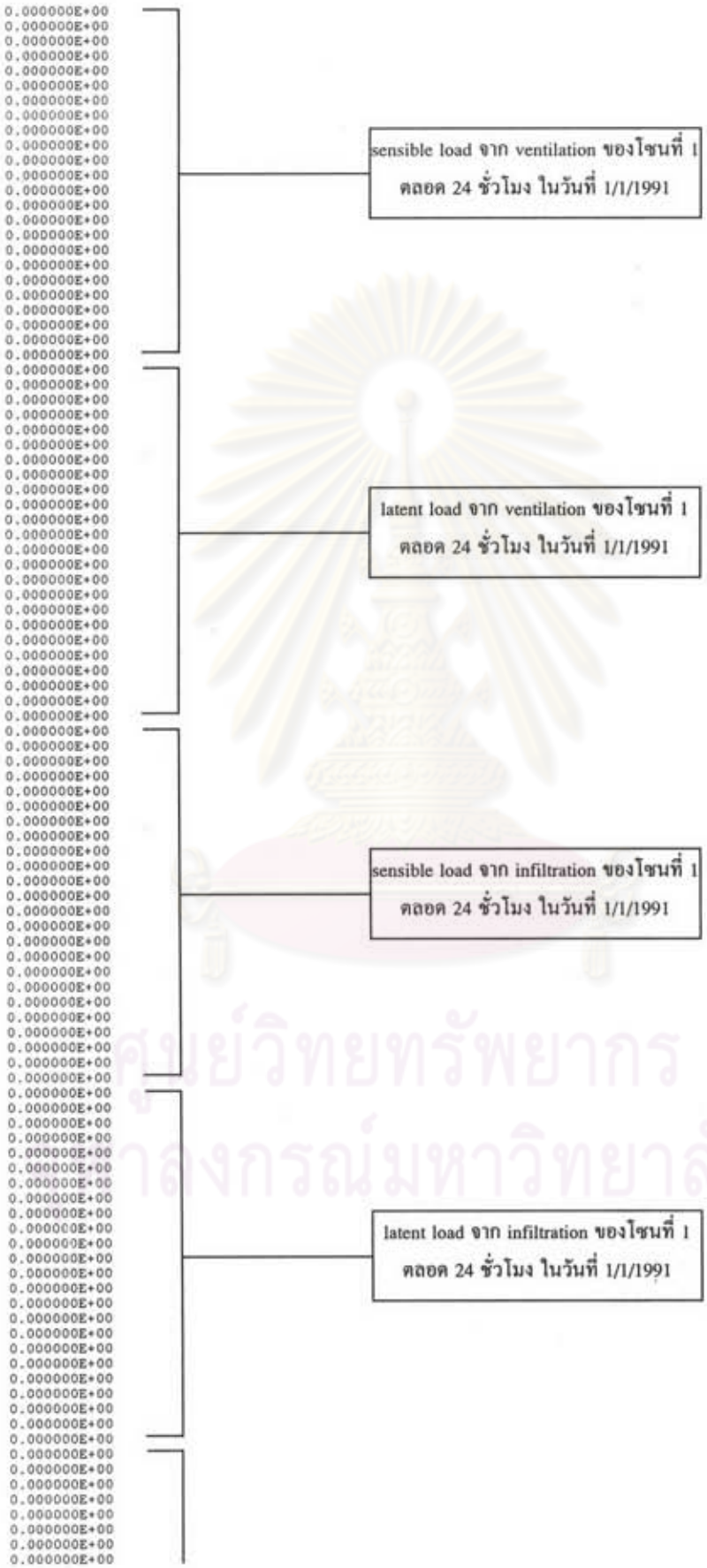
ภาคผนวก ง

OUTPUT FILE (อาคารสำนักงานใหญ่การปิโตรเลียมแห่งประเทศไทย)

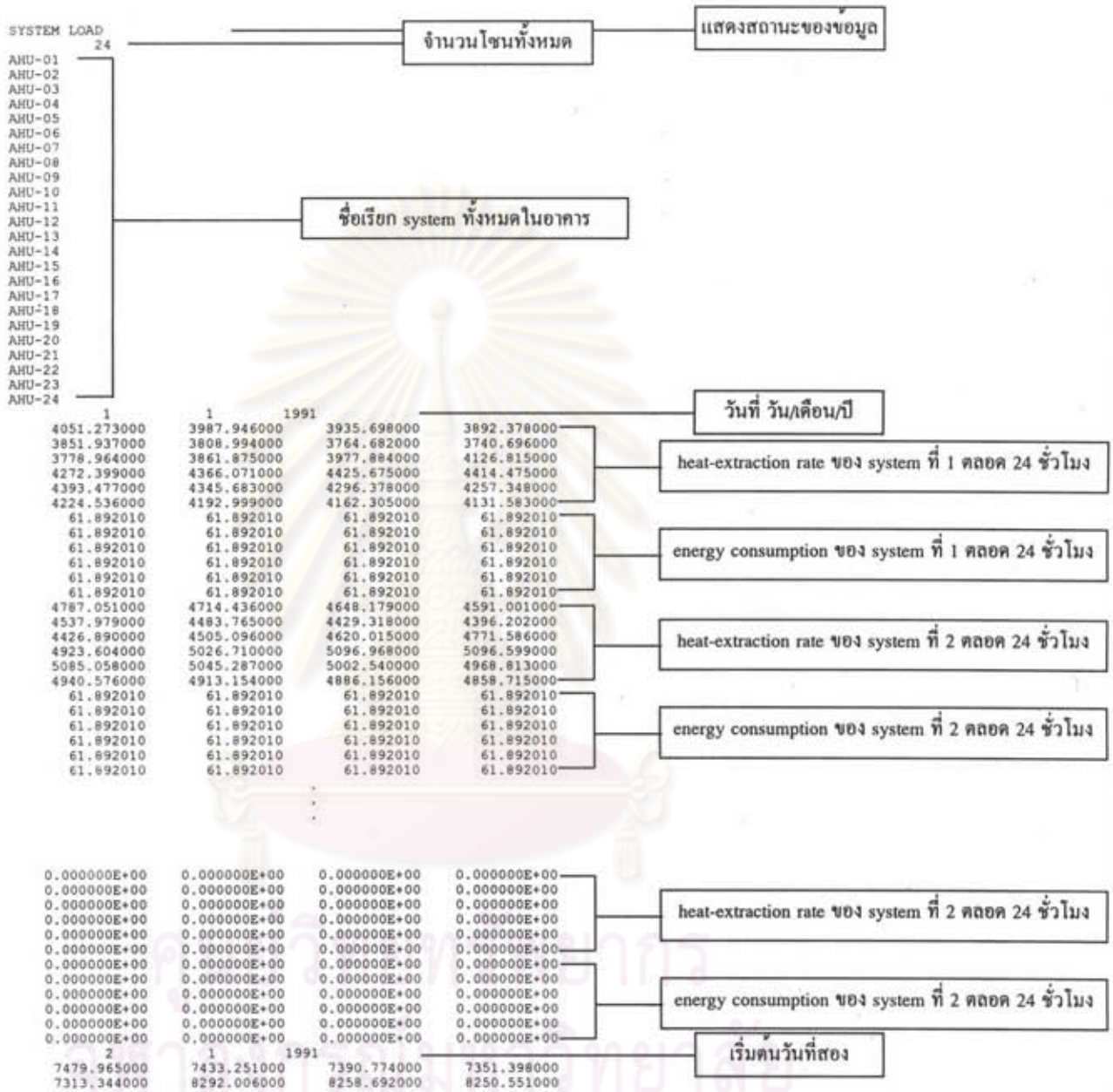
OUTPUT FILE ที่ได้จากโปรแกรม CAL.EXE



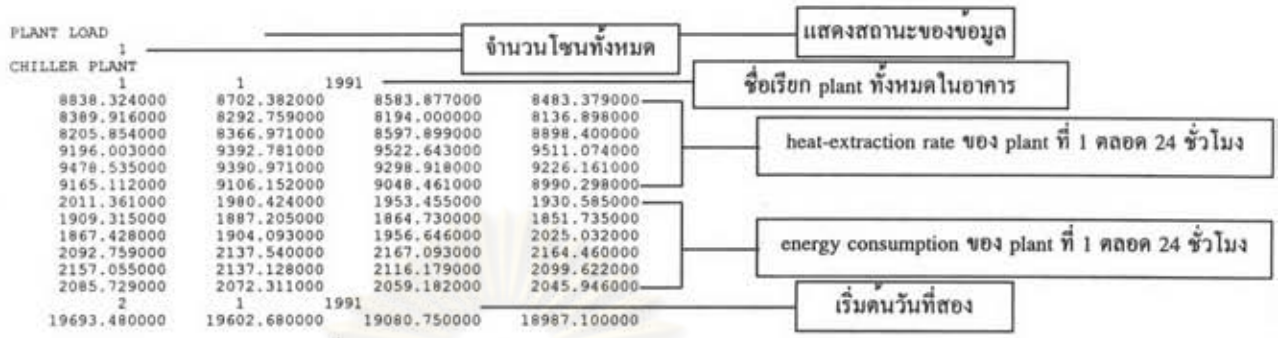

5124.447000
3992.360000
3605.777000
3365.976000
2969.611000
2796.906000
2698.847000
31367.010000
30269.160000
29209.750000
28187.400000
27200.840000
26248.810000
25330.100000
26568.560000
33729.420000
41029.280000
49428.310000
57811.180000
64187.810000
61394.520000
59460.120000
50073.860000
46570.720000
41732.950000
38842.730000
37483.220000
36171.310000
34905.320000
33683.630000
32504.710000
0.000000E+00
solar load ผ่านหน้าต่าง ของโซนที่ 1 ตลอด 24 ชั่วโมง ในวันที่ 1/1/1991
sensible load จากคน ของโซนที่ 1 ตลอด 24 ชั่วโมง ในวันที่ 1/1/1991
latent load จากคน ของโซนที่ 1 ตลอด 24 ชั่วโมง ในวันที่ 1/1/1991
sensible load จากแสงไฟ ของโซนที่ 1 ตลอด 24 ชั่วโมง ในวันที่ 1/1/1991



OUTPUT FILE สำหรับ SYSTEM ที่ได้จากโปรแกรม SYSTEM.EXE



OUTPUT FILE สำหรับ PLANT ที่ได้จากโปรแกรม PLANT.EXE



ภาคผนวก จ

ตารางข้อมูลดิบ

ตาราง จ-1.1 แสดงค่า Total cooling load ที่คำนวณได้จากโปรแกรม BLN-ESP1 และ TMW-CL1 (กราฟ 4.1.1.1 - 4.1.1.12)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Total cooling load from BLN-ESP1 (kW)	Jan	716	700	686	672	661	717	1113	2018	2091	2219	2305	1292	2401	2453	2488	2498	1553	1176	1140	863	827	790	758	734
	Feb	750	733	718	704	692	747	1162	2062	2135	2261	2343	1323	2430	2487	2524	2539	1600	1240	1178	902	865	827	793	769
	Mar	779	762	746	732	719	774	1228	2106	2177	2303	2380	1362	2468	2522	2565	2578	1638	1283	1212	935	897	859	823	798
	Apr	786	768	753	738	725	780	1257	2120	2189	2316	2391	1382	2482	2535	2577	2587	1644	1293	1220	943	905	866	830	805
	May	781	764	748	734	721	776	1258	2116	2184	2312	2392	1376	2480	2531	2571	2579	1636	1293	1215	938	900	861	826	800
	Jun	780	762	747	732	719	774	1251	2110	2180	2308	2389	1371	2477	2528	2566	2574	1633	1298	1213	935	898	859	824	799
	Jul	785	767	752	737	724	779	1250	2113	2184	2313	2391	1378	2482	2532	2572	2582	1641	1307	1218	941	904	865	830	804
	Aug	786	768	753	738	725	780	1250	2117	2188	2317	2394	1383	2485	2539	2580	2589	1645	1295	1221	943	905	866	831	805
	Sep	766	748	733	719	706	761	1234	2101	2171	2298	2383	1362	2472	2525	2566	2573	1621	1240	1199	921	882	844	809	784
	Oct	746	729	714	701	689	744	1211	2078	2145	2276	2361	1347	2452	2507	2541	2546	1581	1214	1176	897	860	823	789	764
	Nov	722	706	691	678	666	722	1168	2042	2108	2243	2331	1320	2426	2477	2507	2509	1539	1185	1147	869	833	796	763	740
	Dec	711	695	681	668	656	712	1125	2016	2085	2220	2308	1299	2407	2456	2488	2491	1532	1170	1133	856	820	784	752	729
Total cooling load from TMW-CL1 (kW)	Jan	568	536	495	449	426	879	1434	2098	2202	2214	2191	1716	2324	2480	2611	2611	1991	1563	1025	832	758	698	650	620
	Feb	584	549	509	462	439	966	1537	2186	2260	2240	2211	1736	2340	2498	2660	2696	2097	1664	1053	855	778	717	667	636
	Mar	596	557	520	472	450	1033	1602	2223	2260	2204	2175	1702	2299	2451	2641	2719	2153	1732	1066	866	789	727	678	647
	Apr	575	533	500	450	431	1025	1563	2146	2149	2067	2045	1577	2164	2302	2508	2619	2091	1706	1031	835	759	699	653	623
	May	585	542	512	461	443	1030	1538	2093	2074	1977	1962	1498	2077	2203	2415	2547	2046	1692	1027	835	762	703	660	631
	Jun	577	534	506	455	437	1013	1506	2047	2020	1919	1908	1446	2022	2142	2354	2493	2003	1665	1012	822	750	692	651	622
	Jul	583	540	510	460	442	1017	1518	2069	2050	1955	1942	1478	2057	2181	2390	2520	2021	1673	1021	830	757	699	657	628
	Aug	571	530	498	448	428	1004	1529	2106	2109	2029	2011	1544	2129	2263	2464	2574	2050	1675	1020	827	752	693	648	618
	Sep	592	554	517	469	447	1007	1560	2175	2213	2161	2136	1665	2260	2407	2591	2666	2103	1695	1054	857	781	720	672	642
	Oct	581	546	507	460	437	951	1511	2154	2226	2206	2181	1706	2309	2463	2622	2658	2064	1641	1044	848	772	712	663	633
	Nov	567	535	494	448	425	873	1423	2083	2186	2198	2177	1701	2309	2463	2594	2594	1976	1553	1022	829	755	696	648	618
	Dec	556	525	483	439	415	837	1382	2048	2161	2186	2166	1690	2299	2453	2573	2560	1934	1513	1007	816	743	685	637	607

