COMPUTER AIDED ARCHITECTURAL EVALUATION AND DESIGN

- A COST MODELLING EXPERIMENT.

(Volume II)

by

Tami A. BELHADJ

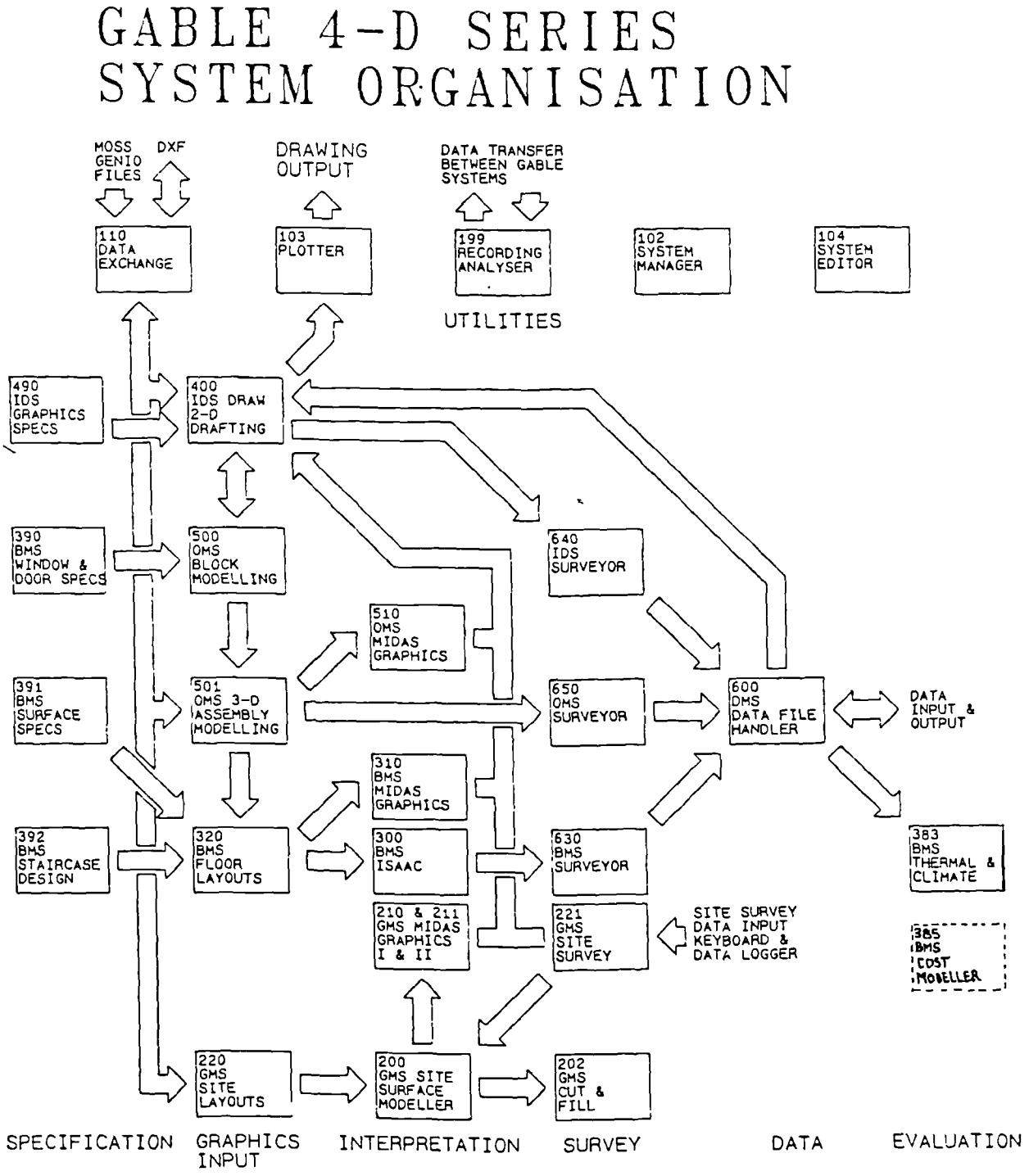
Thesis submitted to the University of Sheffield for the Degree of Doctor of Philosophy in the Faculty of Architectrural Studies.

Department of Architecture.

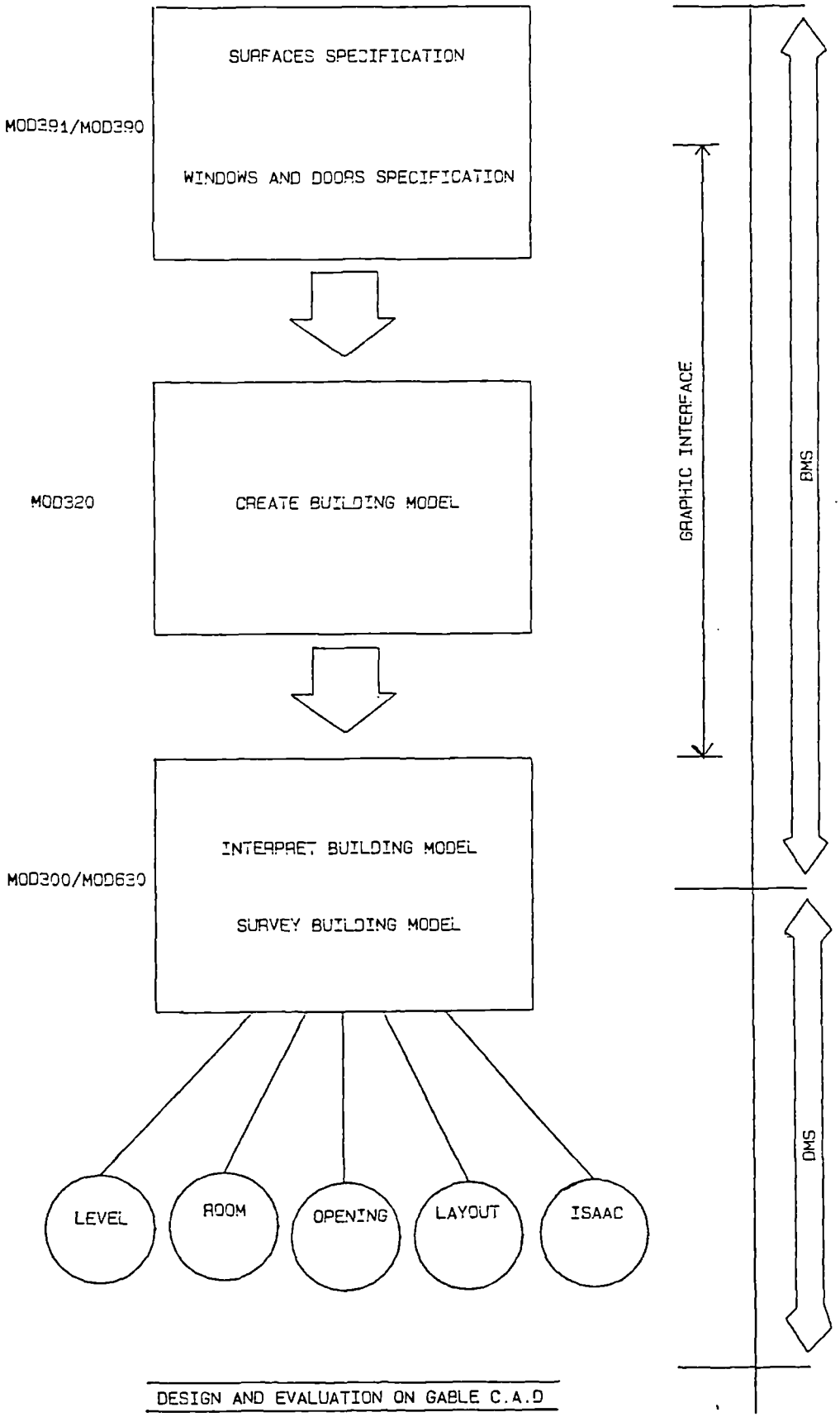
July 1989

APPENDICES

Appendix 10.1 : GABLE 4D-SERIES system orgenisation (Courtessy of GABLE CAD system Ltd.)



Appendix 10.2 : Building Modelling and Surveying.



Appendix 10.3.1 : Files structure specification.

MATERIALS :

File created by Module 391.

Contains a list of building materials for use in creating surfaces which in turn may be used in building models.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- TITLE	-	T49	
- SPEC	Nr	I	(1)
- K VALUE	W/M deg C	R	(2)
- DENSISTY	-	R	
- SPECIFIC HEAT	-	R	

(1) Keyed unique integer.

(2) Thermal conductivity

SURFACES :

File created by Module 391.

Containes a list of building surfaces for use in building models. Module 391 computes the values for U VALUE using field K VALUE in MATRIALs and ADMITTANCE using SPECIFICATION HEAT.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- TITLE	-	T49	
- SPEC	Nr	I	(1)
- THICKNESS	MM	R	(2)
- U VALUE	1/R	R	(3)
- ADMITTANCE	-	R	
- TYPE	Nr	I	(4)

(1) Keyed unique integer.

(2) Dimensioned thickness.

(3) The U-VALUE is the heat flow characteristic of a surface.

(4) This reflects the type of surface as defined in Module 391 and this has consequences for evaluation roitines.

Options are :

10	EXTERNAL WALL
20	INTERNAL WALL
30	ROOF WITH A ROOM BELOW
31	ROOF GLAZED WITH ROOM BELOW
32	ROOF WHOLLY EXTERNAL
40	CEILING WITH A ROOM ABOVE
41	CEILING WITH A ROOF ABOVE
43	CEILING WITHIN A ROOM
50	FLOOR WITH A ROOM BELOW
51	FLOOR WITH GROUND BELOW
52	FLOOR WITH OUTSIDE AIR BELOW
53	FLOOR WITHIN A ROOM
60	CONSTRUCTION

WINDOOR :

File created by Module 390.

Contains a list of windows and doors for use in building models.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- TITLE	-	T49	
- SPEC	Nr	I	(1)
- TYPE	-	T10	(2)
- WIDTH	MM	R	
- DEPTH	MM	R	
- HEIGHT	MM	R	
- SEGMENTS	Nr	I	(3)
- TOTAL AREA	M2	R	
- GLAZED AREA	M2	R	
- U VALUE	1/R	R	

(1) Unique keyed integre, this number is used together with the type in naming the BLOCK element.

(2) Options WINDOW or DOOR.

(3) Segments are 0 for windows and represent number of door leaves for doors.

Appendix 10.3.2 : Files structure specification.

LEVEL :

File created by Module 630.

May take any name but LEVEL is the default name.
Contains a list of floor levels in the building models.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- LEVEL	Nr	I	(1)
- DATUM	M	R	(2)
- CEILING HEIGHT	MM	R	(3)
- DOWNSTAND	MM	R	(4)
- NO OF ROOMS	Nr	I	
- GROSS AREA	M2	R	(5)
- EST VOLUME	M3	R	(6)

(1) 0 to 99 levels in the building.

(2) Default floor datum.

(3) Default ceiling height.

(4) Floor thickness or external wall downstand as stored in Module 320.

(5) Total GROSS AREA in the file ROOM for this level.

(6) Total EST VOLUME in file ROOM for this level.

ROOM:

Created by Module 630.

May take any name but ROOM is the default name.
Containes a list of all rooms in the building model.

Field name	Unit	Data Type	Description
- CODE	-	T10	
- ROOM NAME	-	T49	If any inserted.
- ROOM	Nr	I	Including level.
- LEVEL	Nr	I	
- ROOM TYPE	-	I	(1)
- GROSS AREA	M2	R	Area inside walls.
- FLOOR AREA	M2	R	(2)
- OTHER AREA	M2	R	(3)
- EST VOLUME	M3	R	(4)
- GROSS GIRTH	M	R	Total perimeter.
- NET GIRTH	M	R	(5)
- WINDOWS	Nr	I	Number of windows.
- DOORS	Nr	I	Number of doors.

(1) The room type will be default room type for that floor level unless a room marker was put in this particular room.

(2) Actual floor area (types 50, 51 and 52).

(3) Other floor area (type 53).

(4) Gross area and average height.

(5) Tota room perimeter less doors and floor level windows.

LAYOUT :

Created by Module 630.

May take any name but LAYOUT is the default name.

Containes a list of walls, floors planes, ceiling planes and roofs as drawn in Module 320. Enhencements made to the interpretation and survey modules during the course of this research have caused some alterations to the LAYOUT file record structure.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- ELEMENT	Nr	I	(1)
- SPEC	Nr	I	
- LEVEL	Nr	I	
- WALL NUMBER	Nr	I	(2)
- GROSS AREA	M2	R	
- GIRTH	M	R	(3)
- ROOM No 1	Nr	I	(4) New
- ROOM No 2	Nr	I	(4) New
- ORIENTATION	Deg	R	

(1) Options for elements as specified in Module 391.

(2) Only applies to walls, other types of surfaces have 0. Number indicates level and number for original wall.

(3) Walls gives lenght of centre line in plan, roofs, ceilings, floors gives perimetre length.

(4) These new fields indicate the room number on both sides of the wall. If the exterior is on one side, then room number is zero (0).

ISAAC:

Created by Module 630.

May take any name but LAYOUT is the default name.

Containes a list of surfaces as viewed from the inside of all rooms in the building.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- ELEMENT	Nr	I	(1)
- SPEC	Nr	I	
- LEVEL	Nr	I	
-WALL NUMBER	Nr	I	
- GROSS AREA	M2	R	
- NET AREA	M2	R	
- GIRTH	M	R	(2)
- ROOM	Nr	I	
- ORIENTATION	Deg	R	
- INCLINATION	Deg	R	
- INT EXT	-	T10	(3)

(1) Options for elements as specified in Module 391.

(2) Lenght of walls in plan, total primeter of floors or ceilings.

(3) Options are : EXTERNAL or INTERNAL.

OPENING :

Created by Module 630.

May take any name but OPENING is the default name.

Containes a list of windows and doors in building model.

Field name	Unit Description	Data Type	
- CODE	-	T10	
- ELEMENT	Nr	I	(1)
- SPEC	Nr	I	
- LEVEL	Nr	I	
-WALL NUMBER	Nr	I	
- GROSS AREA	M2	R	
- GIRTH	M	R	(2)
- ROOM NUMBER 1	Nr	I	(3)
- ROOM NUMBER 2	Nr	I	(3)
- ORIENTATION	Deg	R	
- INCLINATION	Deg	R	
- INT EXT	-	T10	(4)

(1) Options are : WINDOW or DOOR.

(2) Total perimeter of opening.

(3) It in external wall one room number will be zero (0).

(4) Options are : EXTERNAL or INTERNAL.

APPENDIX 10.4 : Space Fundamental Quantities (SFQ).

SFQ (Space Fundamental Quantities) file record structure.

Created by Module 385.

May take any name but SFQ is the default name.

Contains a list of space fundamental quantities.

Field type	Unit	Data Type	Description
- CODE	Nr	T10	
- LEVEL	Nr	I	
- GROSS AREA	M2	R	
- SPACE NAME	TEXT	R	
- SPACE TYPE	Nr	I	
- EXTERNAL WALL BEDS	M2	R	
- INTERNAL WALL BEDS	M2	R	
- EXTERNAL STOREY HEIGHT	M	R	
- INTERNAL STOREY HEIGHT	M	R	
- EXTERNAL WALL AREA	M2	R	
- INTERNAL WALL AREA	M2	R	
- EXTERNAL WINDOW AREA	M2	R	
- INTERNAL WINDOW AREA	M2	R	
- EXTERNAL DOOR AREA	M2	R	
- INTERNAL DOOR AREA	M2	R	
- CEILING AREA	M2	R	
- SUSPENDED CEILING AREA	M2	R	
- GROUND SLAB AREA	M2	R	
- ROOF AREA	M2	R	

APPENDIX 10.5 : Floor Fundamental Quntities (FFQ).