ตาราง จ-1.2 แสดงค่า Conduction load ที่คำนวณได้จากโปรแกรม BLN-ESP1 และ TMW-CL1 (กราฟ 4.1.2.1 - 4.1.2.12)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Conduction cooling load from BLN-ESP1 (kW)	Jan	451	443	436	430	427	427	431	435	443	461	488	518	544	567	583	590	589	580	563	540	517	493	472	459
	Feb	466	458	451	445	442	442	446	450	459	477	505	534	560	582	597	603	602	595	579	557	533	509	488	475
	Mar	479	471	465	459	455	456	460	464	474	494	522	552	577	597	612	617	617	609	593	571	547	523	502	488
	Apr	480	473	466	460	457	457	461	466	478	498	526	556	581	600	614	619	619	610	594	572	548	524	503	489
	May	475	467	461	455	452	452	457	462	474	494	522	551	575	595	609	614	613	604	588	566	543	519	497	484
	Jun	472	465	458	452	449	450	454	459	471	490	518	547	571	591	604	609	608	600	584	563	539	515	494	481
	Jul	474	467	460	454	451	452	456	461	472	492	519	548	572	592	605	611	609	601	586	565	541	517	496	483
	Aug	477	469	463	457	454	454	459	463	474	494	522	552	576	596	610	615	614	606	590	568	545	520	499	486
	Sep	469	461	455	449	445	446	450	455	467	488	516	546	571	591	606	611	611	602	584	561	536	512	491	477
	Oct	463	455	448	442	439	439	444	449	461	481	509	539	565	587	601	607	606	596	577	553	529	505	484	471
	Nov	453	445	439	433	429	430	434	439	450	470	497	527	553	576	591	597	596	585	566	542	519	495	474	461
	Dec	448	440	433	427	424	424	429	432	442	460	487	517	544	567	583	589	588	578	560	536	513	489	469	456
Conduction cooling load from TMW-CL1 (kW)	Jan	447	420	399	356	342	325	319	348	392	441	517	562	631	659	690	695	682	656	612	575	539	505	492	475
	Feb	456	429	408	365	351	334	328	358	401	450	526	571	640	668	699	704	691	665	621	584	548	514	501	484
	Mar	465	438	417	375	361	343	337	367	410	459	536	581	649	678	709	713	700	674	631	594	558	524	511	494
	Apr	449	422	401	359	345	327	321	351	394	443	520	565	634	662	693	697	684	658	615	578	542	508	495	478
	May	466	439	418	375	361	344	338	367	411	460	536	581	650	678	709	713	701	674	631	594	558	524	511	494
	Jun	462	435	414	372	358	340	334	364	407	456	532	578	646	675	706	710	697	671	628	591	555	521	508	491
	Jul	466	439	418	375	361	344	338	367	411	460	536	581	650	678	709	713	701	674	631	594	558	524	511	494
	Aug	449	422	401	359	345	327	321	351	394	443	520	565	634	662	693	697	684	658	615	578	542	508	495	478
	Sep	465	438	417	375	361	343	337	367	410	459	536	581	649	678	709	713	700	674	631	594	558	524	511	494
	Oct	456	429	408	365	351	334	328	358	401	450	526	571	640	668	699	704	691	665	621	584	548	514	501	484
	Nov	447	420	399	356	342	325	319	348	392	441	517	562	631	659	690	695	682	656	612	575	539	505	492	475
	Dec	438	412	391	348	334	317	310	340	383	433	509	554	623	651	682	686	674	647	604	567	531	497	484	467

ตาราง จ-1.3 แสดงค่า Solar load ที่คำนวณได้จากโปรแกรม BLN-ESP1 และ TMW-CL1 (กราฟ 4.1.3.1 - 4.1.3.12)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Solar cooling load from BLN-ESP1 (kW)	Jan	215	207	200	193	186	180	177	269	299	314	329	346	358	366	374	382	369	276	266	256	247	239	230	222
	Feb	232	224	216	208	201	194	208	296	326	337	349	357	369	382	394	407	400	323	287	277	267	258	249	240
	Mar	248	239	231	223	215	207	261	326	352	363	368	380	390	401	420	431	423	352	307	296	286	276	266	257
	Apr	256	247	238	230	222	214	293	341	364	375	378	399	404	416	433	442	431	364	317	306	295	285	275	265
	May	256	247	238	230	222	214	297	340	363	374	383	397	406	416	431	439	428	370	317	306	295	285	275	265
	Jun	257	248	240	231	223	215	293	338	361	374	384	396	407	417	431	438	429	380	319	307	297	286	276	267
	Jul	260	251	243	234	226	218	289	339	364	378	385	402	411	420	436	444	436	387	322	311	300	290	280	270
	Aug	259	250	242	233	225	217	288	341	367	380	385	404	411	423	440	448	436	371	321	310	299	289	279	269
	Sep	245	236	228	220	212	205	278	331	355	365	378	386	401	412	427	433	413	317	303	293	282	273	263	254
	Oct	233	224	217	209	202	195	263	315	336	350	364	379	389	400	408	411	379	298	288	278	268	259	250	241
	Nov	219	212	204	197	190	184	231	290	312	331	348	367	376	382	386	387	350	281	272	262	253	244	236	227
	Dec	212	204	197	190	184	177	191	269	294	314	333	352	364	368	373	374	348	272	262	253	244	236	227	220
Solar cooling load from TMW-CL1 (kW)	Jan	41	38	35	34	32	336	597	760	824	775	556	414	513	753	853	836	717	481	83	69	61	55	48	45
	Feb	42	39	36	35	33	328	544	659	709	686	523	341	486	669	736	726	649	461	87	72	63	57	49	46
	Mar	44	41	37	36	34	351	537	604	582	523	431	235	408	515	602	657	629	474	91	75	66	60	52	48
	Apr	45	42	38	37	35	389	558	588	524	405	257	247	258	397	531	628	638	508	96	79	69	62	54	50
	May	47	43	39	38	36	431	594	606	518	386	296	255	290	373	516	639	671	554	100	82	71	64	55	52
	Jun	47	43	40	38	36	455	620	624	532	406	323	258	311	389	528	656	699	584	101	83	72	65	56	53
	Jul	47	43	40	38	36	432	594	607	520	391	305	260	298	377	518	640	672	556	102	83	72	65	56	53
	Aug	47	43	39	38	36	389	557	587	523	406	265	258	267	399	531	627	637	508	100	82	71	64	55	52
	Sep	45	41	38	36	35	345	530	596	575	518	429	244	407	511	597	650	620	467	94	78	68	61	53	50
	Oct	43	40	36	35	33	324	535	645	694	670	515	340	480	655	720	711	639	454	89	73	64	58	50	47
	Nov	41	39	36	34	33	335	594	754	816	769	553	414	512	748	845	829	713	479	85	70	62	56	49	46
	Dec	40	37	35	33	32	347	620	789	856	774	558	431	514	752	886	868	745	498	81	67	59	54	47	44

ตาราง จ-2 แสดงค่า Heat-extraction rate ที่คำนวณได้จากโปรแกรม BLN-ESP1, TRACE600, BLAST และ การตรวจวัดในวันที่ 6 กค. 2538 (กราฟ 4.2.1 - 4.2.12)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Heat extraction rate from BLN-ESP1 (kW)	Jan	16	16	16	16	16	17	1897	2348	2449	2539	2631	2027	2639	2682	2630	2536	1697	1468	1399	1219	1172	1128	16	16
	Feb	17	17	17	16	16	18	2066	2465	2573	2675	2758	2261	2883	2924	2865	2762	1867	1641	1559	1375	1321	1272	17	17
	Mar	19	19	19	18	18	20	2278	2832	2922	3004	3066	2389	3119	3132	3061	2981	2025	1756	1661	1450	1398	1348	19	19
	Apr	19	18	18	18	18	19	2197	2739	2806	2865	2956	2323	2959	2910	2835	2770	1817	1589	1506	1315	1268	1224	19	19
	May	18	18	18	18	18	19	2203	2780	2854	2927	2970	2301	3008	2945	2895	2813	1887	1633	1546	1338	1290	1246	18	18
	Jun	17	17	17	17	17	18	2019	2562	2628	2687	2761	2118	2762	2775	2717	2620	1750	1515	1438	1243	1199	1153	17	17
	Jul	17	16	16	16	16	18	1955	2431	2477	2541	2594	2020	2668	2654	2616	2550	1625	1402	1330	1151	1109	1069	17	17
	Aug	17	17	16	16	16	18	1981	2523	2582	2648	2740	2124	2809	2796	2720	2649	1748	1510	1437	1243	1199	1157	17	17
	Sep	17	17	16	16	16	18	1937	2458	2550	2637	2731	2108	2761	2717	2707	2604	1641	1408	1345	1157	1113	1070	17	17
	Oct	16	16	16	16	16	17	1869	2394	2484	2593	2698	2048	2671	2618	2571	2495	1626	1389	1338	1154	1109	1067	16	16
	Nov	14	14	14	14	14	15	1642	2116	2240	2351	2425	1865	2516	2569	2481	2412	1552	1320	1269	1094	1049	1008	15	15
	Dec	13	13	13	13	12	14	1366	1760	1881	1969	1993	1521	2013	2119	2066	2013	1267	1075	1038	893	857	822	13	13
Heat extraction rate from TRACE600 (kW)	Jan	7	7	7	7	7	7	3547	2406	2720	2681	2648	2030	1276	2141	2127	2177	2160	1410	1189	7	7	7	7	7
	Feb	7	7	7	7	7	7	3548	2317	2668	2595	2551	2009	1266	2099	2089	2124	2109	1402	1190	7	7	7	7	7
	Mar	7	7	7	7	7	7	3580	2709	2940	2901	2845	2832	2188	2240	2392	2398	2387	1650	1441	7	7	7	7	7
	Apr	7	7	7	7	7	7	3589	2758	2956	2921	2866	2852	2373	2206	2365	2337	2354	1648	1448	7	7	7	7	7
	May	7	7	7	7	7	7	3589	3000	3190	3181	3174	3196	3267	3329	3326	3328	3277	2599	1779	7	7	7	7	7
	Jun	7	7	7	7	7	7	3575	2990	3216	3121	3248	3317	3259	3285	3261	2675	2714	1854	1617	7	7	7	7	7
	Jul	7	7	7	7	7	7	3570	2818	3071	2935	2983	2778	964	2640	2433	2607	2569	1758	1478	7	7	7	7	7
	Aug	7	7	7	7	7	7	3560	2761	3052	3056	3116	3107	1977	2611	2521	2657	2535	1731	1481	7	7	7	7	7
	Sep	7	7	7	7	7	7	3566	2622	2899	2864	2934	2924	1457	2357	2366	2423	2501	1717	1514	7	7	7	7	7
	Oct	7	7	7	7	7	7	3550	2487	2778	2732	2489	1845	1315	2180	2026	2070	2063	1288	1125	7	7	7	7	7
	Nov	7	7	7	7	7	7	3532	2277	2638	2614	2472	1876	1389	2261	2196	2199	2097	1310	1125	7	7	7	7	7
	Dec	7	7	7	7	7	7	3305	2182	2480	2483	2010	1789	1249	2061	2088	2148	2069	1264	1065	7	7	7	7	7

ตาราง จ-2 (ต่อ) แสดงค่า Heat-extraction rate ที่คำนวณได้จากโปรแกรม BLN-ESP1, TRACE600, BLAST และ การตรวจวัดในวันที่ 6 กค. 2538 (กราฟ 4.2.1 - 4.2.12)

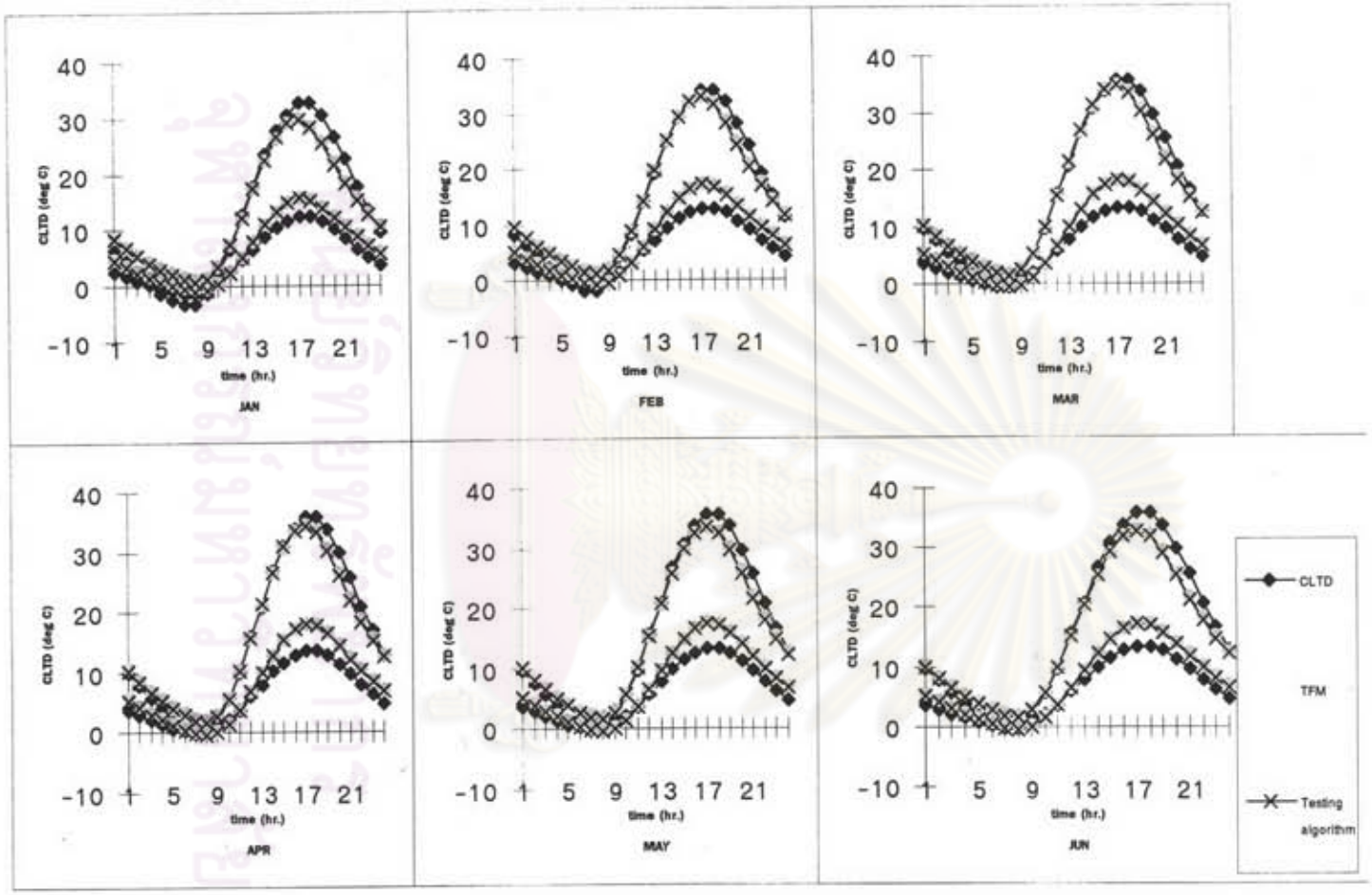
	Jan	11	10	10	10	9	24	544	1038	1566	1928	2202	1981	2366	2402	2419	2326	1555	1146	848	624	551	490	11	11
	Feb	10	10	9	9	8	20	427	939	1507	1886	2140	1986	2445	2504	2485	2403	1612	1212	907	670	581	533	11	10
	Mar	14	14	14	14	13	57	1209	1884	2445	2837	3012	2683	3083	3107	3151	3082	2048	1605	1228	1013	907	851	15	14
Heat	Apr	15	15	14	14	14	68	1365	2085	2521	2854	3064	2731	3176	3225	3232	3147	2227	1754	1369	1140	1054	973	15	15
extraction	May	15	15	14	14	14	55	1259	1988	2442	2724	2893	2619	3044	3000	2874	2772	1943	1658	1379	1138	1073	1012	16	15
rate	Jun	13	13	13	12	12	57	1139	1781	2187	2484	2633	2308	2660	2737	2588	2424	1584	1314	1110	912	835	752	14	13
from	Jul	13	13	12	12	12	48	1078	1662	2040	2328	2532	2250	2735	2809	2791	2636	1751	1359	1104	881	811	705	13	13
BLAST	Aug	12	12	11	11	11	44	902	1554	1953	2309	2544	2202	2643	2637	2518	2431	1469	1116	900	692	635	579	12	12
(kW)	Sep	12	11	11	11	10	33	840	1444	1968	2299	2475	2125	2455	2400	2456	2263	1438	1107	894	660	610	542	12	12
	Oct	10	10	10	10	9	29	635	1364	1837	2199	2441	2126	2467	2440	2367	2261	1398	1001	784	591	512	445	11	11
	Nov	8	8	8	7	7	16	332	893	1440	1846	2093	1944	2351	2396	2383	2192	1305	944	736	552	472	391	9	9
	Dec	7	7	6	6	5	12	200	618	1169	1521	1782	1646	1967	2030	2035	1952	1250	877	657	462	379	313	8	7
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
Heat	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
extraction	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
rate	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
from	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
audit	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
(kW)	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112
	6/7/2538	107	119	112	116	120	124	3918	4483	3875	4176	4154	3923	3668	3714	3998	3959	3289	3131	2133	2012	320	112	119	112