FFQ (Floor Fundamental Quntities) file record structure.

Created by Module 385.

May take any name but FFQ is the default name.

Contains a list of floor fundamental quantities.

Field type	Uni	Data Type	Description
- CODE	Nr	T10	
- LEVEL	Nr	I	
- GROSS AREA	M2	R	(1)
- GROUND SLAB AREA	M2	R	(2)
- LETABLE USABLE AREA	M2	R	
- EXTERNAL STOREY HEIGHT	M	R	(3)
- INTERNAL STOREY HEIGHT	M	R	(4)
- EXTERNAL WALL AREA	M2	R	
- INTERNAL WALL AREA	M2	R	
- EXTERNAL WINDOW AREA	M2	R	
- INTERNAL WINDOW AREA	M2	R	
- EXTERNAL DOOR AREA	M2	R	
- INTERNAL DOOR AREA	M2	R	
- ROOF AREA	M2	R	(5)
- CEILING AREA	M2	R	
- STAIRCASE FLIGHTS	Nr	I	
- LIFT FLIGHTS	Nr	I	
- SUSPENDED CEILING AREA	M2	R	
- SANITARY FITTINGS	Nr	I	
- NUMBER OF SPACES	Nr	I	

(1) It is the total area inside walls of all spaces, plus the INTERNAL WALL BEDS.

(2) It is the GROSS AREA plus the EXTERNAL WALL BEDS.

(3) The INTERNAL STOREY HEIGHT of a floor level is calculated by dividing the total NET VOLUME by the total GROSS AREA of all spaces on that particular floor level.

(4) The EXTERNAL STOREY HEIGHT of a floor level is calculated by adding to the INTERNAL STOREY HEIGHT, the floor thickness of the floor above. When it comes to the top floor level of the building the user is prompted with a message asking for the storey height dimension, and suggests the EXTERNAL STOREY HEIGHT of the floor bellow.

(5) The ROOF AREA is the un-folded roof area.

APPENDIX 10.6 : Building Elemental Quantities (BEM).

BEM (Building Elemental Quntities) file record structure.

Created by Module 385.

May take any name but BEM is the default name.

Contains a list of building elemental quantities of the building model in a Ci/Sfb fromal..

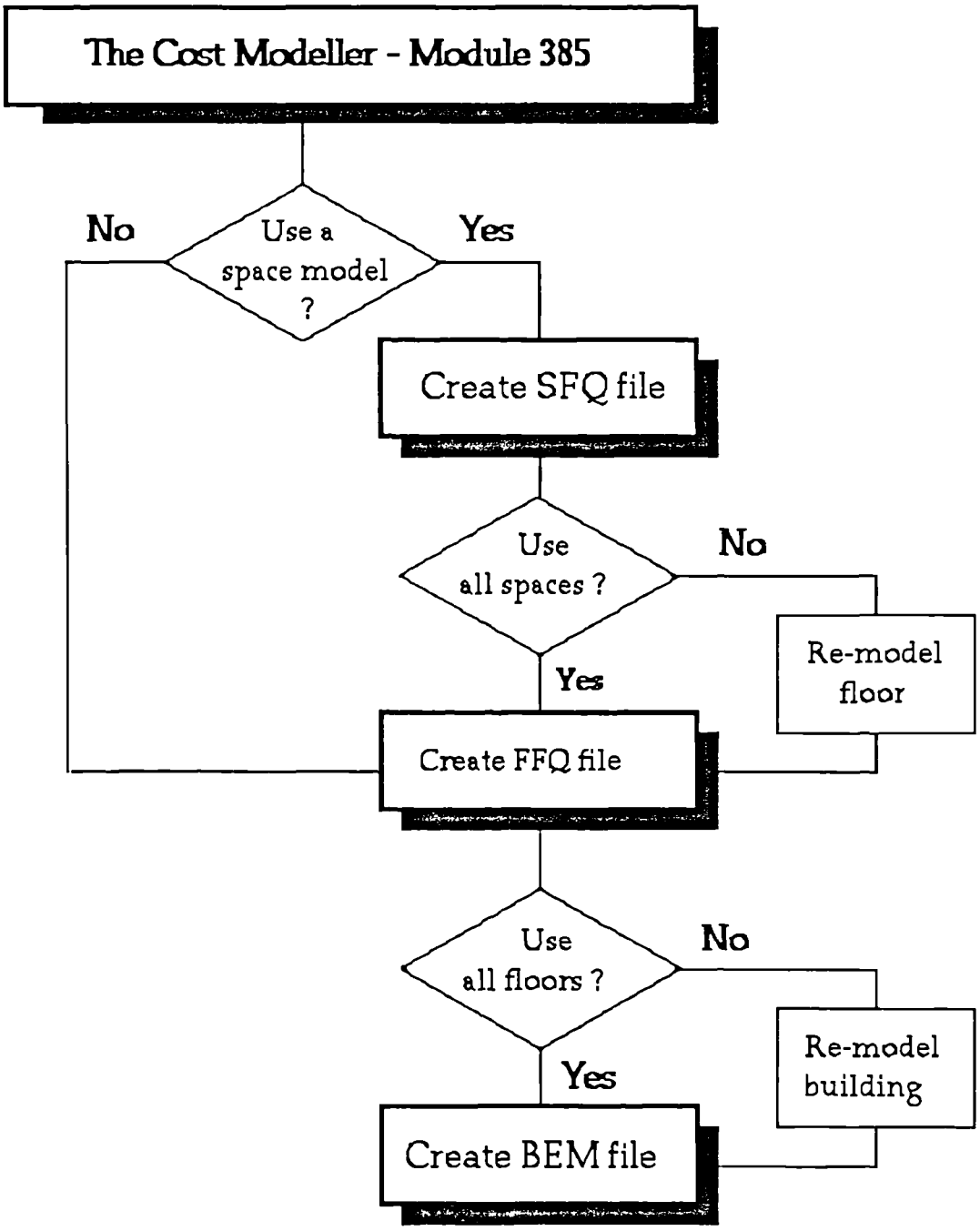
Field type	Unit	Data Type	Description
- CODE	Nr	T10	(1)
- ELEMENT	-	T49	(2)
- UNIT	-	T10	
- ELEMENTAL COST	-	R	
- PROJECT RATE	-	R	
- PRICE ADJUSTEMEMT	-	R	(3)
- NASE RATE	-	R	

(1) Currently containing Ci/Sfb standard code.

(2) The string UNIT is set to blank in case of elemental quantities heading.

(3) The price adjustement is set to 1 (one) by default.

APPENDIX 10.7 : THE COST MODELLER.