ตาราง จ-3 แสดงค่า Energy consumption รายเดือนที่คำนวณได้จากโปรแกรม BLN-ESP1, BLAST และ ข้อมูลจริง

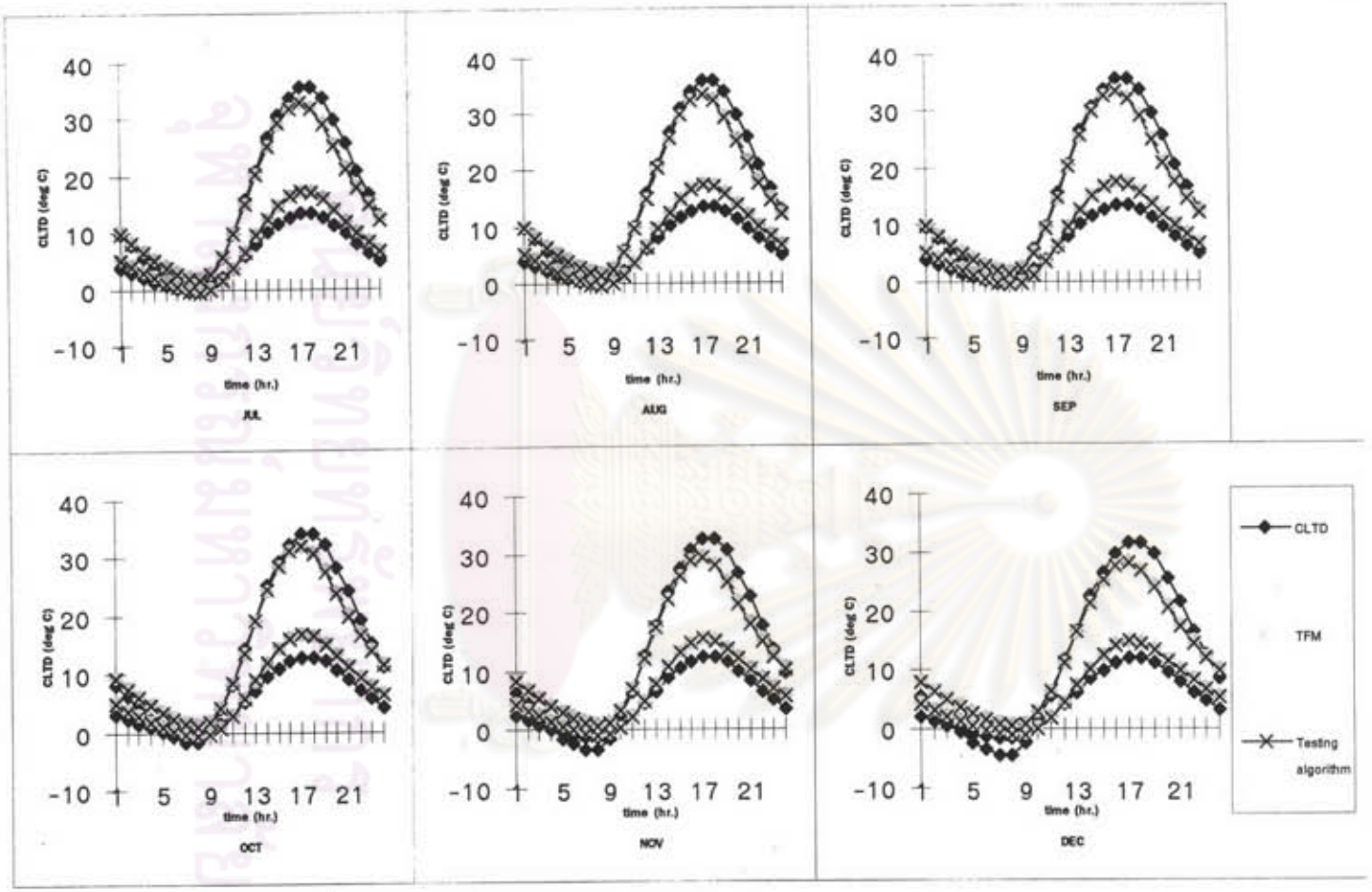
	BLN-ESP1	BLAST	audit
Jan	583,217	548,655	670,000
Feb	536,011	483,745	631,000
Mar	626,706	599,925	727,000
Apr	594,062	601,536	675,000
May	617,663	604,705	728,000
Jun	569,548	550,226	782,000
Jul	580,808	578,668	762,000
Aug	597,395	569,238	727,000
Sep	559,951	530,664	751,000
Oct	581,366	555,642	695,000
Nov	534,507	503,634	669,000
Dec	520,351	513,244	652,000
sum	6,901,585	6,639,879	8,469,000

ภาคผนวก ฉ

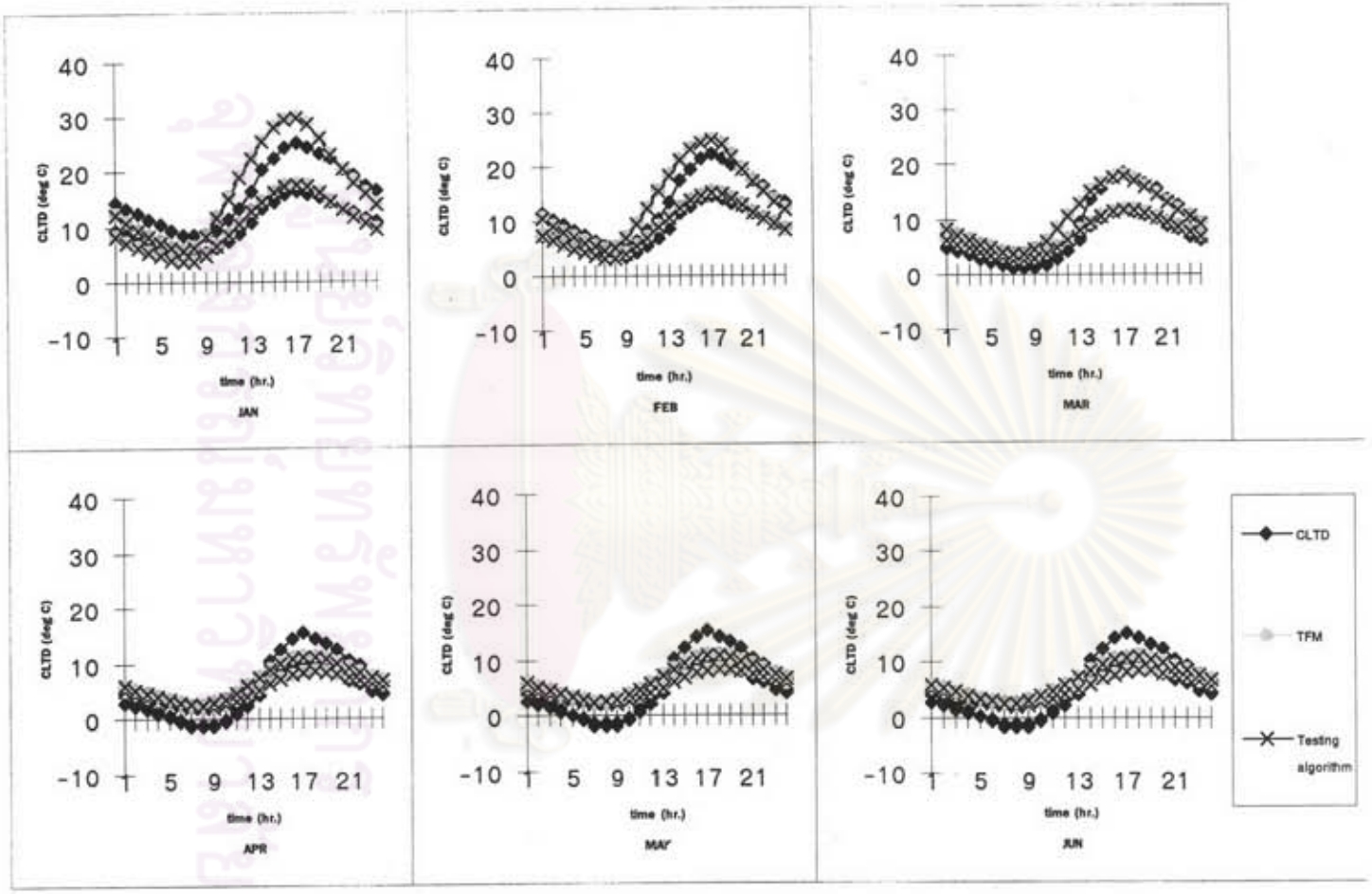
กราฟเปรียบเทียบค่า CLTD และ SCL
จากการคำนวณด้วยวิธี CLTD/CLF และวิธี Transfer function method



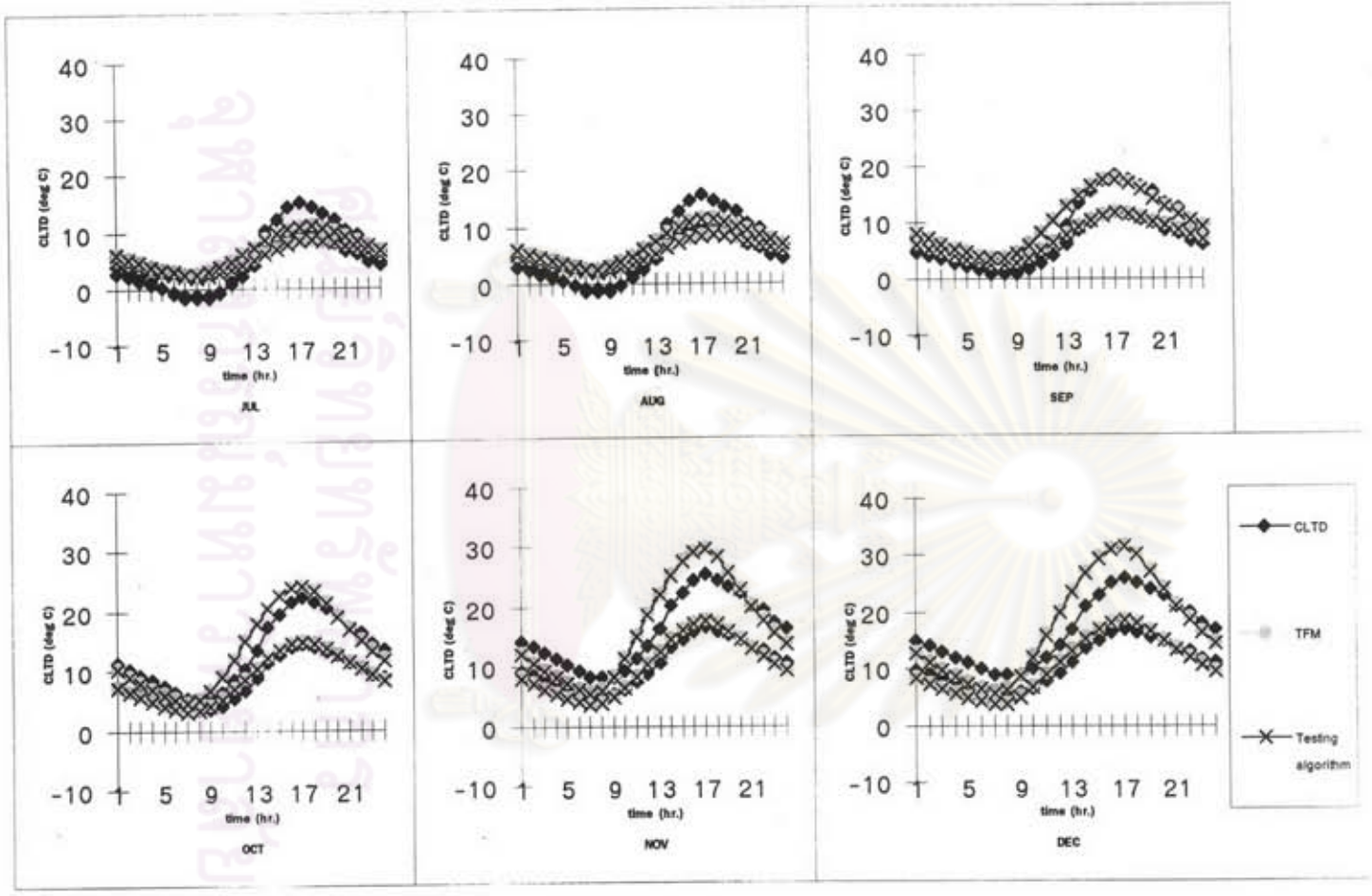
กราฟเปรียบเทียบค่า horrizontal CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 1 - 6



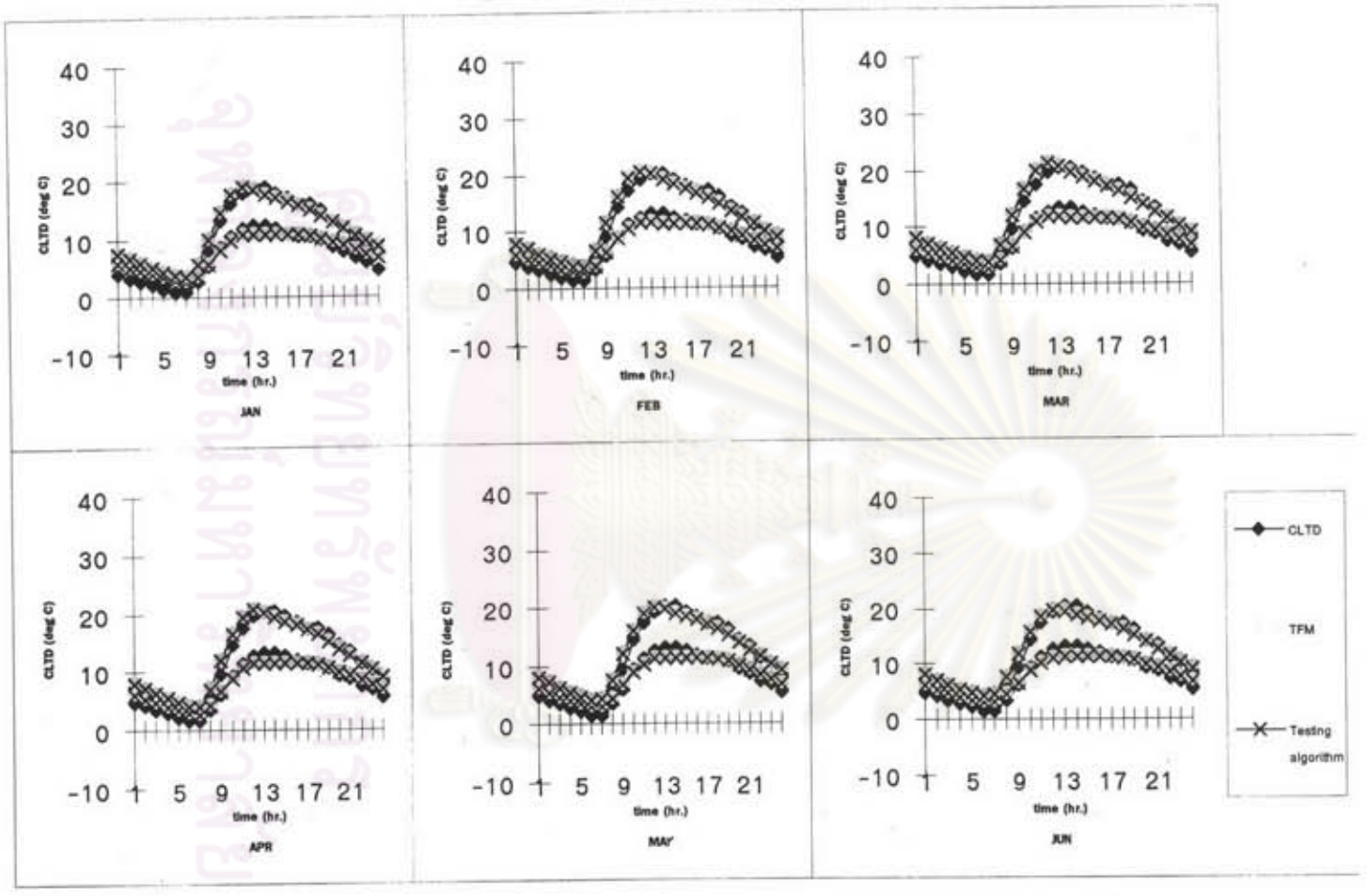
กราฟเปรียบเทียบค่า horrizontal CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 7 - 12