- A general algorithm representing the cost modeller main decisions procedure. -

APPENDIX 10.8.1 : PROGRAMS LISTING.

```
      ********************************
      SUBROUTINE ADDBEM(FNAM,IDUM,OK)
      ********************************
C     *
C     store BEM dms file
C     *
%INCLUDE ':GABLE:V31:MOD385:BEM.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*10 STR3,STR4
      CHARACTER*50 FNAM,STR,STR1,STR2,CHR,FNDUM
      CHARACTER*200 LONG
      REAL*8 NILL
      INTEGER I,ITILD,TILD,IDUM
      LOGICAL OK
C     *
      OK=.TRUE.
      NILL=0D0
C     *
      CALL FILBLD(JOB,'DMS_',FNAM,FNDUM)
      CALL DMSTRA(FNAM,OK)
      IF (.NOT.OK) THEN
        CALL ERRMES('failed to trace DMS file_')
        RETURN
      END IF
C     *
      DO 800 I=1,100
      LONG(1:40)='                                        '
      LONG(41:80)='                                        '
      LONG(81:120)='                          1             '
      LONG(121:160)='                                        '
      LONG(161:200)='                                        '
      STR1=ELMBEQ(I)
      STR2=UNTBEQ(I)
      STR3=COD(I)
      STR4=STA(I)              ! status
      JTILD=INDEX(STR1,'_')-1
      IF(JTILD.GT.0) THEN
        LONG(12:12+JTILD-1)=STR1(1:JTILD)       ! element
      END IF
      JTILD=INDEX(STR2,'_')-1
      LONG(63:63+JTILD-1)=STR2(1:JTILD)         ! unit
      JTILD=INDEX(STR3,'_')-1
      LONG(1:JTILD-1)=STR3(1:JTILD)             ! code
      CALL DROUND(BEQ(I),STR,2)
      JTILD=INDEX(STR,'_')-1
      LONG(74:74+JTILD-2)=STR(2:JTILD)          ! beq
      CALL DROUND(NILL,STR,2)
      JTILD=INDEX(STR,'_')-1
      LONG(85:85+JTILD-2)=STR(2:JTILD)          ! elemental cost
      CALL DROUND(NILL,STR,2)
      JTILD=INDEX(STR,'_')-1
      LONG(96:96+JTILD-2)=STR(2:JTILD)          ! project rate
```

```
C       CALL DROUND(NILL,STR,2)
C       JTILD=INDEX(STR,'_')-1
C       LONG(107:107+JTILD-2)=STR(2:JTILD)          ! price adj
        CALL DROUND(NILL,STR,2)
        JTILD=INDEX(STR,'_')-1
        LONG(118:118+JTILD-2)=STR(2:JTILD)          ! base rate
        CALL IROUND(IDUM,STR,J)
        LONG(129:129+J-1)=STR(2:J)                  ! level
        CALL DROUND(NILL,STR,2)
        JTILD=INDEX(STR,'_')-1
        LONG(139:139+JTILD-2)=STR(2:JTILD)          ! percentage
        JTILD=INDEX(STR4,'_')-1
        LONG(149:149+JTILD-1)=STR4(1:JTILD)
        NREC=NREC+1
        WRITE(IDCHAN,REC=NREC) LONG(1:IFLDW)
800     CONTINUE
C       *-------------------------------------------------------
        RETURN
        END
```

```
          ****************************
          SUBROUTINE ADDFFQ(FNAM,DONE)
          ****************************
C         *
C         stores actual data in file FFQ
C         *
C       01-OCT-87 REV_1.15 HND I=ILOOP to avoid compile errors
C         *
          INTEGER J,JTILD,I,IDUM,ILOOP
          CHARACTER*50 FNAM,STR,FNDUM
          CHARACTER*240 LONG
          LOGICAL DONE,OK
C         *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C         *
C         CALL FILBLD(JOB,'DMS_',FNAM,FNDUM)
          CALL DMSTRA(FNAM,OK)
          IF(.NOT.OK) RETURN
C         *
          DO 100 ILOOP=1,100
            I=ILOOP
            IF(LEVELN(I).EQ.I) THEN      !defined level
              LONG(1:40)='                                        '
              LONG(41:80)='                                        '
              LONG(81:120)='                                        '
              LONG(121:160)='                                        '
              LONG(161:200)='                                        '
              LONG(201:240)='                                        '
              IDUM=I-1                   ! get the level number right
              CALL IROUND(IDUM,STR,J)
              LONG(12:12+J-2)=STR(2:J)      !level number
              LONG(1:1+J-2)=STR(2:J)        !code
              CALL DROUND(GRAREA(I),STR,2)
              JTILD=INDEX(STR,'_')-1
              LONG(23:23+JTILD-1)=STR(2:JTILD)  !gross floor area
              CALL DROUND(GSAREA(I),STR,2)
              JTILD=INDEX(STR,'_')-1
              LONG(34:34+JTILD-1)=STR(2:JTILD)  !ground slab area
              CALL DROUND(LUAREA(I),STR,2)
              JTILD=INDEX(STR,'_')-1
              LONG(45:45+JTILD-1)=STR(2:JTILD)  !let/useable area
              CALL DROUND(EXSTHT(I),STR,0)
              JTILD=INDEX(STR,'_')-1
              LONG(56:56+JTILD-1)=STR(2:JTILD)  !external storey ht
              CALL DROUND(INSTHT(I),STR,0)
              JTILD=INDEX(STR,'_')-1
              LONG(67:67+JTILD-1)=STR(2:JTILD)  !internal storey ht
              CALL DROUND(EXWAAR(I),STR,2)
              JTILD=INDEX(STR,'_')-1
              LONG(78:78+JTILD-1)=STR(2:JTILD)  !external wall area
              CALL DROUND(INWAAR(I),STR,2)
              JTILD=INDEX(STR,'_')-1
```

```
            LONG(89:89+JTILD-1)=STR(2:JTILD)  !internal wall area
            CALL DROUND(EXWIAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(100:100+JTILD-1)=STR(2:JTILD)  !external wind area
            CALL DROUND(INWIAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(111:111+JTILD-1)=STR(2:JTILD)  !internal wind area
            CALL DROUND(EXDOAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(122:122+JTILD-1)=STR(2:JTILD)  !external door area
            CALL DROUND(INDOAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(133:133+JTILD-1)=STR(2:JTILD)  !internal door area
            CALL DROUND(ROAREA(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(144:144+JTILD-1)=STR(2:JTILD)  !roof area
            CALL DROUND(CEAREA(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(155:155+JTILD-1)=STR(2:JTILD)  !ceiling area
C           CALL IROUND(PILEHD(I),STR,J)
C           LONG(166:166+J-1)=STR(2:J)          !pile heads
            CALL IROUND(STRFLT(I),STR,J)
            LONG(166:166+J-1)=STR(2:J)          !staircase flights
            CALL IROUND(LFTFLT(I),STR,J)
            LONG(177:177+J-1)=STR(2:J)          !lift flights
            CALL DROUND(SUSCLG(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(188:188+JTILD-1)=STR(2:JTILD)  !suspended ceiling
            CALL IROUND(SANFIT(I),STR,J)
            LONG(199:199+J-1)=STR(2:J)          !sanitary fittings
            CALL IROUND(NUMSPA(I),STR,J)
            LONG(210:210+J-1)=STR(2:J)          !number of spaces
C           *
            NREC=NREC+1
            WRITE(IDCHAN,REC=NREC) LONG(1:IFLDW)
          END IF
100     CONTINUE
        CALL DMSSTR
        DONE=.TRUE.
        RETURN
        END
```

```
      ************************************
      SUBROUTINE ADDSFQ(FNAM,LNUM,NSPA,DONE)
      ************************************
      IMPLICIT NONE
C     *
C     stores actual data in file SFQ
C     *
C   17-FEB-88 REV_1.16 TAB
C     *
      INTEGER J,JTILD,I,IDUM,NSPA,LNUM
      CHARACTER*50 FNAM,STR,FNDUM,STR1
      CHARACTER*260 LONG
      LOGICAL DONE,OK
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
C     CALL FILBLD(JOB,'DMS_',FNAM,FNDUM)
      CALL DMSTRA(FNAM,OK)
      IF (.NOT.OK) RETURN
      DO 100 I=1,100
        IF(I.LE.NSPA) THEN           !defined space
          LONG(1:40)='                                        '
          LONG(41:80)='                                        '
          LONG(81:120)='                                        '
          LONG(121:160)='                                        '
          LONG(161:200)='                                        '
          LONG(201:240)='                                        '
          LONG(241:260)='                    '
          IDUM=I
          CALL IROUND(NREC,STR,J)
          LONG(1:1+J-2)=STR(2:J)                !code
          CALL IROUND(LNUM,STR,J)
          LONG(12:12+J-2)=STR(2:J)              !level number
          CALL DROUND(GAREAS(I),STR,2)
          JTILD=INDEX(STR,'_')-1
          LONG(23:23+JTILD-1)=STR(2:JTILD)      !gross floor area
          STR1=SPATIT(I)
          JTILD=INDEX(STR1,'_')-1
          LONG(34:34+JTILD-1)=STR1(1:JTILD)      !space title
          CALL IROUND(ROONUM(I),STR,J)
          LONG(85:85+J-2)=STR(2:J)               !space number
          CALL IROUND(SPATYP(I),STR,J)
          LONG(96:96+J-2)=STR(2:J)               !space type
          CALL DROUND(EXWABD(I),STR,2)
          JTILD=INDEX(STR,'_')-1
          LONG(107:107+JTILD-1)=STR(2:JTILD)      !ext wall beds
          CALL DROUND(INWABD(I),STR,2)
          JTILD=INDEX(STR,'_')-1
          LONG(118:118+JTILD-1)=STR(2:JTILD)    !int wall beds
          CALL DROUND(ESTHTS(I),STR,2)
          JTILD=INDEX(STR,'_')-1
          LONG(129:129+JTILD-1)=STR(2:JTILD)    !ext storey height
```

```
            CALL DROUND(ISTHTS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(140:140+JTILD-1)=STR(2:JTILD)   !int storey height
            CALL DROUND(EWAARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(151:151+JTILD-1)=STR(2:JTILD)   !ext wall area
            CALL DROUND(IWAARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(162:162+JTILD-1)=STR(2:JTILD)   !int wall area
            CALL DROUND(EWIARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(173:173+JTILD-1)=STR(2:JTILD)   !external wind area
            CALL DROUND(IWIARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(184:184+JTILD-1)=STR(2:JTILD)   !internal wind area
            CALL DROUND(EDOARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(195:195+JTILD-1)=STR(2:JTILD)  !ext door area
            CALL DROUND(IDOARS(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(206:206+JTILD-1)=STR(2:JTILD)  !int door area
            CALL DROUND(CEILAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(217:217+JTILD-1)=STR(2:JTILD)  !ceiling area
            CALL DROUND(SUCEAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(228:228+JTILD-1)=STR(2:JTILD)  !suspended ceiling
            CALL DROUND(GSLABA(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(239:239+JTILD-1)=STR(2:JTILD)  !ground slab area
            CALL DROUND(ROOFAR(I),STR,2)
            JTILD=INDEX(STR,'_')-1
            LONG(250:250+JTILD-1)=STR(2:JTILD)  !roof area
C           *
            NREC=NREC+1
            WRITE(IDCHAN,REC=NREC) LONG(1:IFLDW)
          END IF
100     CONTINUE
        CALL DMSSTR
        DONE=.TRUE.
        RETURN
        END
```

```
      ******************************
      SUBROUTINE ADJRAT(SEL,ELEM,OK)
      ******************************
C     *
C     rates adjustement routine
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
C     *
C     29-MAR-88 REV_1.16 TAB  rate=r1*r2
C     *
      CHARACTER*50 CHR,STR
      INTEGER SEL,ELEM,INREC,IRESP,IFREC
      REAL*8 R1,R2,RATE
      LOGICAL OK
C     *
      OK=.TRUE.
C
      IF (ELEM.LT.SEL*10.OR.ELEM.GT.SEL*10+10) THEN
        CALL ERRMES('response not in range')
        OK=.FALSE.
        RETURN
      END IF
      CALL IROUND(ELEM,STR,J)
      CHR='('//STR(2:)
      JTILD=INDEX(CHR,'~')
      CHR=CHR(1:JTILD)
      CALL FINDRC(1,CHR,0,IFREC)
      IF (IFREC.EQ.0) THEN
        CALL ERRMES('not found')
        INREC=0
        OK=.FALSE.
        RETURN
      ELSE
        DO 2000 I=1,IF9
          IF (LOCFLD(I).NE.2) LOCFLD(I)=1
2000    CONTINUE
        LOCFLD(7)=0
C       *
        CALL EDITRC(IFREC,IRESP)
C       IF (IRESP.LT.1) THEN
C         OK=.FALSE.
C         RETURN
C       ELSE
C         CALL GETFLD(IFREC,7,CHR,K,N,R1)       !PRICE ADJUST.
C         CALL GETFLD(IFREC,8,CHR,K,N,R2)       !BASE RATE
C         RATE=R1*R2
C         CALL PUTFLD(IFREC,6,CHR,3,N,RATE)     !PROJECT RATE
C       END IF
      END IF
C
      RETURN
      END
```

```
      **********************
      SUBROUTINE BEMMNR(OK)
      **********************
C     *
C     Building Elemental Quantities main routine
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
C     *
C     *
      CHARACTER*50 FNAM,NAME,MENUT(20)
      LOGICAL OK,EXIST,DONE,ON,OFF
      INTEGER I,IDUM
C     *
      IDUM=-1
      OK=.TRUE.
C     *
C     selecte FFQ or SFQ
C     *
150   CONTINUE
      MENUT(1)='BUILDING ELEMENTAL QUANTITIES '
      MENUT(2)='Using Floor Fundamental Quantities file '
      MENUT(3)='Using a selected number of Floors '
      MENUT(4)='Display Building Elemental Model file '
C     *
      CALL MENU(MENUT,1,3,I)
      GOTO (1000,2000,3000) I
      RETURN
C     *
 1000 CONTINUE
C     *
C     get FFQ file name
C     *
      NAME='FFQ '
      CALL GETIFL(NAME,FNAM,OK)
      IF (.NOT.OK) RETURN
C     *
      CALL USEFFQ(FNAM,OK)
      IF (.NOT.OK) THEN
       CALL ERRMES('can not use FFQ file ')
       RETURN
      END IF
C     *
C     using the all of FFQ file content
C     *
      OFF=.FALSE.
      CALL CALTOT(OFF,IDUM,OK)
      IF (.NOT.OK) RETURN
      CALL BEMPT1
      NAME='BEM '
      CALL GETOFL(NAME,FNAM,OK)
      IF (.NOT.OK) RETURN
      CALL CREBEM(FNAM)
      IF (.NOT.OK) RETURN
```

```
          CALL ADDBEM(FNAM,IDUM,OK)
          IF (.NOT.OK) THEN
            CALL ERRMES('failed adding BEM file ')
            GOTO 150
          END IF
          CALL DMSSTR
          CALL CHAROT('file'//FNAM//'stored')
          CALL DMSCLO
          GOTO 150
C         *---------------------------------------
2000      CONTINUE
C         *
C         get FFQ file name
C         *
          NAME='FFQ '
          CALL GETIFL(NAME,FNAM,OK)
          IF (.NOT.OK) RETURN
C         *
          CALL USEFFQ(FNAM,OK)
          IF (.NOT.OK) THEN
           CALL ERRMES('can not use FFQ file ')
           RETURN
          END IF
C         *
C         using a selected number of floors of FFQ file
          OFF=.TRUE.
          CALL CALTOT(OFF,IDUM,OK)
          IF (.NOT.OK) RETURN
          CALL BEMPT1
          NAME='BEM '
          CALL GETOFL(NAME,FNAM,OK)
          IF (.NOT.OK) RETURN
          CALL CREBEM(FNAM)
          IF (.NOT.OK) RETURN
          CALL ADDBEM(FNAM,IDUM,OK)
          IF (.NOT.OK) THEN
            CALL ERRMES('failed adding BEM file ')
            GOTO 150
          END IF
          CALL DMSSTR
          CALL CHAROT('file'//FNAM//'stored')
          CALL DMSCLO
          GOTO 150
C         *
 3000     CONTINUE
C         *
C         display BEM file
C         *
          ON=.FALSE.
          CALL DISBEM(ON)
          CALL DMSCLO
          GOTO 150
C         *
          END
```

```
      ********************
      SUBROUTINE BEMPT1
      ********************
C     *
C     work out Building Elemental Quantities from FFQ.COM
C     *
C    29-MAR-88 REV_1.16 TAB see  floor openings