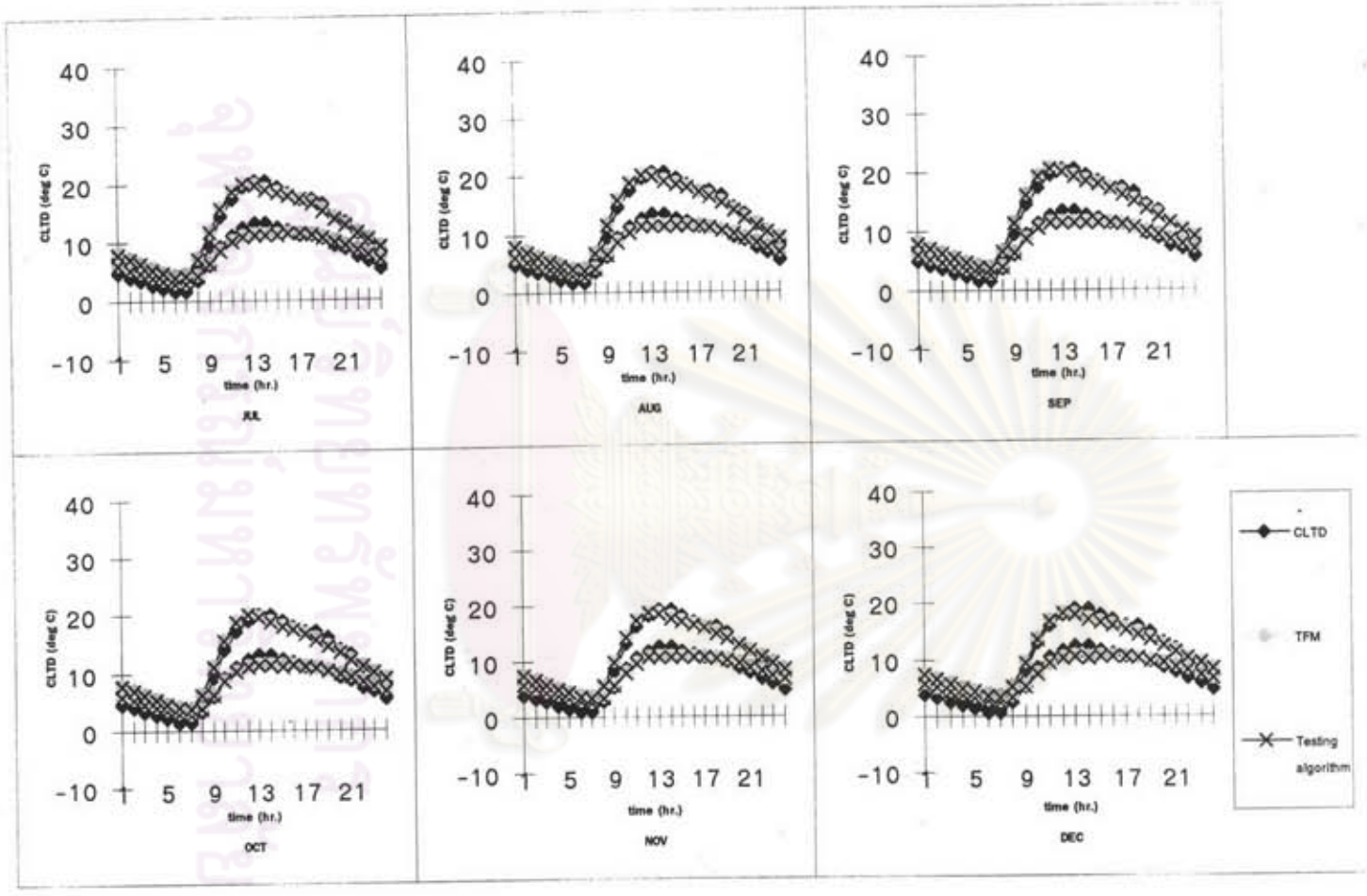
กราฟเปรียบเทียบค่า south CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 1 - 6



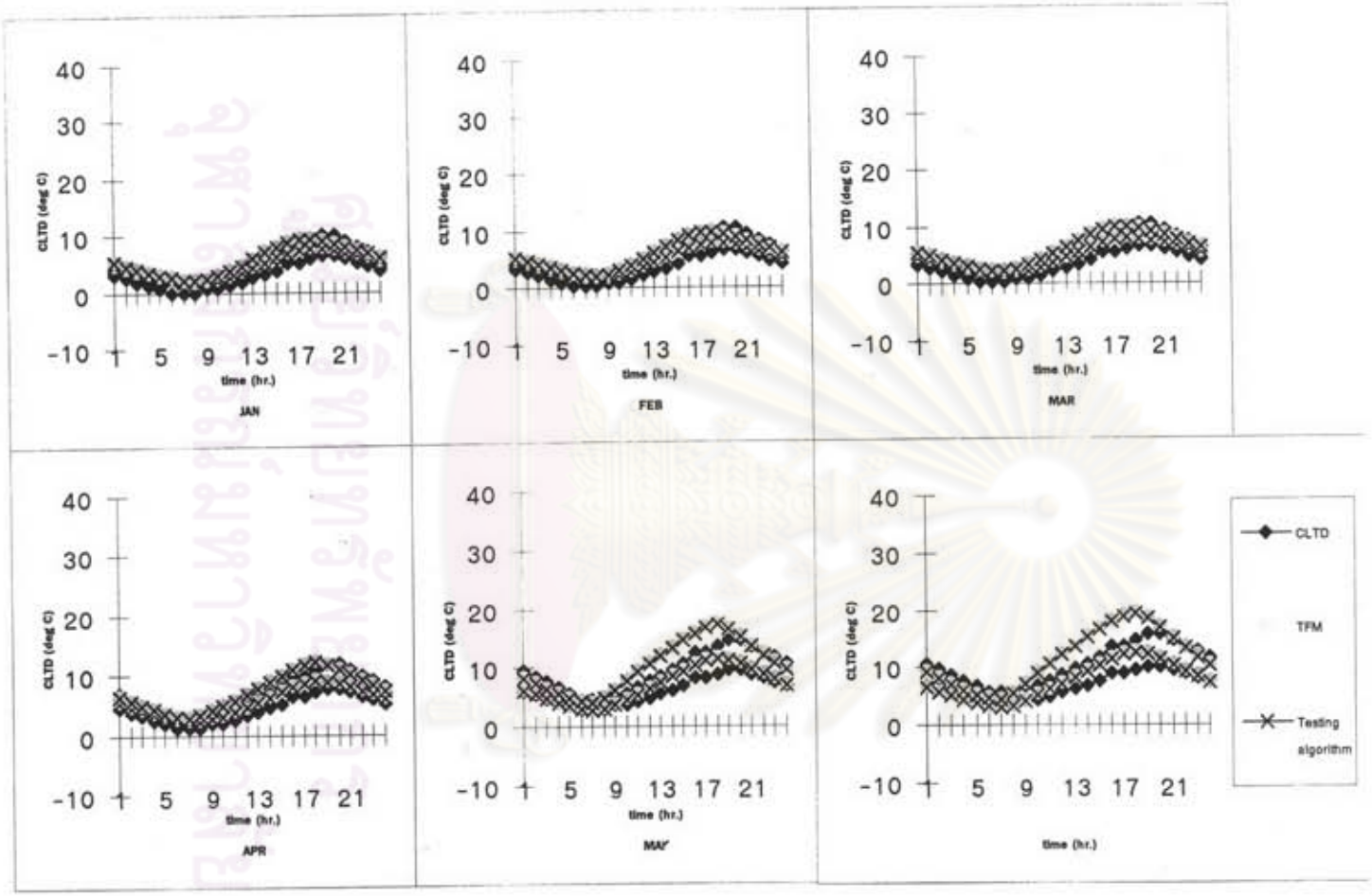
กราฟเปรียบเทียบค่า south CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 7 - 12



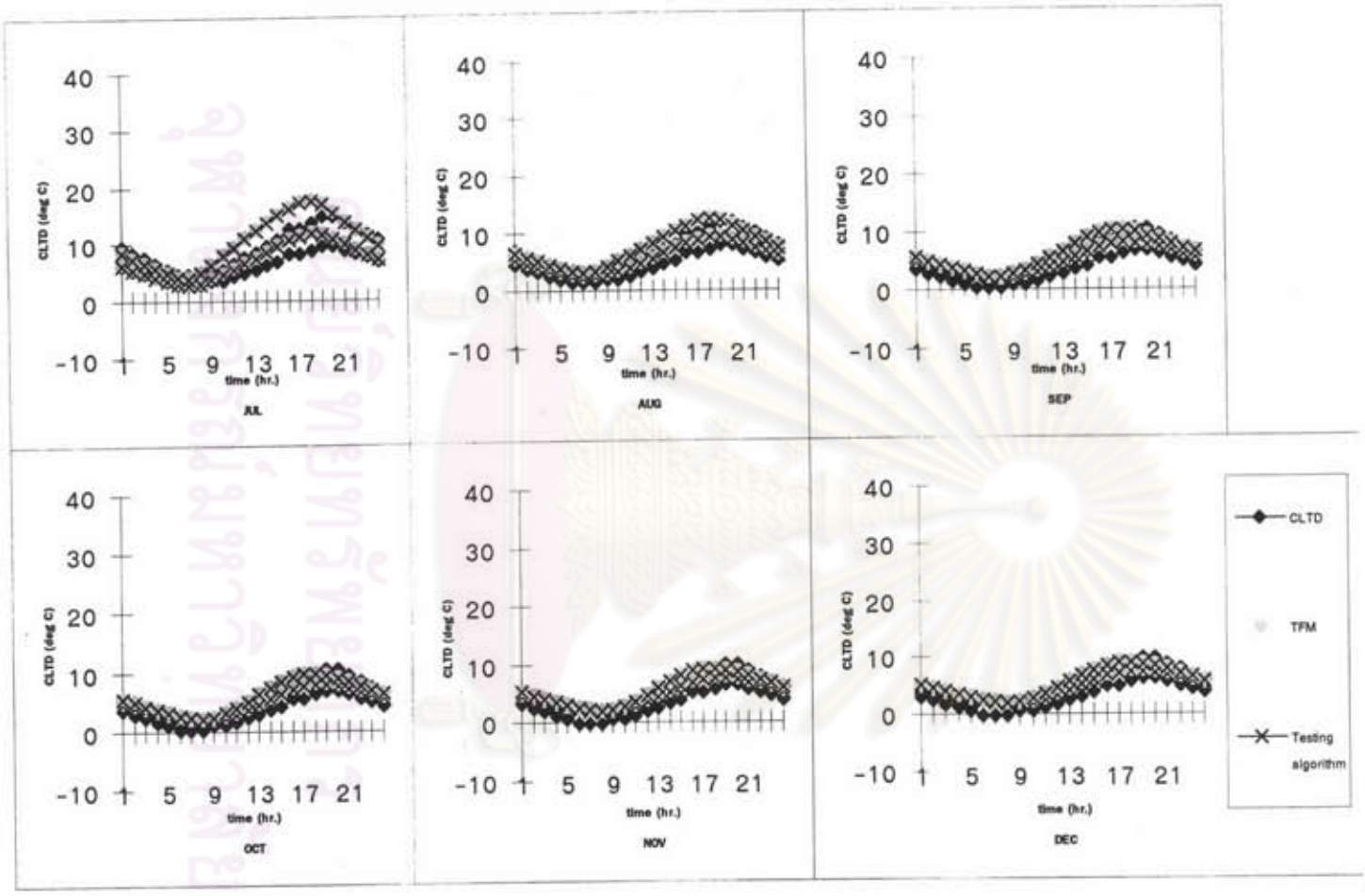
กราฟเปรียบเทียบค่า east CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 1 - 6



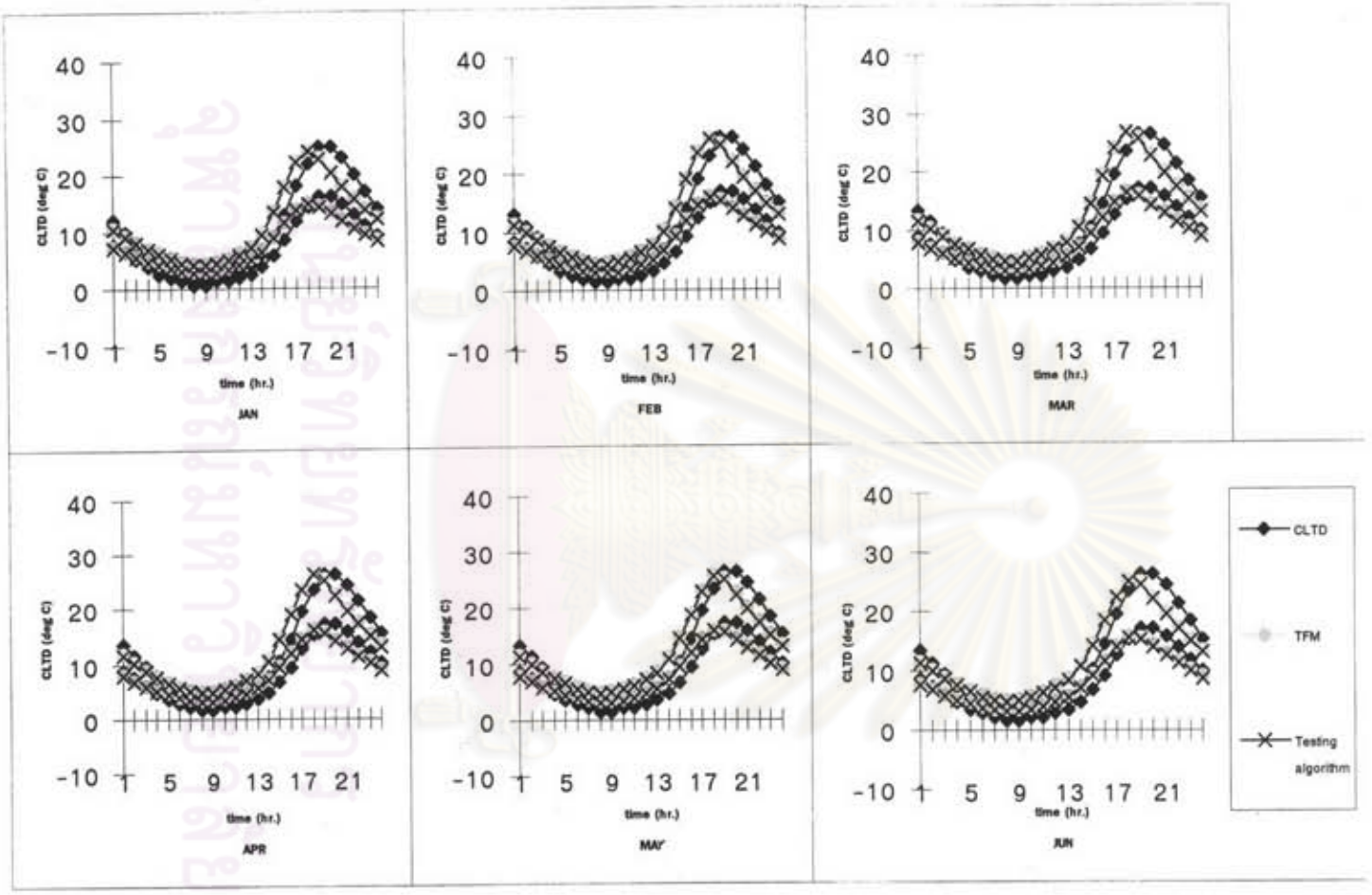
กราฟเปรียบเทียบค่า east CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 7 - 12



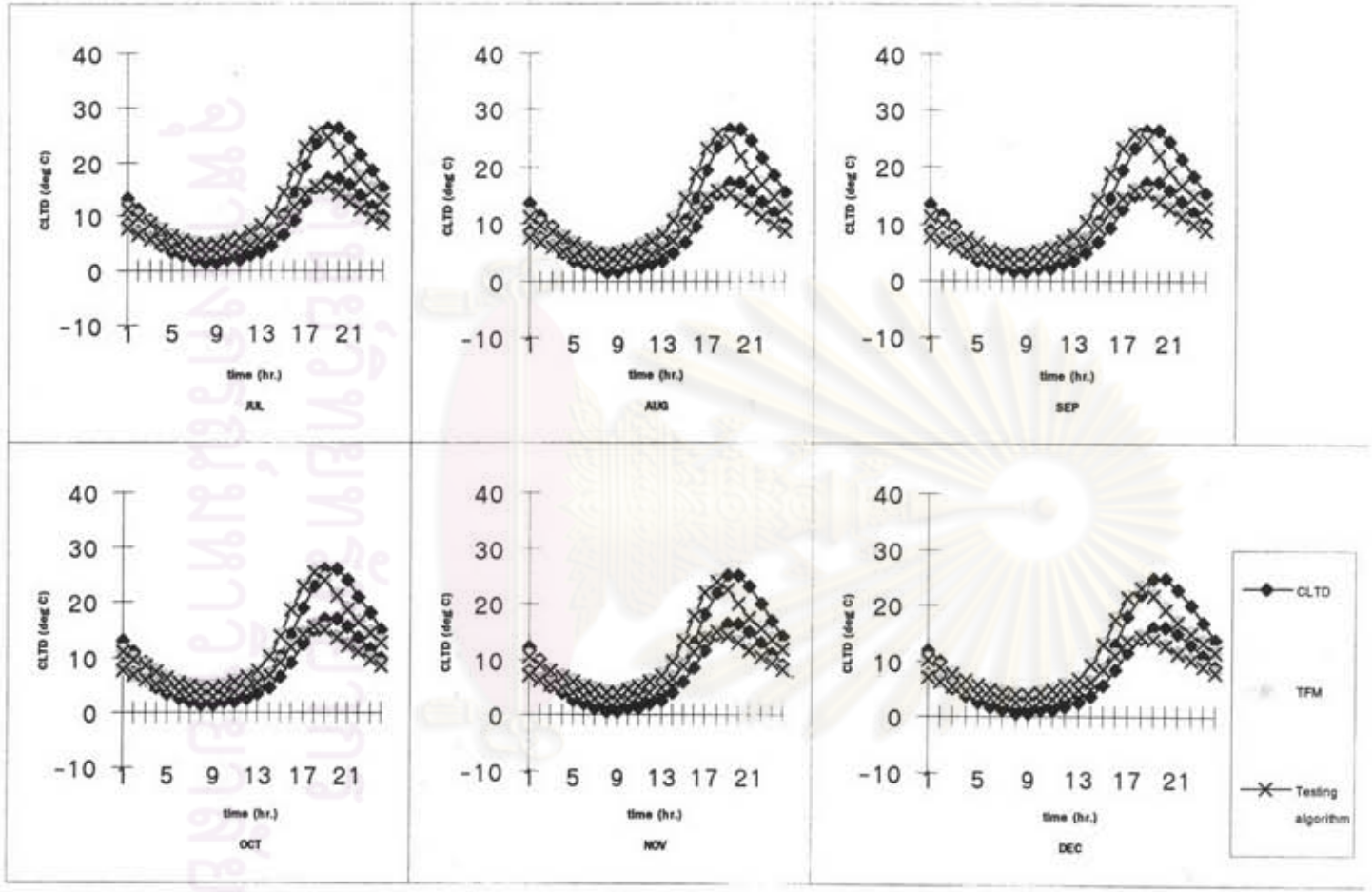
กราฟเปรียบเทียบค่า north CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 1 - 6



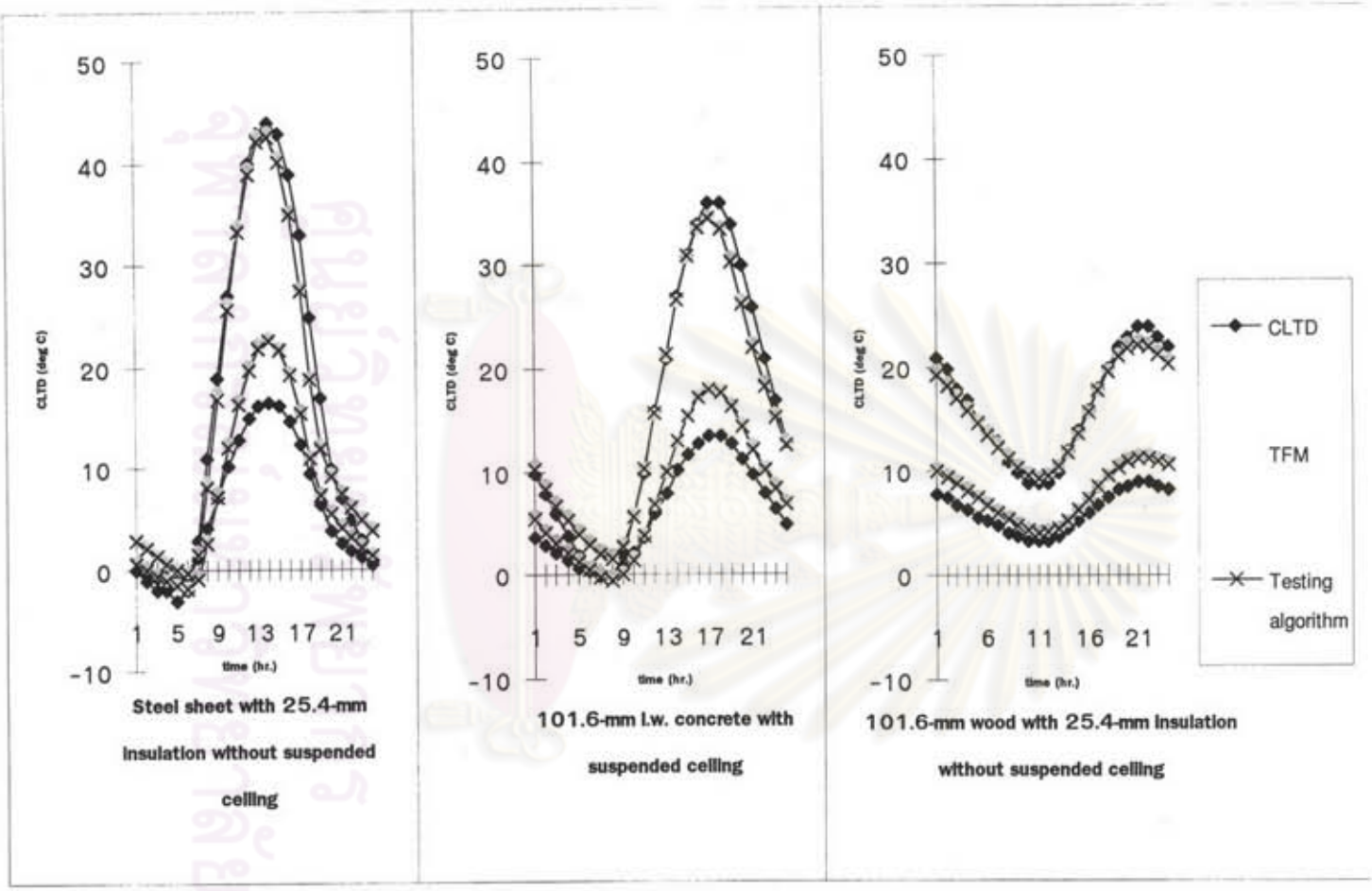
กราฟเปรียบเทียบค่า north CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 7 - 12



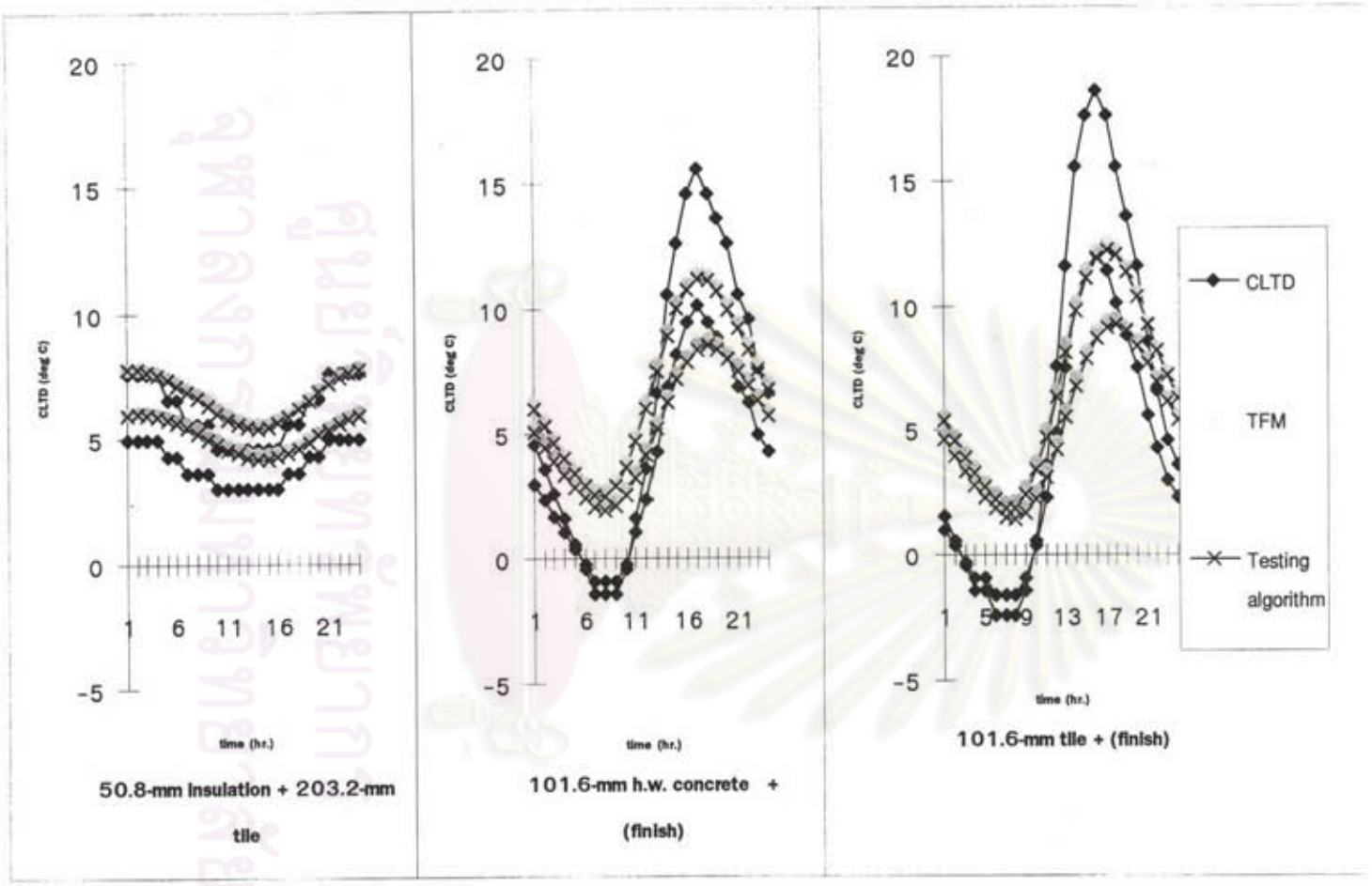
กราฟเปรียบเทียบค่า west CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 1 - 6



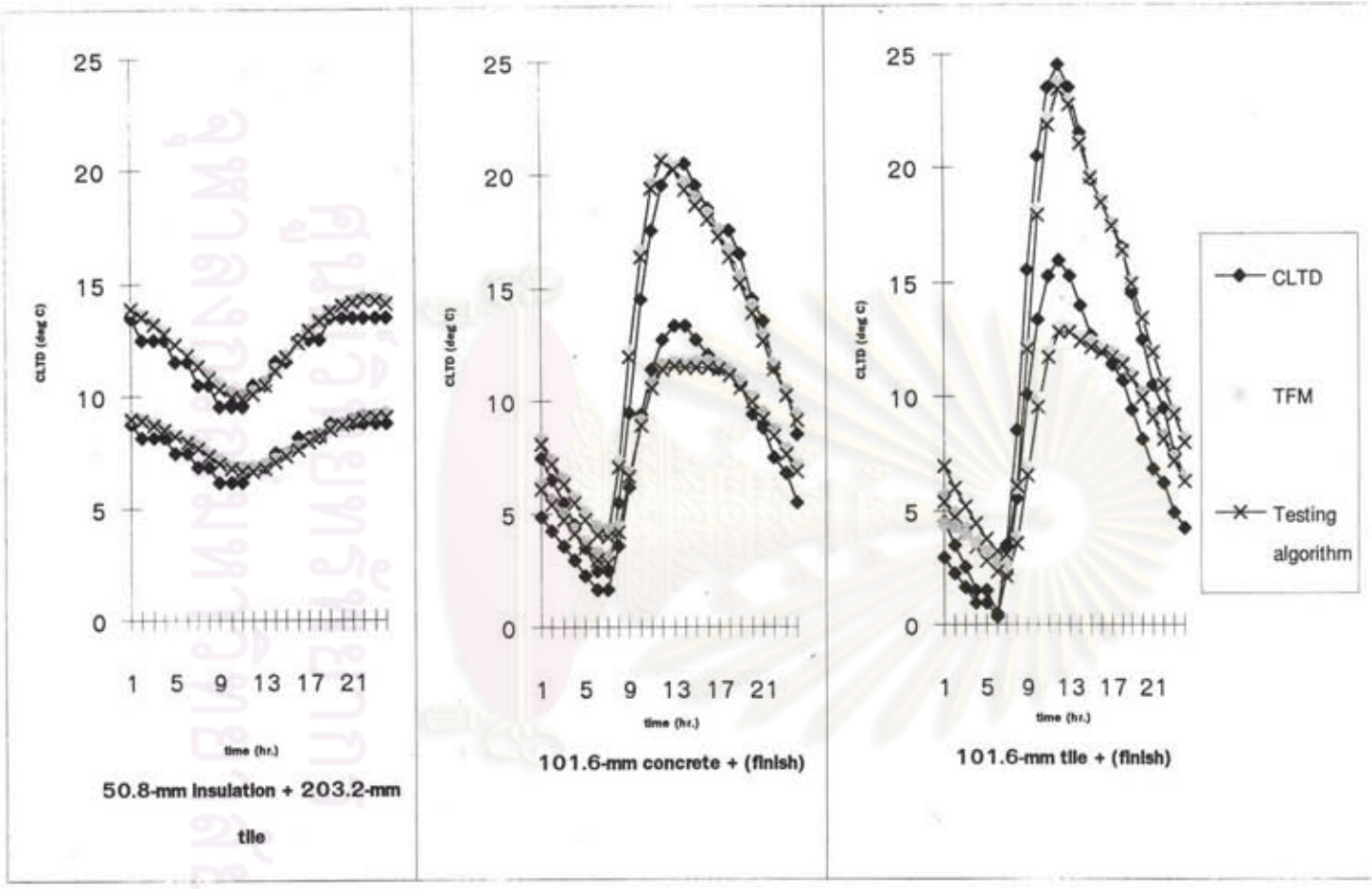
กราฟเปรียบเทียบค่า west CLTD ที่คำนวณด้วยวิธี CLTD, TFM และ testing algorithm ในเดือนที่ 7 - 12