C     *
%INCLUDE ':GABLE:V31:MOD385:TOTFFQ.COM'
%INCLUDE ':GABLE:V31:MOD385:BEM.COM'
C     *
      INTEGER I,JTILD,KTILD,TILD
      CHARACTER*10 STR
      CHARACTER*50 CHR
      LOGICAL RIGHT
C     *
      DO 250 I=1,100
        BEQ(I)=0D0
        ELMBEQ(I)='_'
        UNTBEQ(I)='_'
        COD(I)='_'
        STA(I)='_'
250   CONTINUE
      DO 150 I=1,100
      UNTBEQ(I)='m2_'
      IF (I.EQ.8.OR.I.EQ.15.OR.I.EQ.25.OR.I.EQ.35.OR.I.EQ.57
     +  .OR.I.EQ.65) THEN
        UNTBEQ(I)='Nr_'
      END IF
      IF (INDEX(ELMBEQ(I),'_').LT.2) THEN
        ELMBEQ(I)='NOT USED_'
      END IF
150   CONTINUE
C     *-----------------------------------------
C     get SUBSTRUCTURE elemental quantities
C     *-----------------------------------------
      ELMBEQ(1)='SUBSTRUCTURE_____________'
      UNTBEQ(1)='_'
      ELMBEQ(2)='ground_'
      BEQ(2)=FFQTOT(2)
      ELMBEQ(4)='floor beds_'
      BEQ(4)=FFQTOT(2)
      ELMBEQ(7)='retaining walls, founds_'
      BEQ(7)=FFQTOT(2)
      ELMBEQ(8)='pile foundations_'
      BEQ(8)=FFQTOT(17)
C     *-----------------------------------------
C     get STRUCTURE elemental quantities
C     *-----------------------------------------
      ELMBEQ(11)='STRUCTURE_____________'
      UNTBEQ(11)='_'
      ELMBEQ(12)='external walls area_'
      BEQ(12)=FFQTOT(6)-(FFQTOT(8)+FFQTOT(10))
      ELMBEQ(13)='internal walls area_'
```

```
      BEQ(13)=(FFQTOT(7)/2)-(FFQTOT(9)+FFQTOT(11))
      ELMBEQ(14)='floors '
      BEQ(14)=FFQTOT(14)
      ELMBEQ(15)='stairs '
      BEQ(15)=FFQTOT(16)
      ELMBEQ(18)='roofs area '
      BEQ(18)=FFQTOT(12)
      ELMBEQ(19)='frames '
      BEQ(19)=FFQTOT(1)
C     *-----------------------------------------
C     get COMPLETION elemental quantities
C     *-----------------------------------------
      ELMBEQ(21)='COMPLETION_____________ '
      UNTBEQ(21)=' '
      ELMBEQ(22)='openings in external walls '
      BEQ(22)=FFQTOT(8)+FFQTOT(10)
      ELMBEQ(23)='openings in internal walls '
      BEQ(23)=FFQTOT(9)+FFQTOT(11)
      ELMBEQ(24)='floor openings '              ! ?????????
      BEQ(24)=FFQTOT(1)-FFQTOT(13)              ! just testing
      ELMBEQ(25)='balustrading '
      BEQ(25)=FFQTOT(15)
      ELMBEQ(26)='suspended ceilings area '
      BEQ(26)=FFQTOT(18)
      ELMBEQ(28)='rooflights '
C     BEQ(28)=                                  ! WHAT'S THIS ?
C     *-----------------------------------------
C     get FINISHES elemental quantities
C     *-----------------------------------------
      ELMBEQ(31)='FINISHES_______________ '
      UNTBEQ(31)=' '
      ELMBEQ(32)='external walls area '
      BEQ(32)=BEQ(12)
      ELMBEQ(33)='internal walls area '
      BEQ(33)=BEQ(12)+(BEQ(13)*2)
      ELMBEQ(34)='floor finishes '
      BEQ(34)=FFQTOT(3)
      ELMBEQ(35)='stair finishes '
      BEQ(35)=FFQTOT(16)
      ELMBEQ(36)='ceiling finishes '
      BEQ(36)=FFQTOT(13)
      ELMBEQ(38)='roof finishes '
      BEQ(38)=FFQTOT(12)
C     *-------------------------------------------------
C     get SERVICES building fundamental quantities
C     *-------------------------------------------------
      ELMBEQ(41)='SERVICES_______________ '
      UNTBEQ(41)=' '
      ELMBEQ(42)='refuse '
      ELMBEQ(43)='waste disposal '
      ELMBEQ(44)='hot and cold water '
      ELMBEQ(45)='gases supply '
      ELMBEQ(46)='space cooling '
      ELMBEQ(47)='space heating '
```

```
      ELMBEQ(48)='air conditioning_'
      DO 40 I=42,50
      BEQ(I)=FFQTOT(1)
40    CONTINUE
C     *-------------------------------------------------
C     get INSTALLATIONS fundamental quantities
C     *-------------------------------------------------
      ELMBEQ(51)='INSTALLATION_____________'
      UNTBEQ(51)='_'
      ELMBEQ(53)='power_'
      ELMBEQ(54)='lighting_'
      ELMBEQ(55)='communication_'
      ELMBEQ(57)='transport_'
      ELMBEQ(59)='security_'
      DO 50 I=52,60
      IF (I.NE.56) THEN
        BEQ(I)=FFQTOT(1)
      ELSE
        BEQ(I)=FFQTOT(19)
      END IF
50    CONTINUE
C     *-------------------------------------------------
C     get FIXED FITTINGS fundamental quantities
C     *-------------------------------------------------
      ELMBEQ(61)='FIXED FITTINGS___________'
      UNTBEQ(61)='_'
      ELMBEQ(62)='circulation_'
      ELMBEQ(63)='general fittings_'
      ELMBEQ(64)='culinary_'
      ELMBEQ(65)='sanitary,hugiene_'
      ELMBEQ(66)='cleaning,maintenance_'
      ELMBEQ(67)='storage,screening_'
      ELMBEQ(68)='special activity_'
      DO 60 I=62,70
      IF (I.NE.65) THEN
        BEQ(I)=FFQTOT(1)
      ELSE
        BEQ(I)=FFQTOT(20)
      END IF
60    CONTINUE
C     *-------------------------------------------------
C     get LOOSE FITTINGS fundamental quantities
C     *-------------------------------------------------
      ELMBEQ(71)='LOOSE FITTINGS___________'
      UNTBEQ(71)='_'
      ELMBEQ(72)='circulation_'
      ELMBEQ(73)='general fittings_'
      ELMBEQ(74)='culinary_'
      ELMBEQ(75)='sanitary,hugiene_'
      ELMBEQ(76)='cleaning,maintenance_'
      ELMBEQ(77)='storage,screening_'
      ELMBEQ(78)='special activity_'
      DO 70 I=72,80
       BEQ(I)=FFQTOT(1)
```

```
70      CONTINUE
C       *-----------------------------------------------
C       get SITEWORKS fundamental quanitites
C       *-----------------------------------------------
        ELMBEQ(81)='SITEWORKS_____________'
        UNTBEQ(81)='_'
        ELMBEQ(82)='ground preparationa|_'
        ELMBEQ(83)='minor structures_'
        ELMBEQ(84)='enclosures_'
        ELMBEQ(85)='surface treatment_'
        ELMBEQ(86)='drainage_'
        ELMBEQ(87)='electric services_'
        ELMBEQ(88)='fittings_'
        ELMBEQ(89)='special_'
        DO 80 I=82,90
        IF (I.GT.85) THEN
         BEQ(I)=FFQTOT(1)
        ELSE
         BEQ(I)=FFQTOT(21)
        END IF
80      CONTINUE
C       *-----------------------------------------------
C       get PRELIMINARIES fundamental quantities
C       *-----------------------------------------------
        ELMBEQ(91)='PRELIMINARIES_____________'
        UNTBEQ(91)='_'
        ELMBEQ(92)='preliminaries_'
        ELMBEQ(93)='contingencies_'
        ELMBEQ(94)='price and design_'
C       *- - - - - - - - - - - - - - - - - - - - - - - -
C       seting CODE values.
C       *- - - - - - - - -  - - - - - - - - - - - - - - -
        RIGHT=.TRUE.
        J=1
        K=0
        DO 650 I=1,100
        CALL IROUND(J,STR,JTILD)
        CALL IROUND(K,CHR,KTILD)
        IF (RIGHT) THEN
          COD(I)=STR
          COD(I)(1:1)='('
          COD(I)(JTILD+1:)='-) _'
          RIGHT=.FALSE.
        ELSE
          COD(I)=STR
          COD(I)(1:1)='('
          COD(I)(JTILD+1:)=CHR(2:)
          TILD=INDEX(COD(I),'_')
          COD(I)(TILD:)=') _'
        END IF
        K=K+1
        IF(K.EQ.10) THEN
           K=0
           J=J+1
```

```
            RIGHT=.TRUE.
         END IF
650      CONTINUE
C        *----------------------------------------
         RETURN
         END
```

```
         ************************
        SUBROUTINE BTSELT(SEL,OK)
         ************************
C       *
C       selecting a building type from an index file, by producing a
C       report on the  selected type
C       *
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSC.COM'
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:COMMON:XXOUT.COM'
%INCLUDE ':GABLE:V31:COMMON:DVICES.COM'
%INCLUDE ':GABLE:V31:COMMON:DIALG.COM'
C       *
        INTEGER SEL,J,OPERAT(9),ERROR,LINES,I,K,N
        CHARACTER*15 FIELDS(9)
        CHARACTER*50 STRING(9),NAME,RNAME,STR,FNAMI,FNAM,CHR
        LOGICAL OK,DONE,SET,MESAGE,EXIST
        REAL*8 R
C       *
        OK=.TRUE.
        IROUTE=1
C       *
C       open the DMS LEVEL file
C       *
        FNAMI='INDEX_'
        CALL FILBLD('COST_MOD_','DMS_',FNAMI,FNAM)    !full pathname
C       *
        CALL INQUIR(FNAM,EXIST)
        IF (.NOT.EXIST) THEN
           CALL ERRMES('file not found_')
           OK=.FALSE.
           RETURN
        END IF
C       *
        CALL DMSTRA(FNAM,DONE)
        IF(.NOT.DONE) THEN
          CALL ERRMES('can not trace index file_')
          OK=.FALSE.
          RETURN
        END IF
        IF(NREC.LT.1) THEN
          CALL ERRMES('no levels in LEVEL file_')
          CALL DMSCLO
          OK=.FALSE.
          RETURN
        END IF
C       *
C       set up report format
C       *
        RAWDAT=.TRUE.
```

```
        NFLD=5
        FIELD=.TRUE.
        SPACE=.FALSE.
        HEAD=.FALSE.
        IRPFLD(1)=1
        IRPFLD(2)=3
        IRPFLD(3)=4
        IRPFLD(4)=5
        IRPFLD(5)=6
        IRPCOL(1)=1
        IRPCOL(2)=7
        IRPCOL(3)=45
        IRPCOL(4)=56
        IRPCOL(5)=67
        IRPW=78
        DO 100 I=1,NFLD
        TOTAL(I)=.FALSE.
100     CONTINUE
C       *
C       set up a survey analysis using SEAPAT
C       *
        CALL IROUND(SEL,STR,J)
        JTILD=INDEX(STR,'~')
        FIELDS(1)='TYPE~'
        STRING(1)=STR(2:)
        OPERAT(1)=1
        CALL SEAPAT(1,1,FIELDS,STRING,OPERAT,ERROR)
        IF (ERROR.NE.0) THEN
          CALL ERRMES('error in the search pattern~')
          RETURN
        END IF
C       *
C       report out on screen
C       *
        IROUTE=1                              !report out to screen
        SET=.TRUE.
        LINES=0
        MESAGE=.FALSE.
        CALL DPAGE
        CALL REPOUT(SET,LINES,MESAGE)              !report out
C       *
        CALL CHAROT('PROJECT SELECTION~')
        CALL CHARIN (0,'Select project code  : ~',0,CHR,K,N,R)
        IF (K.EQ.2) THEN
         IF (N.LT.1.OR .N.GT.NREC) THEN
C          CALL ERRMES('uncorrect project code number ~')
           OK=.FALSE.
           RETURN
         ELSE
           CALL PJSELT(N,OK)              !call to project selection
           IF (.NOT.OK) THEN
            CALL ERRMES ('can not select project type~')
            RETURN
           END IF
```

```
            END IF
          END IF
C         *
          RETURN
          END
```

```
      ****************************
      SUBROUTINE CALTOT(OFF,IDUM,OK)
      ****************************
C     *
C     calculat total of Floor Fundamental Quantities
C     *
%INCLUDE ':GABLE:V31:MOD385:TOTFFQ.COM'
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:DIALG.COM'
C     *
C    18-FEB-88 REV_1.16 TAB ground slab area just for floor 0
C     *
      CHARACTER*50 CHR,FNAM,ANSWER,QUEST*80
      REAL*8 R
      INTEGER I,N,LEV1,TEMP,LEV2,IDUM,NUM
      LOGICAL OK,OFF
C     *-----------------------------------------------
      DO 150 I=1,25
        FFQTOT(I)=0D0
150   CONTINUE
C     *
      NUM=0                                ! number of levels used
      OK=.TRUE.
      IF (.NOT.OFF) GOTO 400
C     ************************************************
200   CONTINUE                          !enter range of levels
      CALL CHAROT('_')
      CALL CHAROT('ff1948ffRange of levels in building model - _')
      QUEST = 'ff1949ffstart of range or B for whole building model
      _'
      CALL CHARIN(0,QUEST,0,ANSWER,K,LEV1,R)
      IF(K.EQ.0) RETURN                 !return pressed
      IF(K.EQ.1.AND.(ANSWER(1:1).EQ.'B'.OR.ANSWER(1:1).EQ.'b'))
                    THEN
        OFF=.FALSE.
        GOTO 400           !start normal BEM
      ELSE IF(K.EQ.1.OR.LEV1.LT.0.OR.LEV1.GT.99) THEN
        CALL ERRMES('ff1008ffmust be in range 0-99_')
        GOTO 200                        !try again
C
C     check starting level exists
C
      ELSE IF (LEVELN(LEV1+1).NE.LEV1) THEN
        CALL DROUND(DBLE(LEV1),ANSWER,0)
        J = INDEX(ANSWER,'_')-1
        PARAMS(1)=ANSWER
        NPARAM=1
        CALL ERRMES('ff712ffLevel %1% does not exist_')
        GOTO 200
      ENDIF
C
C     enter end of range
C
```

```
C       *
C       set up report format
C       *
        RAWDAT=.TRUE.
        NFLD=8
        FIELD=.TRUE.
        SPACE=.FALSE.
        HEAD=.FALSE.
        IRPFLD(1)=1
        IRPFLD(2)=2
        IRPFLD(3)=3
        IRPFLD(4)=4
        IRPFLD(5)=6
        IRPFLD(6)=11
        IRPFLD(7)=5
        IRPFLD(8)=10
        IRPCOL(1)=1
        IRPCOL(2)=7
        IRPCOL(3)=35
        IRPCOL(4)=40
        IRPCOL(5)=50
        IRPCOL(6)=59
        IRPCOL(7)=62
        IRPCOL(8)=73
        IRPW=80
        DO 150 I=1,NFLD-1
        TOTAL(I)=.FALSE.
150     CONTINUE
        TOTAL(7)=.TRUE.
        TOTAL(8)=.TRUE.
C       *
        DIREC=1
        SEL=1
        IF (ON) THEN IROUTE=1
        IF (IROUTE.EQ.1) THEN
200       CONTINUE
          CALL DPAGE
          IF (ON) THEN
            CALL SELELM(SEL,OK)           !for rates modelling
            IF (.NOT.OK) RETURN
          END IF
C         *
          IF (SEL.EQ.-1) RETURN           ! nothing selected yet
C         *
          IF (SEL.GT.10) THEN
            SEL=1
          ELSE IF (SEL.LT.1) THEN
            SEL=10
          END IF
          CALL SEARPT(SEL,OK)                  ! new
          IF (.NOT.OK) THEN
            CALL ERRMES('error in search pattern')
            RETURN
          END IF
```

```
C           *
            SET=.TRUE.
            LINES=0
            MESSAGE=.FALSE.
            CALL REPOUT(SET,LINES,MESSAGE)
C           *
            CHR=('A to abandon or Return to continue
     +             (Next/Previous)..')
            CALL CHARIN(0,CHR,0,CHR,K,N,R)
            IF (K.LT.2) THEN
              CALL CHRULC(CHR)
              IF(CHR(1:1).EQ.'A') THEN
                CALL CLROUT
                RETURN
              END IF
              IF(CHR(1:1).EQ.'N') THEN
                DIREC=1
              ELSE IF (CHR(1:1).EQ.'P') THEN
                DIREC=0
              END IF
              IF (DIREC.EQ.1) THEN
                SEL=SEL+1
              ELSE
                SEL=SEL-1
              END IF
            ELSE
              IF (N.LE.10.AND.N.GE.1) THEN
                SEL=N
              END IF
              IF (ON) THEN
                ELEM=N
                CALL DPAGE
                CALL ADJRAT(SEL,ELEM,OK)        !for rate adjustement
                IF (.NOT.OK) RETURN
              END IF
            END IF
            GOTO 200
          ELSE
            DO 300 I=1,10
             IDUM=I
             CALL SEARPT(IDUM,OK)                    ! new
             IF (.NOT.OK) THEN
               CALL ERRMES('error in search pattern')
               RETURN
             END IF
C           *
            SET=.TRUE.
            LINES=0
            MESSAGE=.FALSE.
            CALL REPOUT(SET,LINES,MESSAGE)
            CALL CHAROT(' ')
300          CONTINUE
            CALL TOTOUT                             ! printout total
            CALL CLROUT
```

```
            END IF
C           *
            RETURN
            END
```

```
        ****************************************
        SUBROUTINE FILREP(FLDR,SUBT,SET,ERROR)
        ****************************************
C       *
C       replaces the content of field FLDR by SUBT
C           - in FLDR field to replace
C           - in SUBT value to replace it with
C           - in SET  if true only on selected set of records
C           - out ERROR
C       *
%INCLUDE 'GABLE:V31:DMS:DMSA.COM'
%INCLUDE 'GABLE:V31:DMS:DMSB.COM'
%INCLUDE 'GABLE:V31:DMS:DMSC.COM'
%INCLUDE 'GABLE:V31:DMS:DMSL.COM'
%INCLUDE 'GABLE:V31:DMS:DMS.COM'
C
        CHARACTER*15 FLDR(9)
        CHARACTER*50 SUBT(9)
        INTEGER FLN(9),NB,K
        LOGICAL SET,ERROR,DONE
C       *
        DONE=.TRUE.
        NB=0
        DO 100 I=1,9
        IDUM=I
        IF (FLDR(IDUM)(1:2).EQ.' ') GOTO 100
        CALL FINFLD(FLDR(IDUM),N)
        ITARG=N
C        *
        IF (ITARG.EQ.0) THEN        ! field does not exist
           CALL ERRMES ('selected field does not exist')
           ERROR=.FALSE.
           RETURN
        END IF
C        *
        IF (LOCFLD(ITARG).NE.0) THEN
           CALL ERRMES('selected field is locked cannot be
     +                 replaced')
           ERROR=.FALSE.
           RETURN
        END IF
        CALL CHARAN(SUBT(IDUM),K,N,R)
C       *
        IF(K.NE.1.OR.(IFLDTP(ITARG).NE.1).AND.IFLDTP(ITARG).NE.-1)
     +         THEN
          IF (K.NE.2.OR.(IFLDTP(ITARG).NE.2)) THEN
             IF (K.NE.3.OR.(IFLDTP(ITARG).NE.3)) THEN
                CALL ERRMES('mismatch error ')
                ERROR=.FALSE.
                RETURN
             END IF
          END IF
        END IF
C        *
```

```
        FLN(IDUM)=ITARG          ! field number
        NB=NB+1
100     CONTINUE
C       *
        CALL REPREC(NB,FLN,SUBT,SET,DONE)
        IF (.NOT.DONE) THEN
           CALL ERRMES('error at replacing record')
           RETURN
        END IF
C       *
        RETURN
        END
```

```
      ******************************************************
      SUBROUTINE FFQMNR(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
      ******************************************************
C     *
C     floor fundamental quantities nain routine
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER *50 FNAML,FNAMB,FNAMI,FNAMR,FNAMO,NAME,
     +      FNAM,MENUT(20)
      INTEGER I
      LOGICAL OK,SET,DONE
C     *
150   CONTINUE
      MENUT(1)='FLOOR FUNDAMENTAL QUANTITIES '
      MENUT(2)='using Space Fundamental Quantities file '
      MENUT(3)='using a selected number of Spaces '
      MENUT(4)='using the 5 building survey files '
C     *
      CALL MENU(MENUT,1,3,I)
      GOTO (1000,2000,3000) I
      RETURN
1000  CONTINUE
C     *
C     creating FFQ from Space Fundamental Quantities
C     *
      NAME='SFQ '
      CALL GETIFL(NAME,FNAM,OK)
      IF(.NOT.OK) THEN
C      CALL ERRMES('can not get SFQ file ')
       RETURN
      END IF
C     *
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('can not trace SFQ file ')
        OK=.FALSE.
        RETURN
      END IF
C     *
      IF(NREC.LT.1) THEN
       CALL ERRMES('no element in SFQ file ')
       OK=.FALSE.
       RETURN
      END IF
C     *
      SET=.FALSE.
      CALL SFTOFF(SET,OK)
      IF(.NOT.OK) RETURN
      NAME='FFQ '
      CALL GETOFL(NAME,FNAM,OK)
      IF (.NOT.OK) RETURN
```

```
        CALL CREFFQ(FNAM)
        CALL CHAROT('storing file '//FNAM)
        CALL ADDFFQ(FNAM,OK)
        CALL DMSCLO
        GOTO 150
C       *
C       *
 2000   CONTINUE
C       *
        CALL SGRMNR(OK)
        IF (.NOT.OK) GOTO 150
        NAME='FFQ_'
        CALL GETOFL(NAME,FNAM,OK)
        IF (.NOT.OK) RETURN
        CALL CREFFQ(FNAM)
        CALL CHAROT('storing file '//FNAM)
        CALL ADDFFQ(FNAM,OK)
        CALL DMSCLO
        GOTO 150
3000    CONTINUE
C       *
        CALL GET5FL(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
        IF (.NOT.OK) RETURN
C       *
C       creating FFQ from the 5 building survey files
C       *
        CALL CHAROT('getting level data_')
        CALL FFQPT1(FNAML,OK)
        IF (.NOT.OK) RETURN
        CALL CHAROT('getting layout data_')
        CALL FFQPT2(FNAMB,OK)
        IF (.NOT.OK) RETURN
        CALL CHAROT('getting surface data_')
        CALL FFQPT3(FNAMI,OK)
        IF (.NOT.OK) RETURN
        CALL CHAROT('getting room data_')
        CALL FFQPT4(FNAMR,OK)
        IF (.NOT.OK) RETURN
        CALL CHAROT('getting opening data_')
        CALL FFQPT5(FNAMO,OK)
        IF (.NOT.OK) RETURN
        SET=.FALSE.
        CALL COMFFQ(SET)          ! complementary FFQs per floor
        NAME='FFQ_'
        CALL GETOFL(NAME,FNAM,OK)
        IF (.NOT.OK) RETURN
        CALL CREFFQ(FNAM)
        CALL CHAROT('storing file '//FNAM)
        CALL ADDFFQ(FNAM,OK)
        CALL DMSCLO
        GOTO 150
C       *
        END
```

```
      ****************************
      SUBROUTINE FFQPT1(FNAML,OK)
      ****************************
C     *
C     opens dms level file and extracts data into FFQ common
C     *
      CHARACTER*50 FNAML,CHR,STR
      LOGICAL OK,DONE
      REAL*8 R
      INTEGER I,J,N,IFL,IFV,IFG,IFD,IDUM,TOPLEV,BOTLEV,K,IFN
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
C     initialise LEVELN to 0
C     *
      OK=.FALSE.
      DO 100 I=1,100
        LEVELN(I)=0
        GRAREA(I)=0D0
        GSAREA(I)=0D0
        LUAREA(I)=0D0
        EXSTHT(I)=0D0
        INSTHT(I)=0D0
        EXWAAR(I)=0D0
        INWAAR(I)=0D0
        EXWIAR(I)=0D0
        INWIAR(I)=0D0
        EXDOAR(I)=0D0
        INDOAR(I)=0D0
        ROAREA(I)=0D0
        CEAREA(I)=0D0
        EXWAGI(I)=0D0
        DSTND(I)=0D0
        PILEHD(I)=0
        STRFLT(I)=0
        LFTFLT(I)=0
        SUSCLG(I)=0D0
        SANFIT(I)=0
        NUMSPA(I)=0
100   CONTINUE
C     *
C     open the DMS LEVEL file
C     *
      CALL DMSTRA(FNAML,DONE)
      IF(.NOT.DONE) RETURN
      IF(NREC.LT.1) THEN
        CALL ERRMES('no levels in LEVEL file')
        CALL DMSCLO
        RETURN
      END IF
C     *
      CALL FINFLD('LEVEL ',IFL)
      CALL FINFLD('GROSS AREA ',IFG)
```

```
      CALL FINFLD('EST VOLUME ',IFV)
      CALL FINFLD('DATUM ',IFD)
      CALL FINFLD('DOWNSTAND ',IFS)
      CALL FINFLD('NO OF ROOMS ',IFN)
      IF(IFL.EQ.0.OR.IFG.EQ.0.OR.IFV.EQ.0.OR.IFD.EQ.0.
     +               OR.IFS.EQ.0)  THEN
        CALL ERRMES('LEVEL file format error ')
        CALL DMSCLO
        RETURN
      END IF
C     *
      DO 200 I=1,NREC
        IDUM=I
        CALL GETFLD(IDUM,IFL,CHR,K,N,R)
        IF(N.GE.0.AND.N.LE.99) THEN         !valid level no
          LEVELN(N+1)=N+1
          CALL GETFLD(IDUM,IFG,CHR,K,J,R)
          GRAREA(N+1)=R                     !gros  floor area
          LUAREA(N+1)=R
          CALL GETFLD(IDUM,IFV,CHR,K,J,R)
          INSTHT(N+1)=R/GRAREA(N+1)*1000D0 !internal storey height
          CALL GETFLD(IDUM,IFD,CHR,K,J,R)
          EXSTHT(N+1)=R                     !datums
          CALL GETFLD(IDUM,IFS,CHR,K,J,R)  !downstand
          DSTND(N+1)=R
          CALL GETFLD(IDUM,IFN,CHR,K,J,R)
          NUMSPA(N+1)=J           !number of rooms per floor
        END IF
200   CONTINUE
      CALL DMSCLO
C     ------------------------------------------------
C     get right content of EXSTHT from datums
C     ------------------------------------------------
      BOTLEV=0
      DO 300 I=1,100
        IF (LEVELN(I).EQ.I) THEN
          TOPLEV=I
          IF (BOTLEV.EQ.0) THEN
            BOTLEV=I
          END IF
        END IF
300   CONTINUE
      IF (TOPLEV.GT.BOTLEV) THEN
         DO 400 I=BOTLEV,TOPLEV-1
          EXSTHT(I)=EXSTHT(I+1)-EXSTHT(I) !getting EXSTHT right
400      CONTINUE
         EXSTHT(TOPLEV)=EXSTHT(TOPLEV-1)  !in stade of
C                                             INSTHT(TOPLEV)
      ELSE
         EXSTHT(TOPLEV)=INSTHT(TOPLEV)    !use int storey height
      END IF
C     - - - - - - - - - - - - - - - - - - - - - - - - -
C     get EXSTHT fot the top level of the building
C     - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
450     CONTINUE
        CALL CHAROT('give external storey height for top level_')
        CALL DROUND(INSTHT(TOPLEV),STR,-1)
        CHR='in building [ '//STR(2:)
        ITILD=INDEX(CHR,'_')
        CHR(ITILD:)='mms ] _'
        CALL CHARIN(0,CHR,0,CHR,K,N,R)
        IF (K.EQ.1) THEN
          CALL ERRMES('reply as requested_')
          GOTO 450
        ELSE IF(K.GE.2) THEN
          EXSTHT(TOPLEV)=R
          IF (EXSTHT(TOPLEV).LT.INSTHT(TOPLEV)) THEN
            CALL CHAROT('this is less then the INternal
                    STorey HeighT_')
            GOTO 450
          END IF                              ! testing if block
        END IF
        OK=.TRUE.
        RETURN
        END
```

```
      ****************************
      SUBROUTINE FFQPT2(FNAMB,OK)
      ****************************
      IMPLICIT NONE
C     *
C     opens dms LAYOUT file and extracts data for FFQ common
C     *
      CHARACTER*15 FIELD(9)
      CHARACTER*50 FNAMB,FILE,STR1,STR2,TOT,FNAMBW,CHR,STRING(9)
      REAL*8 R,TOTAL
      LOGICAL OK,DONE,SET
      INTEGER I,N,ERROR,IDUM,ILOW,ITILD,OPERAT(9)
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
C     open the dms LAYOUT file
C     *
      STR2='_'
      CALL DMSTRA(FNAMB,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in LAYOUT file_')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     *
C     copy file FNAMB in a work file called FNAMBW
C     *
      CALL FILBLD(JOB,'DMS_','FNAMBW_',FNAMBW)
      CALL CPYTRA(FNAMB,FNAMBW,ERROR)
C     *
C     cross reference thicknesses from dms SURFACE file
C     built first its full path name
C     *
      CALL FILBLD(JOB,'DMS_','SURFACES_',FILE)
      CALL CRSREF('THICKNESS_','SPEC_',FILE,'THICKNESS_','SPEC_',
     1            .FALSE.,ERROR)
      IF(ERROR.NE.0) THEN
        CALL DMSCLO
        OK=.FALSE.
        RETURN
      END IF
C     *
C     compute fields GIRTH * THICKNESS and store in WALL BED
C     *
      SET=.FALSE.
      CALL COMPUT('WALL BED_','GIRTH*(THICKNESS/1000)_',SET,ERROR)
      IF (ERROR.NE.0) THEN
        CALL DMSCLO
        OK=.FALSE.
        RETURN
```

```
      END IF
C     ------------------------------------------------------
C     NOW going through all existing floor levels
C     ------------------------------------------------------
      ILOW=0          !not yet found lowest level in building
      DO 200 I=1,100
        IDUM=I
        IF (LEVELN(I).EQ.I) THEN     !this level is defined
          IF(ILOW.EQ.0) THEN         !this is the lowest level
            ILOW=1                   !consider slab area
          ELSE
            ILOW=2                   !slab area already considered
          END IF
C         *-----------------------------------------------------
C         get complementary value for GRAREA for FFQ.COM
C         *-----------------------------------------------------
          STR1='INTERNAL WALL_'
          TOT='WALL BED_'
          CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
          IF (.NOT.DONE) THEN
            CALL ERRMES('failed in calculating GRAREA_')
            OK=.FALSE.
            CALL DMSCLO
            RETURN
          END IF
          GRAREA(I)=GRAREA(I)+R
C         *
C         if lowest level then calc slab area
C         *
          IF(ILOW.EQ.1) THEN
C           *-----------------------------------------------------
C           get complementary value for GSAREA for FFQ.COM
C           *-----------------------------------------------------
            STR1='EXTERNAL WALL_'
            TOT='WALL BED_'
            CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
            IF (.NOT.DONE) THEN
              CALL ERRMES('failed in calculating GSAREA_')
              OK=.FALSE.
              CALL DMSCLO
              RETURN
            END IF
            GSAREA(I)=GRAREA(I)+R
          END IF
C         *-----------------------------------------------------
C         get ROAREA for FFQ.COM
C         *-----------------------------------------------------
          CALL IROUND(LEVELN(I)-1,CHR,ITILD)
          FIELD(1)='LEVEL_'
          STRING(1)(1:ITILD)=CHR(2:ITILD+1)
          OPERAT(1)=1
          FIELD(2)='ELEMENT_'
          STRING(2)='ROOF_'
          OPERAT(2)=1
```

```
            FIELD(3)='ELEMENT_'
            STRING(3)='CEILING_'
            OPERAT(3)=2
            CALL SEAPAT(3,1,FIELD,STRING,OPERAT,ERROR)
            IF (ERROR.NE.0) THEN
              CALL ERRMES('error in the search pattern format_')
              OK=.FALSE.
              RETURN
            END IF
C           *- - - - - - - - - - - -  - - - - - - - - - - - -
C           calculate total of roof areas
C           *- - - - - - - - - - - -  - - - - - - - - - - - -
            CALL TOTFLD('GROSS AREA_',.TRUE.,TOTAL,N,ERROR)
            IF (ERROR.NE.0) THEN
              CALL ERRMES('can not calculate total of ROOF AREA_')
              CALL CHAROT(DBSNAM)
              OK=.FALSE.
              RETURN
            ELSE
            END IF
            ROAREA(I)=TOTAL                ! total of roof area
C           *-----------------------------------------------
C           get EXWAGI for FFQ.COM
C           *-----------------------------------------------
            STR1='EXTERNAL WALL_'
            TOT='GIRTH_'
            CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
            IF (.NOT.DONE) THEN
              CALL ERRMES('failed in calculating EXWAGI_')
              OK=.FALSE.
              RETURN
            END IF
            EXWAGI(I)=R
C           *-----------------------------------------------
C           get EXWAAR for FFQ.COM
C           *-----------------------------------------------
            STR1='EXTERNAL WALL_'
            TOT='GROSS AREA_'
            CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
            IF (.NOT.DONE) THEN
               CALL ERRMES('failed in calculating EXWAAR_')
               OK=.FALSE.
               RETURN
            END IF
            EXWAAR(I)=R+EXWAGI(I)*(DSTND(I)/1000D0) ! downstand used
C           *-----------------------------------------------
          END IF
200     CONTINUE
        CALL DMSCLO
        OK=.TRUE.
        RETURN
        END
```

```
      ****************************
      SUBROUTINE FFQPT3(FNAMI,OK)
      ****************************
C     *
C     opens dms ISAAC file and extracts data for FFQ common
C     *