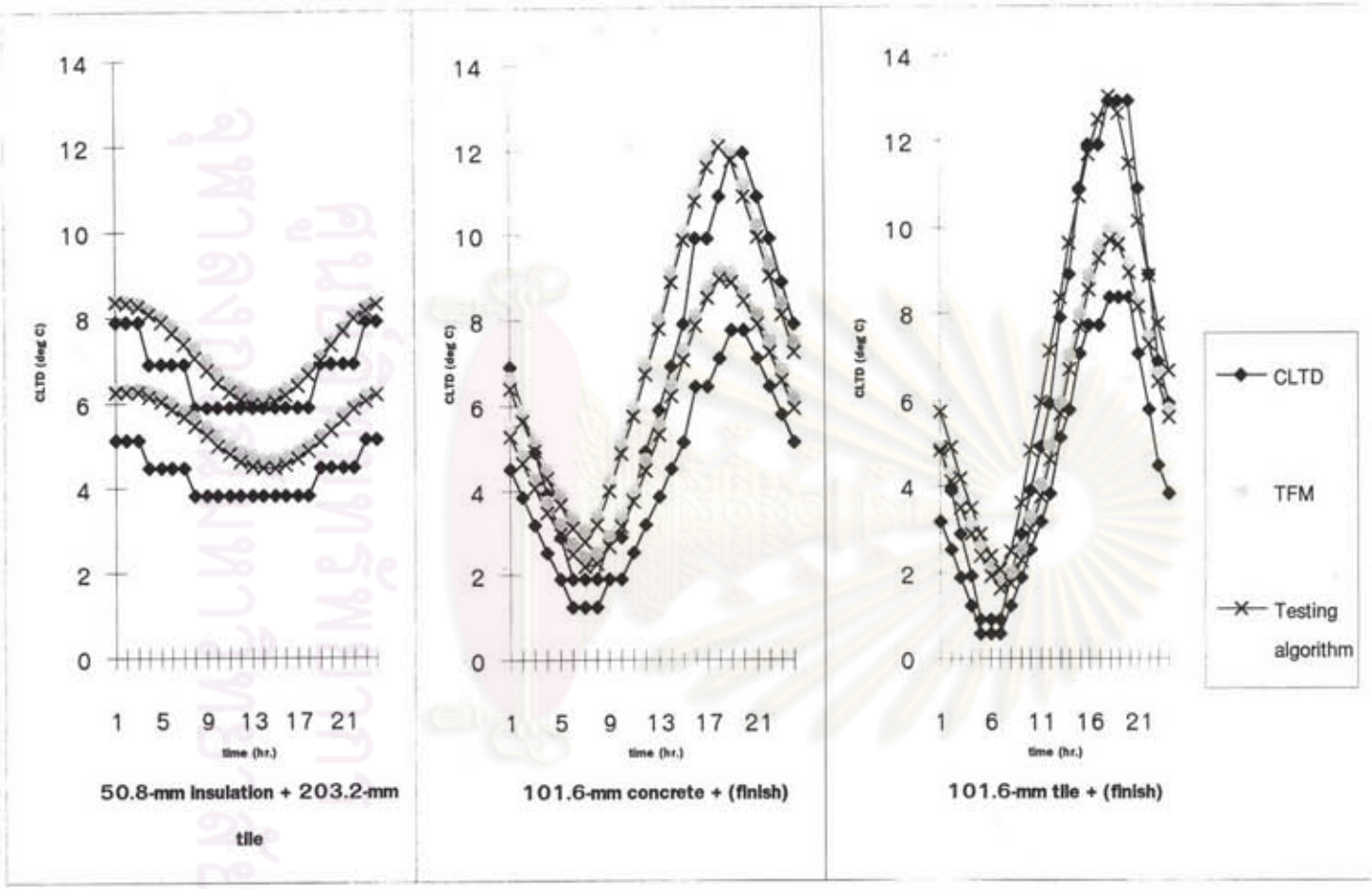
กราฟแสดงการเปรียบเทียบค่า horrizontal CLTD ของหลังคา 3 ชนิด ซึ่งคำนวณโดยวิธีต่างๆ กัน



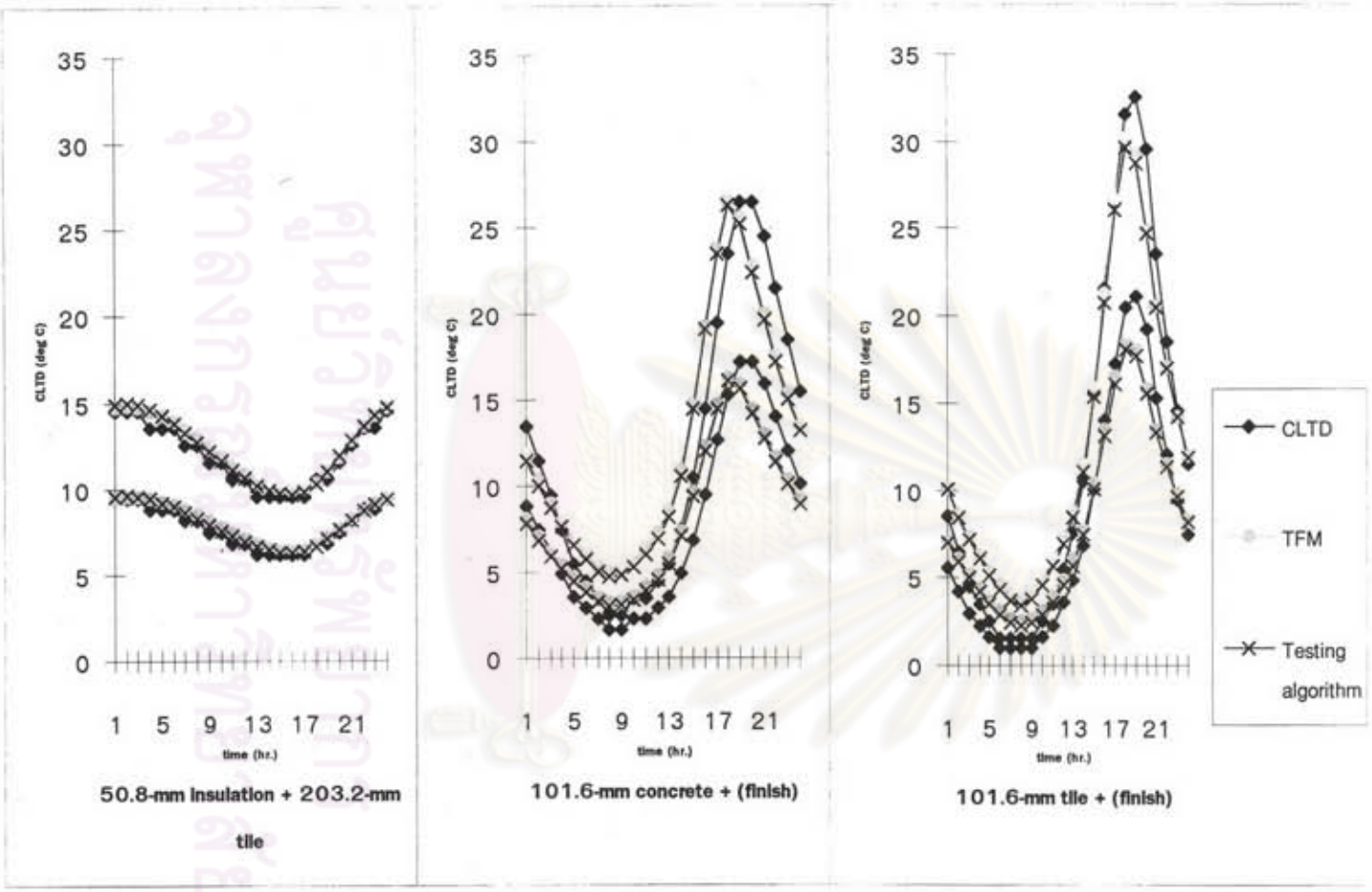
กราฟแสดงการเปรียบเทียบค่า south CLTD ของกำแพง 3 ชนิด ซึ่งคำนวณโดยวิธีต่างๆ กัน



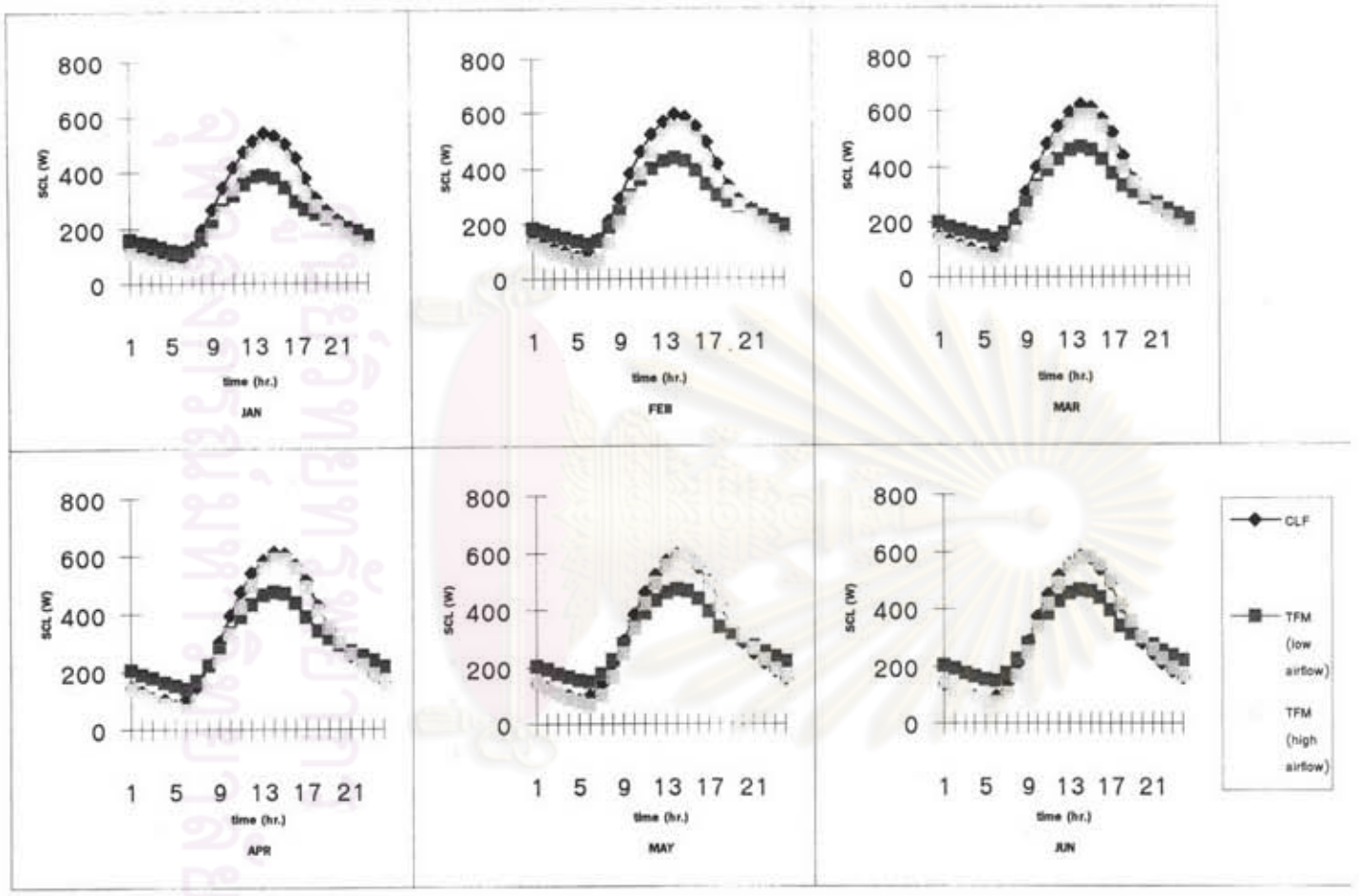
กราฟแสดงการเปรียบเทียบค่า east CLTD ของกำแพง 3 ชนิด ซึ่งคำนวณโดยวิธีต่างๆ กัน



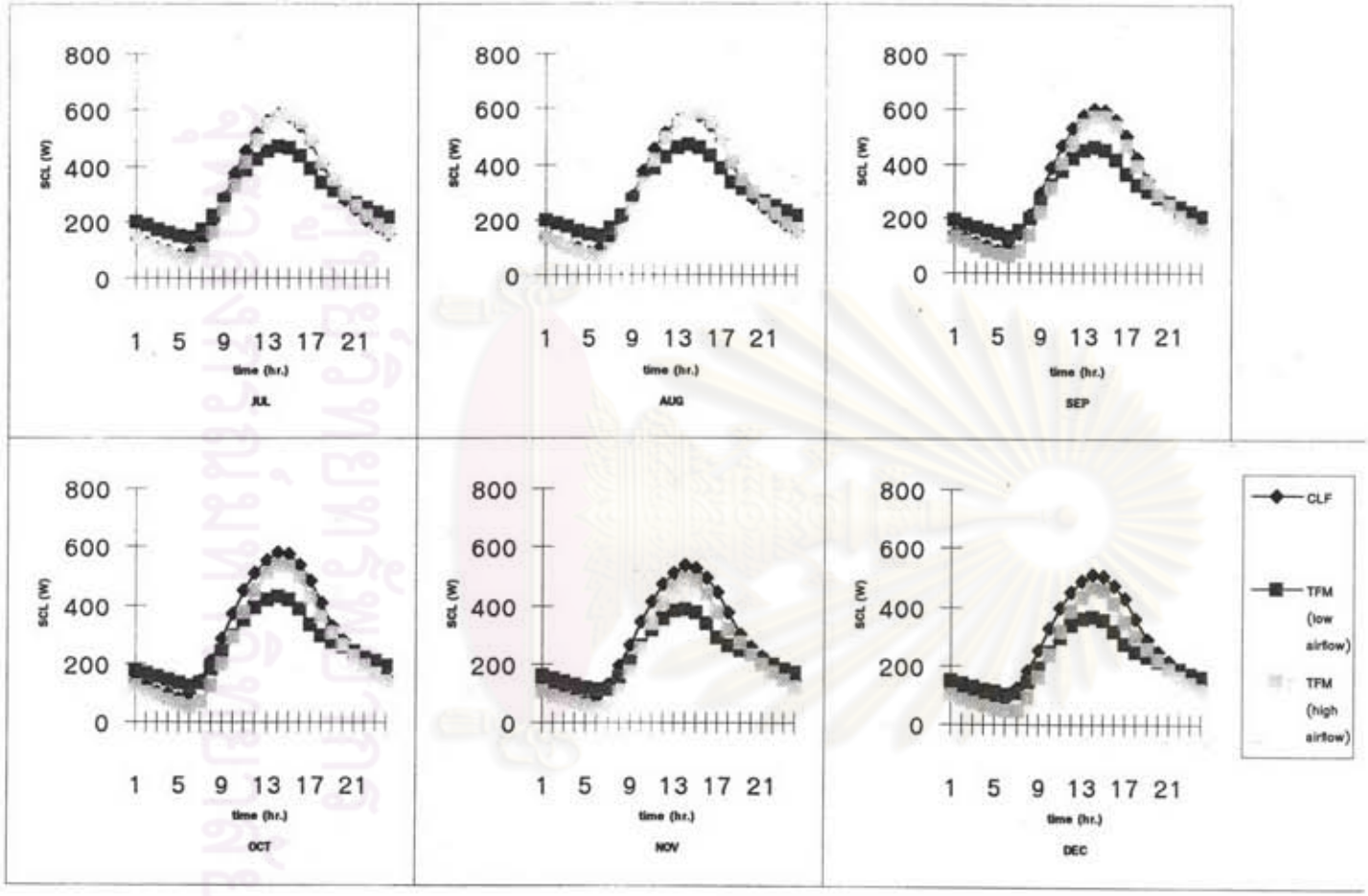
กราฟแสดงการเปรียบเทียบค่า north CLTD ของกำแพง 3 ชนิด ซึ่งคำนวณโดยวิธีต่างๆ กัน