      CHARACTER*50 FNAMI,STR1,STR2,TOT
      REAL*8 R
      LOGICAL OK,DONE
      INTEGER IDUM,I
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
C     open th dms ISAAC file
C     *
      STR2='_'
      CALL DMSTRA(FNAMI,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in ISAAC file_')
        OK=.FALSE.
        CALL DMSCLO
          RETURN
      END IF
C     *
C     ----------------------------------------
C     NOW going through all existing floor levels
C     ----------------------------------------
      DO 300 I=1,100
        IDUM=I
        IF (LEVELN(I).EQ.I) THEN
C         *--------------------------------------
C         get INWAAR for FFQ.COM
C         *--------------------------------------
          STR1='INTERNAL WALL_'
          TOT='GROSS AREA_'
          CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
          IF (.NOT.DONE) THEN
            CALL ERRMES('failed in calculating INWAAR_')
            OK=.FALSE.
            CALL DMSCLO
            RETURN
          END IF
          INWAAR(I)=R
C         *--------------------------------------
C         get CEAREA for FFQ.COM
C         *--------------------------------------
          STR1='CEILING_'              !NEED VERIFICATION
          TOT='GROSS AREA_'
          CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
          IF (.NOT.DONE) THEN
            CALL ERRMES('failed in calculating CEAREA_')
             OK=.FALSE.
```

```
                CALL DMSCLO
              RETURN
            END IF
            CEAREA(I)=R
C           *-----------------------------------------
          END IF
 300    CONTINUE
        CALL DMSCLO
        OK=.TRUE.
        RETURN
        END
```

```
            ***************************
            SUBROUTINE FFQPT5(FNAMO,OK)
            ***************************
C           *
C           opens dms OPENING file and extracts data for FFQ common
C           *
            CHARACTER*50 FNAMO,STR1,STR2,TOT
            REAL*8 R
            LOGICAL OK,DONE
            INTEGER KO
C           *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C           *
C           open the dms OPENING file
C           *
            STR2='_'
            CALL DMSTRA(FNAMO,DONE)
            IF (.NOT.DONE) RETURN
            IF (NREC.LT.1) THEN
              CALL ERRMES('no element in OPENING file_')
            OK=.FALSE.
              CALL DMSCLO
            END IF
C           *
C           ------------------------------------------------------
C           NOW going through all existing floor levels
C           ------------------------------------------------------
            DO 500 I=1,100
            IDUM=I
              IF (LEVELN(I).EQ.I) THEN
C               *-------------------------------------------------
C               get EXWIAR for FFQ.COM
C               *-------------------------------------------------
                STR1='WINDOW_'
                STR2='EXTERNAL_'
                TOT='GROSS AREA_'
                CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
                IF (.NOT.DONE) THEN
                  CALL ERRMES('failed in calculating EXWIAR_')
                  OK=.FALSE.
                  CALL DMSCLO
                  RETURN
                END IF
                EXWIAR(I)=R
C               *-------------------------------------------------
C               get INWIAR for FFQ.COM
C               *-------------------------------------------------
                STR1='WINDOW_'
                STR2='INTERNAL_'
                TOT='GROSS AREA_'
                CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
                IF (.NOT.DONE) THEN
                  CALL ERRMES('failed in calculating INWIAR_')
                  OK=.FALSE.
                  CALL DMSCLO
```

```
                RETURN
              END IF
              INWIAR(I)=R
C             *-------------------------------------------------
C             get EXDOAR for FFQ.COM
C             *-------------------------------------------------
              STR1='DOOR_'
              STR2='EXTERNAL_'
              TOT='GROSS AREA_'
              CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
              IF (.NOT.DONE) THEN
                CALL ERRMES('failed in calculating EXDOAR_')
                OK=.FALSE.
                CALL DMSCLO
                RETURN
              END IF
              EXDOAR(I)=R
C             *-------------------------------------------------
C             get INDOAR for FFQ.COM
C             *-------------------------------------------------
              STR1='DOOR_'
              STR2='INTERNAL_'
              TOT='GROSS AREA_'
              CALL SEAPT1(IDUM,STR1,STR2,TOT,R,DONE)
              IF (.NOT.DONE) THEN
                CALL ERRMES('failed in calculating INDOAR_')
                OK=.FALSE.
                CALL DMSCLO
                RETURN
              END IF
              INDOAR(I)=R
C             *-------------------------------------------------
            END IF
500       CONTINUE
          CALL DMSCLO
          OK=.TRUE.
          RETURN
          END
```

```
            ******************************
            SUBROUTINE GETOFL(NAME,FNAM,OK)
            ******************************
C           *
C           get output DMS file names for cost modeller
C           *
            CHARACTER*50 FNAM,NAME,FNAME,CHR,TXT
            LOGICAL OK,EXIST
C           *
C      26-JAN-88 REV_1.16 TAB type "!" to get out the routine
C        *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C        *
           OK=.TRUE.
          EXIST=.FALSE.
          FNAM='_'
C           *-------------------------------------------
100       CONTINUE
            CALL CHAROT('name of DMS file where data to be stored_')
          CHR='give filename ['// NAME
          I=INDEX(CHR,'_')
          CHR(I:)='] _'
            CALL CHARIN(0,CHR,0,FNAME,K,N,R)
            IF (K.EQ.0) THEN
              FNAME=NAME
            ELSE IF (K.NE.1) THEN
              CALL ERRMES('improper filename_')
              OK=.FALSE.
              RETURN
          END IF
          CALL ABATST(FNAME,OK)                    !testing "!"
          IF (.NOT.OK) RETURN
          CALL FILBLD(JOB,'DMS_',FNAME,FNAM)       !full pathname
          CALL INQUIR(FNAM,EXIST)
          IF (EXIST) THEN
           CALL CHAROT('this file already exists_')
           TXT='do you wish to continue ..Y/N..[N] _'
           CALL CHARIN(0,TXT,0,CHR,K,N,R)
            IF (CHR(1:1).EQ.'Y'.OR.CHR(1:1).EQ.'y') THEN
             GOTO 200
            ELSE
             EXIST=.FALSE.
             GOTO 100
            END IF
          END IF
200       CONTINUE
            RETURN
            END
```

```
      ******************************
      SUBROUTINE GETIFL(NAME,FNAM,OK)
      ******************************
C     *
C     get input DMS file names for cost modeller
C     *
C    26-JAN-88 REV_1.16 TAB type "!" to leave the routine
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*50 FNAM,NAME,FNAME,CHR
      LOGICAL OK,EXIST
C     *
100   CONTINUE
      OK=.TRUE.
      EXIST=.TRUE.
C     *-----------------------------------------
      CALL CHAROT('name of DMS file to be used ')
      CHR='give filename ['// NAME
      I=INDEX(CHR,' ')
      CHR(I:)='] '
      CALL CHARIN(0,CHR,0,FNAME,K,N,R)
      IF (K.EQ.0) THEN
        FNAME=NAME
      ELSE IF (K.NE.1) THEN
        CALL ERRMES('improper filename ')
        OK=.FALSE.
        RETURN
      END IF
      CALL ABATST(FNAME,OK)                  !testing "!"
      IF (.NOT.OK) RETURN
      CALL FILBLD(JOB,'DMS ',FNAME,FNAM)     !full pathname
      CALL INQUIR(FNAM,EXIST)
      IF (.NOT.EXIST) THEN
       CALL ERRMES('this file does not exist ')
       GOTO 100
      END IF
C     *
      RETURN
      END
```

```
      ************************************************
      SUBROUTINE GET5FL(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
      ************************************************
C     *
C     Get five FILE names from COMPLETE BUILDING SURVEY done
C     by MOD630
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      LOGICAL OK,GETYN
      CHARACTER*50 FNAML,FNAMB,FNAMR,FNAMI,FNAMO,FNAME,CHR
C     *
      OK=.TRUE.
C     *
      CALL CHAROT(' ')
      CALL CHAROT('you should have surveyed the BUIDING'//
     +            ' model using MOD630 ')
      CALL CHAROT('if it is to be used for COST MODELLING')
      CHR=('do you wish to continue ')
      IF (GETYN(CHR,'N')) THEN
       OK=.FALSE.
       RETURN
      END IF
100   CONTINUE
C     *
      CALL CHAROT(' ')
      CALL CHAROT('the complete_building survey has created'//
     +            ' 5 DMS files')
      CALL CHAROT('the default names for these files is:-')
      CALL CHAROT('LEVEL ROOM ISAAC OPENING LAYOUT')
      CALL CHAROT('and any files of those_names already'//
     +            ' existing will be used')
      CALL CHAROT('or you may select your own names')
      CALL CHAROT(' ')
      CHR='do you want default names '
      IF (.NOT.GETYN(CHR,'Y')) THEN
C        ---------------------------------------------
C        GET file equivalent to 'LEVEL' of MOD630
C        ---------------------------------------------
         FNAME='LEVEL'
         CALL GETIFL(FNAME,FNAML,OK)
         IF (.NOT.OK) RETURN
C        ---------------------------------------------
C        GET file equivalent to 'LAYOUT' of MOD630
C        ---------------------------------------------
         FNAME='LAYOUT'
         CALL GETIFL(FNAME,FNAMB,OK)
         IF (.NOT.OK) RETURN
C        ---------------------------------------------
C        GET file equivalent to 'ROOM' of MOD630
         FNAME='ROOM'
         CALL GETIFL(FNAME,FNAMR,OK)
         IF (.NOT.OK) RETURN
C        ---------------------------------------------
```

```
C           GET file equivalent to 'ISAAC' of MOD630
C           ----------------------------------------------
            FNAME='ISAAC_'
            CALL GETIFL(FNAME,FNAMI,OK)
            IF (.NOT.OK) RETURN
C           ----------------------------------------------
C           GET file equivalent to 'OPENING' of MOD630
C           ----------------------------------------------
            FNAME='OPENING_'
            CALL GETIFL(FNAME,FNAMO,OK)
            IF (.NOT.OK) RETURN
         ELSE
            FNAME='LEVEL_'
            CALL FILBLD(JOB,'DMS_',FNAME,FNAML)
            FNAME='LAYOUT_'
            CALL FILBLD(JOB,'DMS_',FNAME,FNAMB)
            FNAME='ROOM_'
            CALL FILBLD(JOB,'DMS_',FNAME,FNAMR)
            FNAME='ISAAC_'
            CALL FILBLD(JOB,'DMS_',FNAME,FNAMI)
            FNAME='OPENING_'
            CALL FILBLD(JOB,'DMS_',FNAME,FNAMO)
         END IF
C
         OK=.TRUE.
         RETURN
         END
```

```
      ************************
      SUBROUTINE GTSURF(SPEC,OK)
      ************************
C     *
C
C
C
%INCLUDE ':GABLE:V31:BMS:BMSSP1.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*50 CHR,STR
      INTEGER SPEC
      LOGICAL OK
C     *
      OK=.TRUE.
      CALL CHAROT(' ¯')
      CALL CHAROT('internal wall bordering excluded spaces will'//
     +    ' need new wall specification¯')
100   CONTINUE
      CHR='give external wall specification..¯'
      CALL CHARIN(0,CHR,0,STR,K,N,R)
      IF (K.EQ.0) THEN
         OK=.FALSE.
         RETURN
      END IF
      IF (K.EQ.1.OR.N.LT.1.OR.N.GT.99) THEN
         CALL ERRMES('must be in range 1-99¯')
         GOTO 100
      END IF
C     *
      IF (WALL(N).EQ.1) THEN
       SPEC=N
       RETURN
      END IF
C     *
      CALL ERRMES('un-specified material ¯')
      OK=.FALSE.
      RETURN
      END
```

```
      ****************************
      SUBROUTINE ISAREP(SPEC,DONE)
      ****************************
C     *
C     replace data in ISAAC file
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*15 FIELD(9)
      CHARACTER*50 FNAMIW,SUBT(9),STR
      INTEGER SPEC
      LOGICAL SET,DONE,OK
C     *-------------------------------------
      DONE=.TRUE.
      DO 100 I=1,9
       FIELD(I)='_'
       SUBT(I)='_'
100   CONTINUE
C     *
      FIELD(1)='SPEC_'
      CALL IROUND(SPEC,STR,ITILD)
      SUBT(1)(1:ITILD)=STR(2:ITILD+1)
      FIELD(2)='ELEMENT_'
      SUBT(2)='EXTERNAL WALL_'
      FIELD(3)='ROOM_'
      SUBT(3)='0_'
      SET=.TRUE.
      CALL FILREP(FIELD,SUBT,SET,DONE)
      IF (.NOT.DONE) THEN
         CALL ERRMES('error at replacing ISAAC data_')
         RETURN
      END IF
C     *
      CALL DMSCLO
C     *
      RETURN
      END
```

```
      ****************************
      SUBROUTINE LAYREP(SPEC,DONE)
      ****************************
C     *
C     replace data in ISAAC file
C     *
%INCLUDE':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*15 FIELD(9)
      CHARACTER*50 FNAMBW,SUBT(9),STR
      INTEGER SPEC
      LOGICAL SET,DONE,OK
C     *--------------------------------------
      DONE=.TRUE.
      DO 100 I=1,9
       FIELD(I)=' '
       SUBT(I)=' '
100   CONTINUE
      C       *
      FIELD(1)='SPEC '
      CALL IROUND(SPEC,STR,ITILD)
      SUBT(1)(1:ITILD)=STR(2:ITILD+1)
      FIELD(2)='ELEMENT '
      SUBT(2)='EXTERNAL WALL '
      FIELD(3)='ROOM NUMBER 1 '
      SUBT(3)='0 '
      FIELD(4)='ROOM NUMBER 2 '
      SUBT(4)='0 '
      SET=.TRUE.
      CALL FILREP(FIELD,SUBT,SET,DONE)
      IF (.NOT.DONE) THEN
         CALL ERRMES('error at replacing LAYOUT data ')
         RETURN
      END IF
C     *
      CALL DMSCLO
C     *
      RETURN
      END
```

```
*************
PROGRAM M385
*************

C       *
C       BMS MODULE M385  BMS cost modelling
C       *
C     02-FEB-88 REV_1.16  TAB include DVICES.COM
C     03-MAY-88 REV_1.16  HND call LINK to exit
C       *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:COMMON:BOARD.COM'
%INCLUDE ':GABLE:V31:COMMON:DVICES.COM'
%INCLUDE ':GABLE:V31:COMMON:XXOUT.COM'
%INCLUDE ':GABLE:V31:COMMON:HLP.COM'
C       *
        CHARACTER*50 CHR,STR,MENUT(20)
        LOGICAL OK,DONE,EXIST
        REAL*8 R
C       *
        CALL INIT
        CALL SYSINI(.FALSE.,IER)
        IF(IER.NE.0) STOP
        HLPMOD(1:6) = 'D_385_'
        CALL HLPINI        ! init help refs
        CALL GRFINI
CDS>>>
        CALL ALFINI('M385_')
C       CALL TRMSET
CDS<<<
        CALL ALFAON
        MODULE='BUILDING_'
100     CONTINUE
        MENUT(1)='GABLE BMS  - module_385 COST MODELLER_'
        MENUT(2)='create a cost model_'
        MENUT(3)='directory_'
        MENUT(4)='summary of building model_'
        MENUT(5)='list/display contents of a DMS file_'
        MENUT(6)='list constructions_'
        HLPLVL = 1
        CALL MENU(MENUT,1,5,IMENU)
        HLPLVL = 0
        GOTO (1000,2000,3000,4000,5000) IMENU
        CALL CHAROT(' _')           !LINK
        CALL CHAROT('LINK_')
        CALL CHAROT(' _')
C       CHR='do you really wish to leave this module..Y/N.. _'
C       CALL CHARIN(0,CHR,0,CHR,K,N,R)
C       IF(CHR(1:1).NE.'Y'.AND.CHR(1:1).NE.'y') GOTO 100
        CALL LINK(OK)
        IF(.NOT.OK) GOTO 100
```

```
      GOTO 9999
C
C      BUILDING COST MODELLING **********( Tami )********
1000  CONTINUE
      CALL CMVER3          !main cost model routine
      CALL CHAROT('_')
      CALL CHARIN(0,'press RETURN to continue_',0,CHR,K,N,R)
      GOTO 100
C     directory*****************************************
2000  CONTINUE
      CALL LISDIR(MODULE)
      CALL CHARIN(0,'press RETURN to continue_',0,CHR,K,N,R)
      GOTO 100
C     building model summary*********************************
3000  CONTINUE
      CALL MODSUM
      GOTO 100
C     display dms file***************************************
4000  CONTINUE
      CALL DISFIL
      GOTO 100
C     LIST CONSTRUCTIONS*************************************
5000  CONTINUE
      CALL WALINI
      CALL WALIST
      GOTO 100
9999  CONTINUE
      END
```

```
      **************************
      SUBROUTINE PJSELT(PRJ,OK)
      **************************
C     *
C     project selection routine
C     *
      CHARACTER*50 CHR,MENUT(20)
      INTEGER K,N,PRJ
      REAL*8 R
      LOGICAL OK
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
C     *
      OK=.TRUE.
100   CONTINUE
C     *
      CALL DPAGE                                        ! clear screen
      MENUT(1)='SELECTED PROJECT¯'
      MENUT(2)=' display details on selected project¯'
C     MENUT(3)=' display rates on selected project¯'
      MENUT(3)=' import rates of selected project¯'
      CALL MENU(MENUT,0,2,I)
      GOTO (1000,2000)I
      RETURN
C     *
1000  CONTINUE
      CALL DPAGE
      CALL DIS1RC(PRJ)                  !display selected poject
      CALL CHAROT(' ¯')
      CALL CRWAIT
      GOTO 100
C     *
2000  CONTINUE
      CALL UPDRAT(OK)
      IF (.NOT.OK) RETURN
      GOTO 100
C     *
      END
```

```
      ******************
      SUBROUTINE PRCENT
      ******************
C     *
C     calculate percentages for each record in BEM file after
C     update and computation
C     *
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:CALMEM.COM'
C     *
      INTEGER ITARG,ERROR,COUNT
      REAL*8 R,TOTAL
      CHARACTER*50 CHR,CALCHR,FIELD
      LOGICAL SET,DONE
C     *
      SET=.FALSE.
      TOTAL=0D0
      FIELD='ELEMENTAL COST_'
      CALL TOTFLD(FIELD,SET,TOTAL,COUNT,ERROR)
      IF (ERROR.NE.0) THEN
        CALL ERRMES('failed calculating total of field_')
        RETURN
      END IF
      CALAZ(20)=TOTAL/100
C     *
C     work out percentages of ELEMENTAL COST in BEQ file
C     *
      IF (T.LE.0) THEN
       CALL ERRMES('illegal operation, no total cost_')
       RETURN
      END IF
C     *
      ITARG=10                                  ! percent
      CALCHR='ELEMENTAL COST/T_'
      NDECP=1
      CALL COMPRC(ITARG,CALCHR,SET,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('failed computing percentages_')
        RETURN
      END IF
C     *
      RETURN
      END
```

```
      ************************************
      SUBROUTINE REPREC(NF,FLN,SUB,SET,DONE)
      ************************************
C     *
C      -in NF number of fields
C      -in FLN array of fields
C      -in SUB array of substitutions
C      -in SET set search pattern
C      -out DONE done it
C     *
      CHARACTER*15 CHR
      CHARACTER*50 SUB(9)
      INTEGER IDUM,JREC,K,NF,FLN(9),ITILD,ITARG,N
      LOGICAL SET,OK,DONE,TEST(99)
      COMMON/LOGIC/TEST
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSC.COM'
%INCLUDE ':GABLE:V31:DMS:DMSL.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSFF.COM'
C     *
      IREC=1
      CALL CHAROT('replacing data_')
10    CONTINUE
      IF (IREC.GT.NREC) GOTO 5000
      IDUM=IREC
      IF (SET) THEN
       CALL EDTSEA(IDUM,1,JREC,.FALSE.,OK)
       IF (.NOT.OK) GOTO 5000
       IREC=JREC
       IDUM=IREC
      END IF
      READ(IDCHAN,REC=IREC) TEXREC(1:IFLDW)
      DO 100 K=1,NF
        CHR=FLDNAM(FLN(K))
        IF ((CHR(1:14).EQ.'ROOM NUMBER 1_'.OR.CHR(1:14).EQ.
     +          'ROOM NUMBER 2_'))THEN
          DO 200 J=1,NELEM
            IF(.NOT.TEST(J).AND.FFNUM(J).EQ.FLN(K)) THEN
              GOTO 100
            END IF
200       CONTINUE
        END IF
        J1=ISTA(FLN(K))
        J2=IFIN(FLN(K))
C       IF (IFLDTP(FLN(K)).EQ.2) THEN   ! if field integer
C         IDEC=0              ! number after decimal point=0
C       ELSE
C         IDEC=NDECP
C       END IF
        ITILD=INDEX(SUB(K),'_')
        IF (ITILD.EQ.1) GOTO 100
```

```
            IF (ITILD.LT.(J2-J1)) THEN
              TEXREC(J1:J2)=SUB(K)(1:ITILD-1)
              WRITE(IDCHAN,REC=IREC) TEXREC(1:IFLDW)
            END IF
100       CONTINUE
          IREC=IREC+1
          GOTO 10
C         *
5000      CONTINUE
          DONE=.TRUE.
          RETURN
          END
```

```
      ******************
      SUBROUTINE SCSURF
      ******************
C     *
C     search in surface file for materials.
C     *
%INCLUDE':GABLE:V31:DMS:DMSA.COM'
%INCLUDE':GABLE:V31:MOD391:SURF.COM'
C
      CHARACTER*50 FNAM,NAME
C
      CALL SURTRA                    ! trace SURFACE.DAT from SPEC
C
      NAME='SURFACES_'
      CALL FILBLD(JOB,'DMS_',NAME,FNAM)
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('can not trace file_')
        RETURN
      END IF
C
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in DMS file _')
        CALL DMSCLO
        RETURN
      END IF
C
C     'INTEROGATE THE BUILDING MODEL ABOUT THE MATERIALS
C     THAT ENTERS IN THE COMPOSITION OF SURFACES.'
C      ( similar to 391 list and query )
C
      RETURN
      END
```

```
      ***********************************************
      SUBROUTINE SEAPT1(LEVEL,STR1,STR2,TOT,TOTAL,OK)
      ***********************************************
C     *
C     search pattern for LEVEL number and ELEMENT type (STR1)
C     an additional test could be used as a 3rd operator for the
C     INT EXT field by including value in STR2
C     a total of field TOT is calculated and returned by TOTFLD as
C     TOTAL
C     *
C     OK is true if search pattern accepted
C     *
      CHARACTER*15 FIELD(9)
      CHARACTER*50 STRING(9),STR1,STR2,TOT,CHR
      INTEGER OPERAT(9),LEVEL,ERROR,OP
      LOGICAL OK
      REAL*8 TOTAL
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
C     *
      OK=.TRUE.
      DO 100 I=1,9
        STRING(I)='_'
        FIELD(I)='_'
        OPERAT(I)=0
100   CONTINUE
      IF(LEVEL.LT.1.OR.LEVEL.GT.100) THEN
        CALL ERRMES('illegal level number_')
        OK=.FALSE.
        RETURN
      ELSE IF(LEVELN(LEVEL).NE.LEVEL) THEN   !undefined level
        CALL ERRMES('undefined level_')
        OK=.FALSE.
        RETURN
      END IF
      CALL IROUND(LEVEL-1,CHR,ITILD)         !getting level right
      FIELD(1)='LEVEL_'
      STRING(1)(1:ITILD)=CHR(2:ITILD+1)
      OPERAT(1)=1
      FIELD(2)='ELEMENT_'
      STRING(2)=STR1
      OPERAT(2)=1
      IF (STR2(1:9).EQ.'EXTERNAL_'.OR.STR2(1:9).EQ.'INTERNAL_')THEN

        OP=3
        FIELD(3)='INT EXT_'
        STRING(3)=STR2
        OPERAT(3)=1
      ELSE
        OP=2
      END IF
      CALL SEAPAT(OP,1,FIELD,STRING,OPERAT,ERROR)
      IF (ERROR.NE.0) THEN
        CALL ERRMES('error in the search pattern format_')
```

```
300     QUEST='ff1950ffend of range (or press RETURN for'//
     +        ' single level) '
        CALL CHARIN(0,QUEST,0,ANSWER,K,LEV2,R)
        IF(K.EQ.0) THEN      !return pressed for single level
          LEV2=LEV1
        ELSE IF(K.EQ.1.OR.LEV2.LT.0.OR.LEV2.GT.99) THEN
          CALL ERRMES('ff1008ffmust be in range 0-99')
          GOTO 300
        ELSE IF (LEVELN(LEV2+1).NE.LEV2) THEN
          CALL DROUND(DBLE(LEV1),ANSWER,0)
          J = INDEX(ANSWER,'')-1
          PARAMS(1)=ANSWER
          NPARAM=1
          CALL ERRMES('ff712ffLevel %1% does not exist')
          GOTO 300
        ELSE IF(LEV1.GT.LEV2) THEN
          TEMP=LEV1                  !swaping between LEV1 and LEVELN
          LEV1=LEV2
          LEV2=TEMP
        ENDIF
400     CONTINUE
C       ************************************************
        IF (.NOT.OFF) THEN
          LEV1=1
          LEV2=100
        ELSE
          LEV1=LEV1+1
          LEV2=LEV2+1
        END IF
C       ------------------------------------------------
        DO 500 I=LEV1,LEV2
        IDUM=I-1
          IF (LEVELN(I).EQ.IDUM) THEN
            FFQTOT(1)=FFQTOT(1)+GRAREA(I)
            IF (LEVELN(I).EQ.0) THEN             !just for floor 0
             FFQTOT(2)=FFQTOT(2)+GSAREA(I)
C            *-----------------------------------------------
900          CONTINUE
             CALL CHAROT('give siteworks_area')
             CALL CHARIN(0,'default [0] ',0,CHR,K,N,R)
             IF (K.EQ.1) THEN
               CALL ERRMES('reply as requested')
               GOTO 900
             ELSE IF (K.GE.2) THEN
               FFQTOT(21)=R
             END IF
            END IF
            FFQTOT(3)=FFQTOT(3)+LUAREA(I)
C           FFQTOT(4)=FFQTOT(4)+EXSTHT(I)          !not used
C           FFQTOT(5)=FFQTOT(5)+INSTHT(I)          !not used
            FFQTOT(6)=FFQTOT(6)+EXWAAR(I)
            FFQTOT(7)=FFQTOT(7)+INWAAR(I)
            FFQTOT(8)=FFQTOT(8)+EXWIAR(I)
            FFQTOT(9)=FFQTOT(9)+INWIAR(I)
```

```
            FFQTOT(10)=FFQTOT(10)+EXDOAR(I)
            FFQTOT(11)=FFQTOT(11)+INDOAR(I)
            FFQTOT(12)=FFQTOT(12)+ROAREA(I)
            FFQTOT(13)=FFQTOT(13)+CEAREA(I)
C           *
            IF (LEVELN(I).EQ.0) THEN
              FFQTOT(14)=0
            ELSE
              FFQTOT(14)=FFQTOT(14)+GRAREA(I)
            END IF
C           *
            FFQTOT(15)=FFQTOT(15)+STRFLT(I)
            FFQTOT(16)=FFQTOT(16)+STRFLT(I)
            FFQTOT(17)=FFQTOT(17)+PILEHD(I)
            FFQTOT(18)=FFQTOT(18)+SUSCLG(I)
            FFQTOT(19)=FFQTOT(19)+LFTFLT(I)
            FFQTOT(20)=FFQTOT(20)+SANFIT(I)
            NUM=NUM+1
          END IF
500       CONTINUE
C         *
C         *+++++++++++++++++++++++++++++++++++++++++
          IF (LEV1.NE.LEV2) THEN                    ! field LEVEL in BEM
            IDUM=NUM*(-1)
            IF (.NOT.OFF) THEN
              IDUM=IABS(IDUM)
            END IF
          END IF
          CALL BEMPT1
C         *
          RETURN
          END
```

```
      ********************
      SUBROUTINE COMRAT
      ********************
C     *
C     compute rate with quantity of BEQ file
C     *
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:CALMEM.COM'
      CHARACTER*50 CALCHR,STRING(9),STR,CHR,FIELD
      CHARACTER*15 FIELDS(9)
      INTEGER ITARG,OPERAT(9),ERROR,COUNT
      REAL*8 TOTAL
      LOGICAL SET,DONE
C     *
C     set up search pattern for preliminaries exclusion
C     *
      FIELDS(1)='CODE_'
      STRING(1)='(10_'
      OPERAT(1)=2
      CALL SEAPAT(1,1,FIELDS,STRING,OPERAT,ERROR)
      IF (ERROR.NE.0) THEN
        CALL ERRMES('error in the search pattern_')
        RETURN
      END IF
C     *
      ITARG=6                          ! PROJECT RATE
      CALCHR='BASE RATE*PRICE ADJUST._'
      SET=.TRUE.
      NDECP=2
      CALL COMPRC(ITARG,CALCHR,SET,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('failed computing base rate and price adj._')
        RETURN
      END IF
C     *
      ITARG=5                           ! ELEMENTAL COST
      CALCHR='QUANTITY*BASE RATE*PRICE ADJUST._'
      SET=.TRUE.
      NDECP=0
      CALL COMPRC(ITARG,CALCHR,SET,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('failed computing rate and quantity_')
        RETURN
      END IF
C     *
C     work out total of ELEMENTAL COST
C     *
      FIELD='ELEMENTAL COST_'
      CALL TOTFLD(FIELD,SET,TOTAL,COUNT,ERROR)
      IF (ERROR.NE.0) THEN
        CALL ERRMES('failed calculating total of a field_')
        RETURN
```

```
      END IF
      CALAZ(12)=TOTAL/100              ! L=TOTAL
C     *
C     work out ELEMENTAL COST for preliminaries
C     *
      FIELDS(1)='CODE_'
      STRING(1)='(10_'
      OPERAT(1)=1
      CALL SEAPAT(1,1,FIELDS,STRING,OPERAT,ERROR)
      IF (ERROR.NE.0) THEN
        CALL ERRMES('error in search pattern_')
        RETURN
      END IF
      CALCHR='PROJECT RATE*L_'
      SET=.TRUE.
      CALL COMPRC(ITARG,CALCHR,SET,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('failed computing preliminaries_')
        RETURN
      END IF
C     *
      CALL PRCENT                      !work out percentages
C     *
      CALL SETSTA                      !set up status
C     *
      RETURN
      END
```

```
      ******************
      SUBROUTINE COSMOD
      ******************
C     *
C     main host routine for cost model
C     *
      CHARACTER*50 FNAML,FNAMB,FNAMI,FNAMR,FNAMO,FNAMF,FNAMQ
      LOGICAL OK,DONE
C     *
      CALL GET5FL(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
      IF(.NOT.OK) RETURN
      CALL GETOFL(FNAMF,FNAMQ,OK)           !get output filenames
      IF(.NOT.OK) RETURN
      CALL CHAROT('getting level data...')
      CALL FFQPT1(FNAML,OK)
      IF(.NOT.OK) RETURN
      CALL CHAROT('getting layout data...')
      CALL FFQPT2(FNAMB,OK)
      IF(.NOT.OK) RETURN
      CALL CHAROT('getting surface data...')
      CALL FFQPT3(FNAMI,OK)
      IF(.NOT.OK) RETURN
C     CALL CHAROT('getting room data...')
C     CALL FFQPT4(FNAMR,OK)
C     IF(.NOT.OK) RETURN
      CALL CHAROT('getting opening data...')
      CALL FFQPT5(FNAMO,OK)
      IF(.NOT.OK) RETURN
      CALL CREFFQ(FNAMF)
      CALL CHAROT('storing file '//FNAMF)
      CALL ADDFFQ(FNAMF,OK)
      CALL DMSCLO
      CALL CREBEQ(FNAMQ)
      CALL CALTOT
      CALL BEQPT1(FNAMQ,DONE)
      CALL DMSCLO
      RETURN
      END
```

```
      ******************
      SUBROUTINE COSMNR
      ******************
C     *
C     cost modelling main routine
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      CHARACTER*50 MENUT(20),CHR,NAME,FNAM
      REAL*8 RADJ
      INTEGER IREC,N,IFR
      LOGICAL ON,DONE,OK
C     *
150   CONTINUE
C     *
      MENUT(1)='COST MODELLING '
      MENUT(2)='Rate based cost modelling'
      MENUT(3)='Reset cost model to imported rate'
      MENUT(4)='Display Building Elemental Model file'
C     *
      CALL MENU(MENUT,1,3,I)
      GOTO (1000,3000,4000) I
      RETURN
C     *