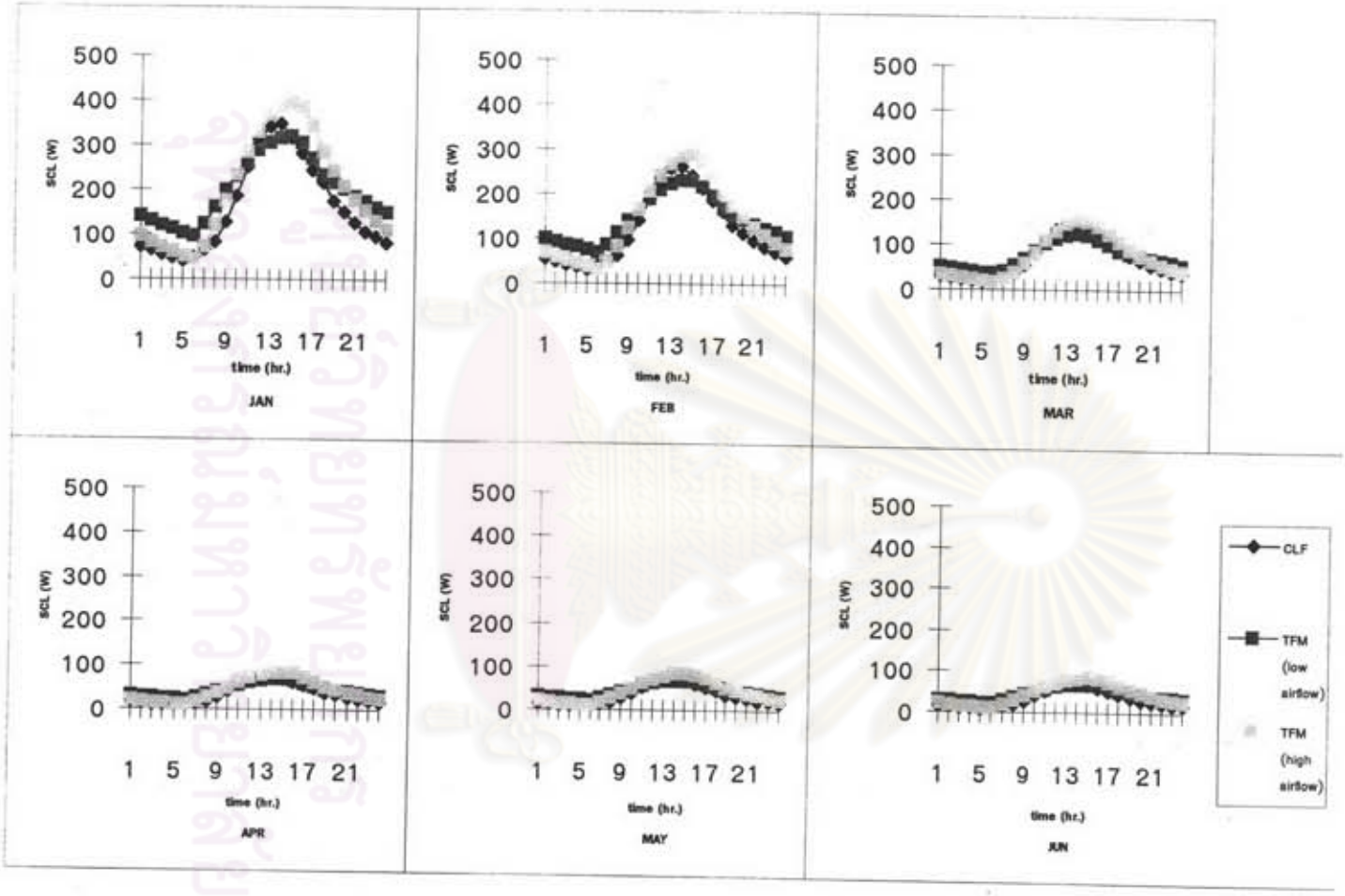
กราฟแสดงการเปรียบเทียบค่า west CLTD ของกำแพง 3 ชนิด ซึ่งคำนวณโดยวิธีต่างๆ กัน



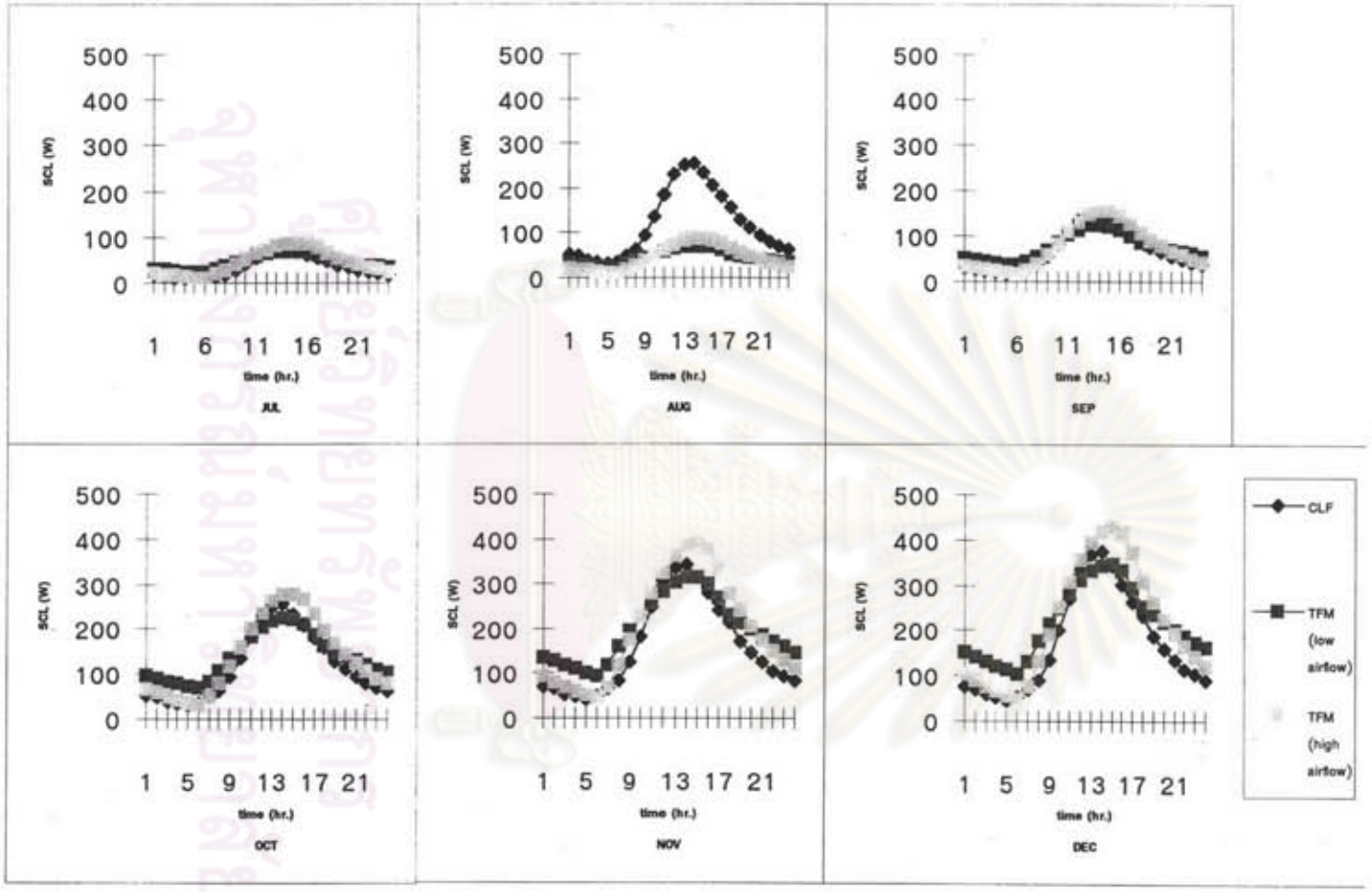
กราฟเปรียบเทียบค่า horrizontal SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 1 - 6



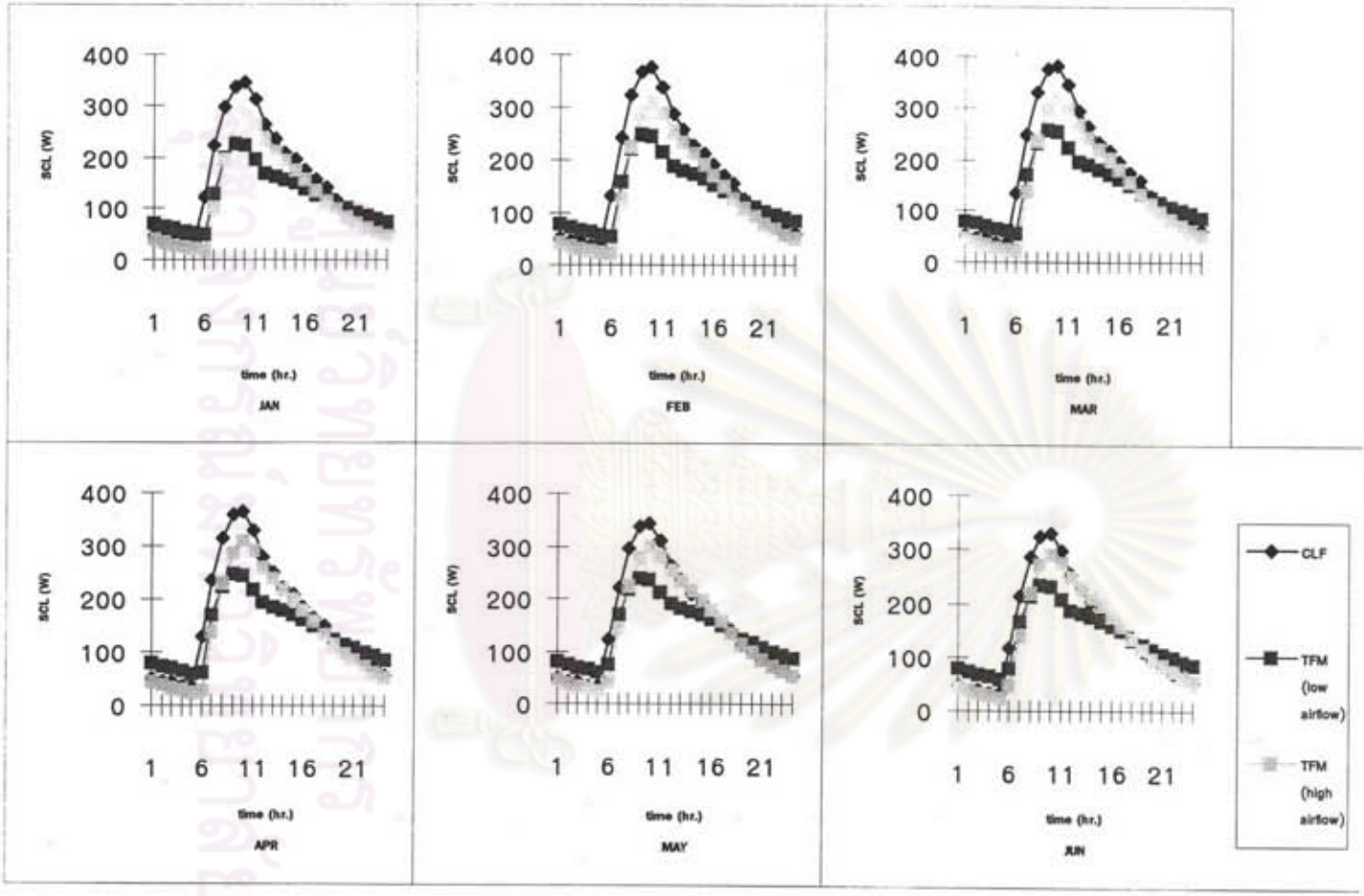
กราฟเปรียบเทียบค่า horrizontal SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 7 - 12



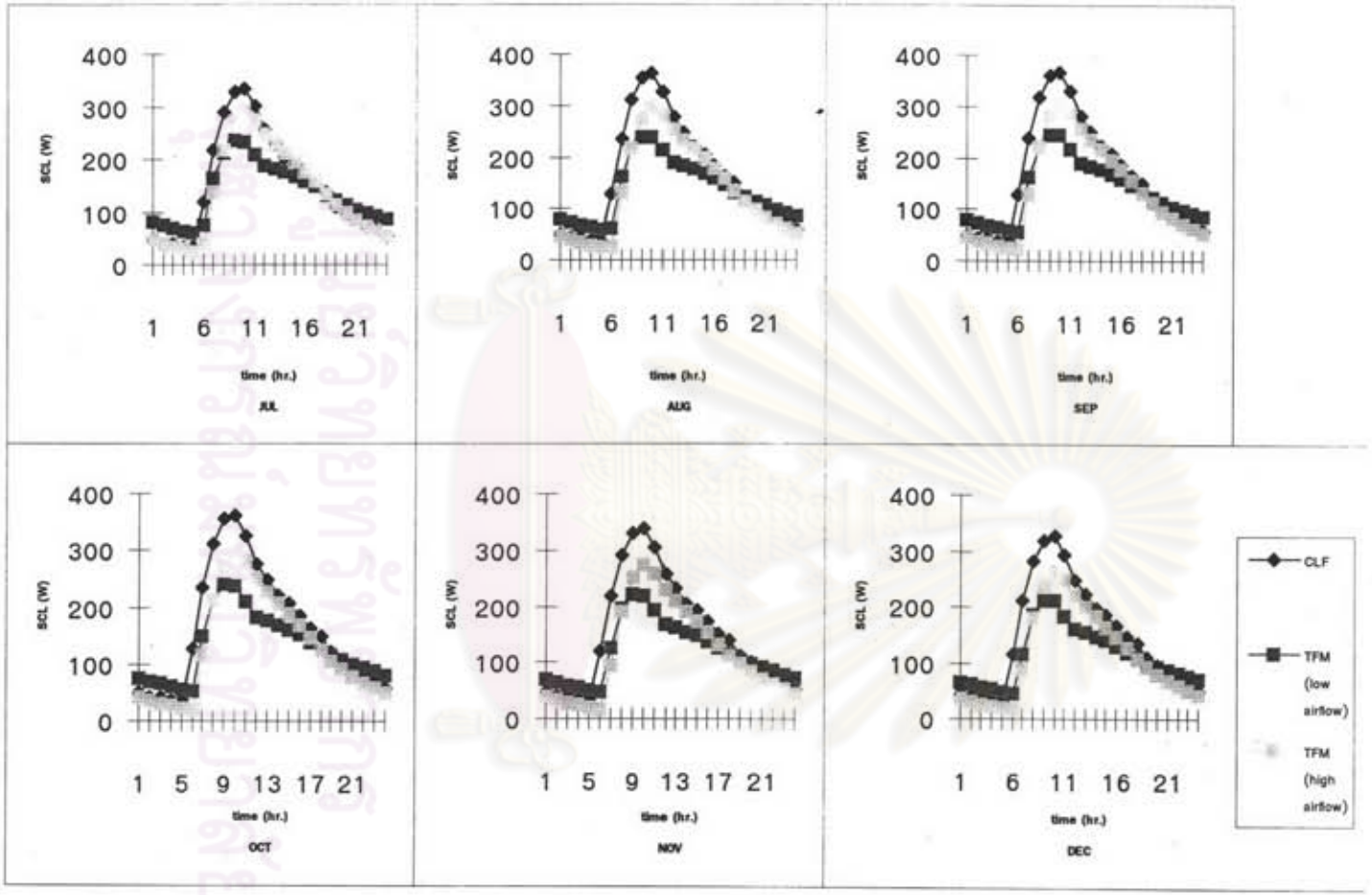
กราฟเปรียบเทียบค่า south SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 1 - 6



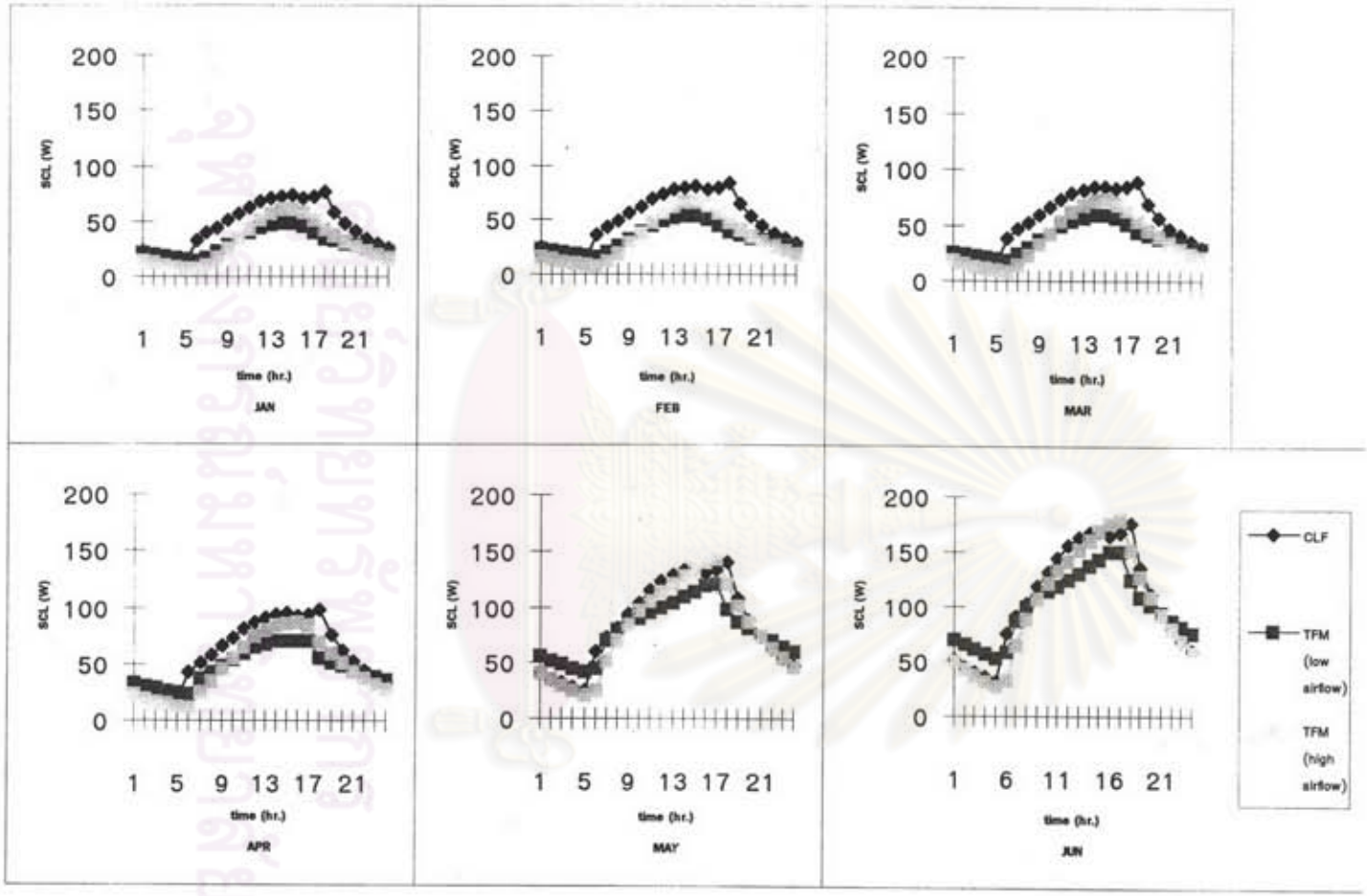
กราฟเปรียบเทียบค่า south SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 7 - 12



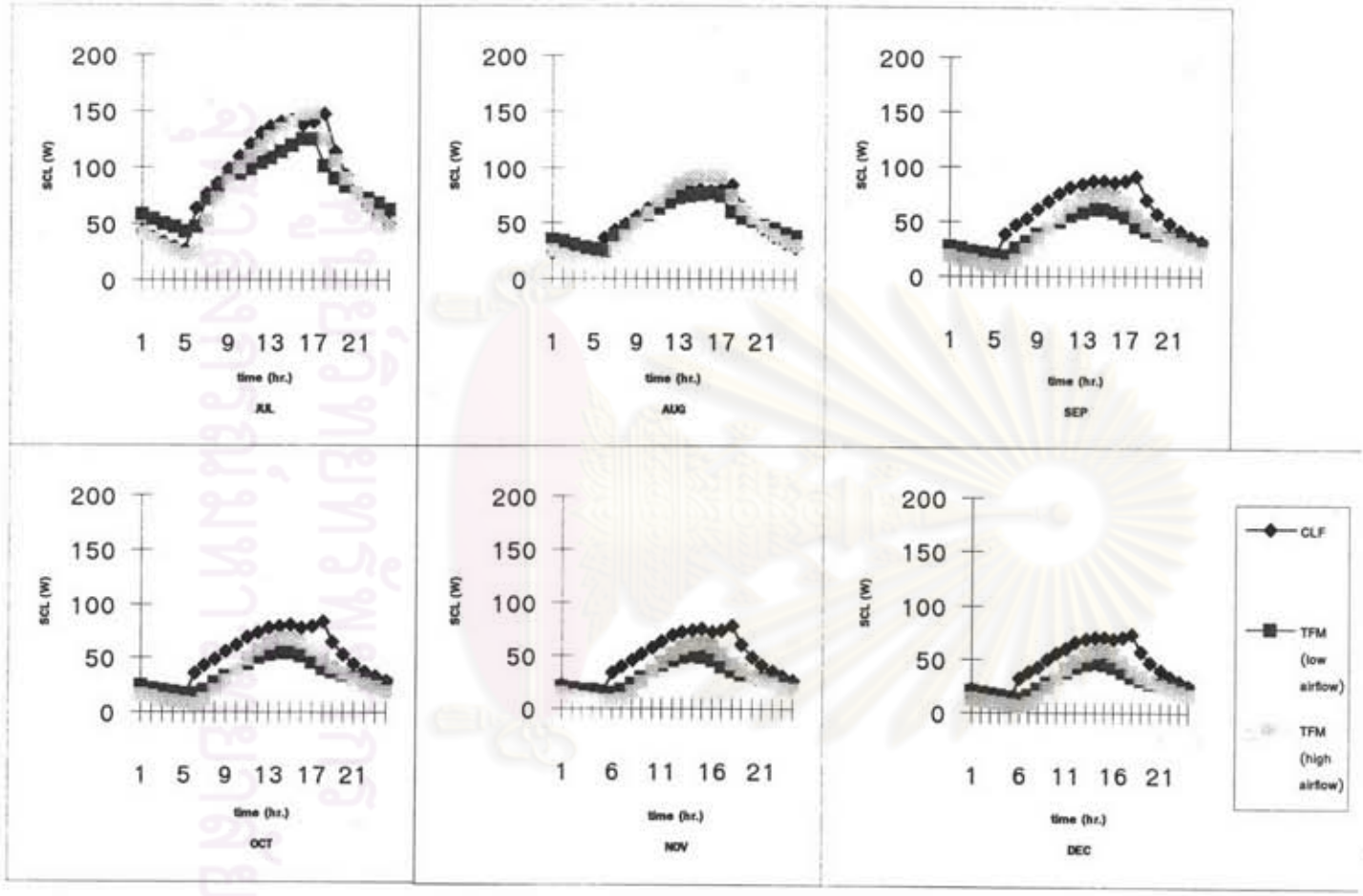
กราฟเปรียบเทียบค่า east SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 1 - 6



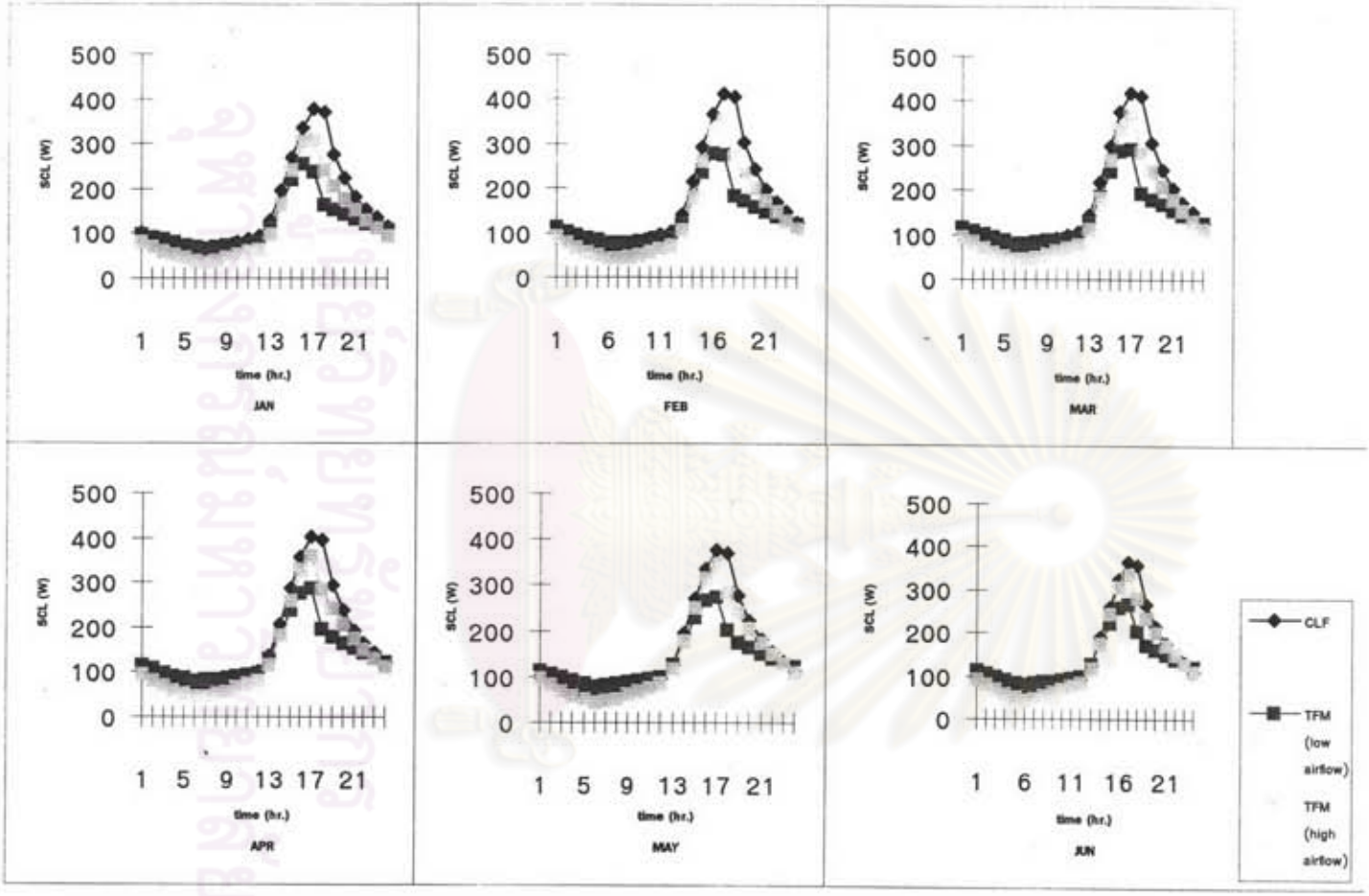
กราฟเปรียบเทียบค่า east SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 7 - 12



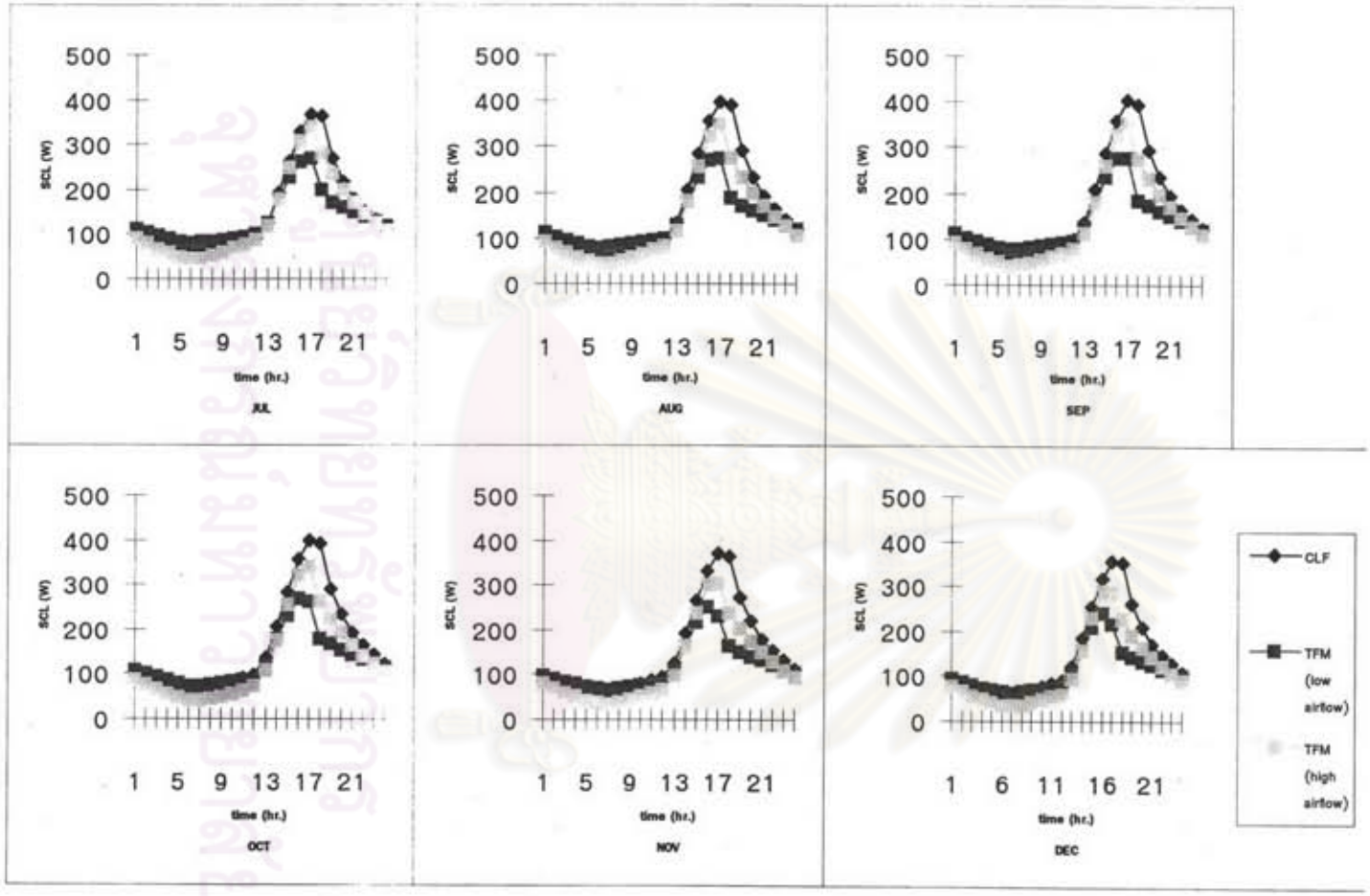
กราฟเปรียบเทียบค่า north SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 1 - 6



กราฟเปรียบเทียบค่า north SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 7 - 12



กราฟเปรียบเทียบค่า west SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 1 - 6



กราฟเปรียบเทียบค่า west SCL ที่คำนวณด้วยวิธีการต่างๆ กัน ตั้งแต่เดือน 7 - 12

ประวัติผู้เขียน

นายบุญชัย เลิศนุวัฒน์ เกิดที่ เขตยานาวา กรุงเทพมหานคร เมื่อวันที่ 15 ตุลาคม พ.ศ. 2514 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2536 และ เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537