C     SFQ based cost modelling
C     *
1000  CONTINUE
      ON=.TRUE.
      CALL DISBEM(ON)
C     *
      CALL COMRAT
      DO 1500 I=1,IF9
        IF (LOCFLD(I).NE.2) LOCFLD(I)=0
1500  CONTINUE
      LOCFLD(7)=1
      GOTO 150
C     *
C     FFQ based cost modelling
C     *
2000  CONTINUE
      ON=.TRUE.
      CALL DISBEM(ON)
C     *
      CALL COMRAT
      DO 1600 I=1,IF9
        IF (LOCFLD(I).NE.2) LOCFLD(I)=0
      1600    CONTINUE
      LOCFLD(7)=1
      GOTO 150
C     *
C     reset price adjustement to 1
C     *
3000  CONTINUE
```

```
        NAME='BEM_'
        CALL GETIFL(NAME,FNAM,OK)
        IF (.NOT.OK) RETURN
C       *
C       trace DMS file
C       *
        CALL DMSTRA(FNAM,DONE)
        IF (.NOT.DONE) THEN
          CALL ERRMES('can not trace BEM file_')
          OK=.FALSE.
          RETURN
        END IF
        CALL FINFLD('PRICE ADJUST._',IFR)
        IF (IFR.EQ.0) THEN
          CALL ERRMES('file format error_')
          CALL DMSCLO
          RETURN
        END IF
        RADJ=1D0                                ! resetting rate
        DO 200 I=1,NREC
          IREC=I
          CALL PUTFLD(IREC,IFR,CHR,3,N,RADJ)
200     CONTINUE
C       *
        CALL COMRAT
        GOTO 150
C       *
C       display BEM file
C       *
4000    CONTINUE
        ON=.FALSE.
        CALL DISBEM(ON)
        GOTO 150
C       *
        END
```

```
      ******************
      SUBROUTINE COSTSL
      ******************
C     *
C     cost selection main routine
C     *
      CHARACTER*50 CHR,MENUT(20)
      INTEGER K,N
      REAL*8 R
      LOGICAL OK
C     *
%INCLUDE ':GABLE:V31:COMMON:DVICES.COM'
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
C     *
      OK=.TRUE.
C     *
      CALL DPAGE
C     *
      MENUT(1)='BUILDING SELECTION_'
      MENUT(2)='  - Miscellaneous._'
      MENUT(3)='  - Transport and industrial buildings._'
      MENUT(4)='  - Administration/Commercial buildings._'
      MENUT(5)='  - Health and Welfare._'
      MENUT(6)='  - Refreshment/Entertainment/Recreation._'
      MENUT(7)='  - Religious buildings._'
      MENUT(8)='  - Educational/Cultural/Scientific._'
      MENUT(9)='  - Residential buildings._'
C     *
      CALL MENU(MENUT,0,8,I)
      IF (I.EQ.0) RETURN
      IF (I.LT.1.OR.I.GT.8) THEN
        CALL ERRMES('not a building type_')
        RETURN
      ELSE
       CALL BTSELT(I,OK)
       IF (.NOT.OK) THEN
        CALL ERRMES('no building type selection_')
        RETURN
       END IF
      END IF
C     *
      RETURN
      END
```

```
      ******************
      SUBROUTINE CMVER3
      ******************
C     *
C     second version of COSMOD main host routine
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:COMMON:DVICES.COM'
      %INCLUDE ':GABLE:V31:COMMON:BOARD.COM'
%INCLUDE ':GABLE:V31:COMMON:HLP.COM'
C     *
      CHARACTER*50 FNAML,FNAMB,FNAMI,FNAMR,FNAMO,FNAM,MENUT(20)
+      ,NAME
      LOGICAL OK,DONE,TEST
      REAL*8 LVHT(100)
      INTEGER LEVELN(100),SPANUM(100),LTOT,SEL
C     *
      OK=.TRUE.
C     *
100   CONTINUE
      CALL DPAGE
      MENUT(1)='COST MODELLING'
      MENUT(2)='Create Space Fundamental Quantities'
      MENUT(3)='Create Floor Fundamental Quantities'
      MENUT(4)='CISFB Building Elemental Model'
      MENUT(5)='Cost Data Selection'
      MENUT(6)='Cost Modeller'
      CALL MENU(MENUT,1,5,I)
      HLPLVL=0
      GOTO (1000,2000,3000,4000,5000)I
      RETURN
C     *
C     creating SFQ*****************************************
C     *
1000  CONTINUE
      CALL SFQMNR(OK)
      IF(.NOT.OK) RETURN
      GOTO 100
C     *
C     creating FFQ*****************************************
C     *
2000  CONTINUE
      CALL FFQMNR(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
      IF (.NOT.OK) RETURN
      GOTO 100
C     *
C     CISFB building elemental quantities******************
C     *
3000  CONTINUE
C
      CALL BEMMNR(OK)
      IF (.NOT.OK) RETURN
      GOTO 100
C     *
```

```
C       cost data selection*********************************
C
4000    CONTINUE
        CALL COSTSL
        GOTO 100
C       *
C       cost modeller***************************************
C       *
5000    CONTINUE
C       NAME='COST_MOD_'
C       CALL JOBCHK(NAME,OK)
C       IF (.NOT.OK) RETURN
        CALL COSMNR
        GOTO 100
C       *
        END
```

```
      **********************
      SUBROUTINE COMFFQ(SET)
      **********************
C     *
C     complementary floor fundamental quantities typed in
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
C     *
C    24-FEB-88 REV_1.16  TAB canceled suspended ceiling area input
C     *
      CHARACTER*50 CHR,STR
      REAL*8 R
      INTEGER IDUM,I,N,J
      LOGICAL LEVSEL(100),SPASEL(2000),SET
      COMMON/SELECT/LEVSEL,SPASEL
C     *
      IF (.NOT.SET) THEN
       DO 110 I=1,100
          LEVSEL(I)=.TRUE.
110    CONTINUE
       DO 120 I=1,2000
          SPASEL(I)=.TRUE.          !initialising selected spaces
120    CONTINUE
      END IF
C     *
      DO 100 I=1,100
             IDUM=I-1
200   CONTINUE
      IF (LEVELN(I).EQ.I) THEN
       IF (LEVSEL(LEVELN(I))) THEN
C       *-------------------------------------------------
        CALL IROUND(IDUM,STR,J)
        CHR='complementary FFQs for floor'//STR
        CALL CHAROT(CHR)
C       *-------------------------------------------------
C       CALL CHAROT('give number of_piles in building foundations_')
C       CALL CHARIN(0,'default [0] _',0,CHR,K,N,R)
C       IF (K.EQ.1) THEN
C         CALL ERRMES('reply as requested_')
C         GOTO 200
C       ELSE IF (K.GE.2) THEN
C         PILEHD(I)=N
C       END IF
C       *-------------------------------------------------
        CALL CHAROT('give number of_staircases in building_')
        CALL CHARIN(0,'default [0] _',0,CHR,K,N,R)
        IF (K.EQ.1) THEN
          CALL ERRMES('reply as requested_')
          GOTO 200
        ELSE IF (K.GE.2) THEN
          STRFLT(I)=N
        END IF
C       *-------------------------------------------------
        CALL CHAROT('give number of lift flights _')
```

```
            CALL CHARIN(0,'default [0] ¯',0,CHR,K,N,R)
            IF (K.EQ.1) THEN
              CALL ERRMES('reply as requested¯')
              GOTO 200
            ELSE IF (K.GE.2) THEN
              LFTFLT(I)=N
            END IF
C           *-----------------------------------------------
            CALL CHAROT('give sanitary fittings number¯')
            CALL CHARIN(0,'default [0] ¯',0,CHR,K,N,R)
            IF (K.EQ.1) THEN
              CALL ERRMES('reply as requested¯')
              GOTO 200
            ELSE IF (K.GE.2) THEN
              SANFIT(I)=N
            END IF
C           *-----------------------------------------------
          END IF
        END IF
100     CONTINUE
C       *
        RETURN
        END
```

```
      ************************
      SUBROUTINE CREFFQ(FNAMF)
      ************************
C     *
C     creates dms file FNAMF for floor fundamental quantities
C     *
C     *
      CHARACTER*50 FNAMF,FNAMI,FNAME
      LOGICAL EXIST
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      FNAME=FNAMF
      CALL DBSINI
      CALL STAFIN
      DBSNAM=FNAME
      ITILD=INDEX(FNAME,'_')
      FNAMI=FNAME(1:ITILD-1)//'.IND_'
      CALL FILDEL(DBSNAM,EXIST)
      CALL FILDEL(FNAMI,EXIST)
C     *
      FLDNAM(1)='CODE_'
      IFLDTP(1)=1
      FLDNAM(2)='LEVEL_'
      IFLDTP(2)=2
      FLDNAM(3)='GROSS AREA_'
      IFLDTP(3)=3
      FLDNAM(4)='GRND SLAB AREA_'
      IFLDTP(4)=3
      FLDNAM(5)='LET USE AREA_'
      IFLDTP(5)=3
      FLDNAM(6)='EXT STOREY HT_'
      IFLDTP(6)=3
      FLDNAM(7)='INT STOREY HT_'
      IFLDTP(7)=3
      FLDNAM(8)='EXT WALL AREA_'
      IFLDTP(8)=3
      FLDNAM(9)='INT WALL AREA_'
      IFLDTP(9)=3
      FLDNAM(10)='EXT WIND AREA_'
      IFLDTP(10)=3
      FLDNAM(11)='INT WIND AREA_'
      IFLDTP(11)=3
      FLDNAM(12)='EXT DOOR AREA_'
      IFLDTP(12)=3
      FLDNAM(13)='INT DOOR AREA_'
      IFLDTP(13)=3
      FLDNAM(14)='ROOF AREA_'
      IFLDTP(14)=3
      FLDNAM(15)='CEILING AREA_'
      IFLDTP(15)=3
C     FLDNAM(16)='PILE HEADS_'
```

```
C       IFLDTP(16)=2
        FLDNAM(16)='STRCASE FLIGHT_'
        IFLDTP(16)=2
        FLDNAM(17)='LIFT FLIGHTS_'
        IFLDTP(17)=2
        FLDNAM(18)='SUSPENDED CEAR_'
        IFLDTP(18)=3
        FLDNAM(19)='SANITARY FITNS_'
        IFLDTP(19)=2
        FLDNAM(20)='NO OF SPACES_'
        IFLDTP(20)=2
        IF9=20
        CALL STAFIN
        NREC=0
        CALL FILOPY(DBSNAM,0,IFLDW,IDCHAN)
        CALL DMSSTR
        RETURN
        END
```

```
      ************************
      SUBROUTINE CREBEM(FNAMQ)
      ************************
C     *
C     creates DMS file FNAMQ for building elemental quantities
C     *
      CHARACTER*50 FNAMQ,FNAMI,FNAME
      LOGICAL EXIST
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      FNAME=FNAMQ
      CALL DBSINI
      CALL STAFIN
      DBSNAM=FNAME
      ITILD=INDEX(FNAME,'_')
      FNAMI=FNAME(1:ITILD-1)//'.IND_'
      CALL FILDEL(DBSNAM,EXIST)
      CALL FILDEL(FNAMI,EXIST)
C     *
      FLDNAM(1)='CODE_'
      IFLDTP(1)=1
      FLDNAM(2)='ELEMENT_'
      IFLDTP(2)=-1
      FLDNAM(3)='UNIT_'
      IFLDTP(3)=1
      FLDNAM(4)='QUANTITY_'
      IFLDTP(4)=3
      FLDNAM(5)='ELEMENTAL COST_'
      IFLDTP(5)=3
      FLDNAM(6)='PROJECT RATE_'
      IFLDTP(6)=3
      FLDNAM(7)='PRICE ADJUST._'
      IFLDTP(7)=3
      FLDNAM(8)='BASE RATE_'
      IFLDTP(8)=3
      FLDNAM(9)='LEVEL_'
      IFLDTP(9)=2
      FLDNAM(10)='PERCENT_'
      IFLDTP(10)=3
      FLDNAM(11)='S//T_'                    ! STATUS
      IFLDTP(11)=1
      IF9=11
      CALL STAFIN
      NREC=0
      CALL FILOPY(DBSNAM,0,IFLDW,IDCHAN)
      CALL DMSSTR
RETURN
END
```

```
      ************************
      SUBROUTINE CRESFQ(FNAMF)
      ************************
C     *
C     creates dms file FNAMF for Sloor Fundamental Quantities
C     *
C    22-FEB-88 REV_1.16 TAB added suspended ceiling area
C     *
      CHARACTER*50 FNAMF,FNAMI,FNAME
      LOGICAL EXIST
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      FNAME=FNAMF
      CALL DBSINI
      CALL STAFIN
      DBSNAM=FNAME
      ITILD=INDEX(FNAME,'_')
      FNAMI=FNAME(1:ITILD-1)//'.IND_'
      CALL FILDEL(DBSNAM,EXIST)
      CALL FILDEL(FNAMI,EXIST)
C     *
      FLDNAM(1)='CODE_'
      IFLDTP(1)=1
      FLDNAM(2)='LEVEL_'
      IFLDTP(2)=2
      FLDNAM(3)='GROSS AREA_'
      IFLDTP(3)=3
      FLDNAM(4)='SPACE TITLE_'
      IFLDTP(4)=-1
      FLDNAM(5)='SPACE NUMBER_'
      IFLDTP(5)=2
      FLDNAM(6)='SPACE TYPE_'
      IFLDTP(6)=2
      FLDNAM(7)='EXT WALL BEDS_'
      IFLDTP(7)=3
      FLDNAM(8)='INT WALL BEDS_'
      IFLDTP(8)=3
      FLDNAM(9)='EXT STOREY HT_'
      IFLDTP(9)=3
      FLDNAM(10)='INT STOREY HT_'
      IFLDTP(10)=3
      FLDNAM(11)='EXT WALL AREA_'
      IFLDTP(11)=3
      FLDNAM(12)='INT WALL AREA_'
      IFLDTP(12)=3
      FLDNAM(13)='EXT WIND AREA_'
      IFLDTP(13)=3
      FLDNAM(14)='INT WIND AREA_'
      IFLDTP(14)=3
      FLDNAM(15)='EXT DOOR AREA_'
      IFLDTP(15)=3
```

```
FLDNAM(16)='INT DOOR AREA '
IFLDTP(16)=3
FLDNAM(17)='CEILING AREA '
IFLDTP(17)=3
FLDNAM(18)='SUSP. CEILING '
IFLDTP(18)=3
FLDNAM(19)='GR. SLAB AREA '
IFLDTP(19)=3
FLDNAM(20)='ROOF AREA '
IFLDTP(20)=3
IF9=20
CALL STAFIN
NREC=0
CALL FILOPY(DBSNAM,0,IFLDW,IDCHAN)
CALL DMSSTR
RETURN
END
```

```
      ************************
      SUBROUTINE DISBEM(ON)
      ************************
C     *
C     display Building Elemental Quantities file, using a
C     report ont the selected type.
C     *
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSC.COM'
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:COMMON:XXOUT.COM'
C     *
      INTEGER SEL,J,ERROR,LINES,I,K,N,OP,DIREC,IDUM,ELEM
      REAL*8 R
      CHARACTER*50 NAME,RNAME,STR,FNAM
      CHARACTER*80 CHR
      LOGICAL OK,DONE,SET,MESSAGE,ON
C     *
      IROUTE=1
      OK=.TRUE.
C     *
      CALL DPAGE
      CALL CHAROT('BUILDING ELEMENTAL MODEL ')
      CALL CHAROT(' ')
C     *
C     get BEM file name
C     *
      NAME='BEM'
      CALL GETIFL(NAME,FNAM,OK)
      IF (.NOT.OK) RETURN
C     *
C     trace DMS file
C     *
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('can not trace BEM file')
        OK=.FALSE.
        RETURN
      END IF
      IF (NRES.LT.1) THEN
        CALL ERRMES('no levels in LEVEL file')
        CALL DMSCLO
        OK=.FALSE.
        RETURN
      END IF
      IF (.NOT.ON) THEN
        CALL CHAROT(' ')
        CALL CHAROT('specify output device for report')
        CALL GTROUT(DONE)
        IF(.NOT.DONE) RETURN
      END IF
```

```
            OK=.FALSE.
            RETURN
          END IF
C         *
          CALL TOTFLD(TOT,.TRUE.,TOTAL,N,ERROR)
C         *
          IF (ERROR.NE.0) THEN
            CALL ERRMES('can not calculate total of '//TOT)
            CALL CHAROT(DBSNAM)
            OK=.FALSE.
          END IF
          RETURN
          END
```

```
      ******************************
      SUBROUTINE SEAPT2(SSEL,NUM,OK)
      ******************************
C     *
C     search pattern for wall spec replacement in FNAMIW file
C     *
      CHARACTER*15 FIELD(100)
      CHARACTER*50 STRING(100),STR
      INTEGER OPERAT(100),ERROR,SSEL(100),NUM,JTILD
      LOGICAL OK
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      WRITE(*,*) 'SEAPT2 : ---------SEAPT2-------'
      OK=.TRUE.
      DO 100 I=1,100
        STRING(I)='_'
        FIELD(I)='_'
        OPERAT(I)=0
100   CONTINUE
C     *
      FIELD(1)='LEVEL_'                          ! LEVEL field
      CALL IROUND(SSEL(1),STR,J)
      STRING(1)=STR
      OPERAT(1)=1
      FIELD(2)='ELEMENT_'                        ! INTERNAL WALL
      STRING(2)='INTERNAL WALL_'
      OPERAT(2)=1
      FIELD(3)='ELEMENT_'                        ! EXTERNAL WALL
      STRING(3)='EXTERNAL WALL_'
      OPERAT(3)=1
C     *
      DO 2000 K=1,NUM
        KDUM=K+3
        FIELD(KDUM)='ROOM_'
        CALL IROUND(SSEL(KDUM),STR,JTILD)
        STRING(KDUM)(1:JTILD)=STR(2:JTILD+1)
        OPERAT(KDUM)=1
C
2000  CONTINUE
      DO 90 J=1,KDUM
        WRITE(*,*) 'FIELD(',J,')= ',FIELD(J),'   STRING=',STRING(J)

90    CONTINUE
C     *
C     IF (KDUM.EQ.2) THEN
C        CALL SEAPAT(KDUM,1,FIELD,STRING,OPERAT,ERROR)
C     ELSE
C        CALL SEAPAT(KDUM,7,FIELD,STRING,OPERAT,ERROR)
C     END IF
C     *
      IF (ERROR.NE.0) THEN
        CALL ERRMES('error in the search pattern format_')
        OK=.FALSE.
        RETURN
```

```
      END IF
C     *
      RETURN
      END
```

```
      ******************************
      SUBROUTINE SEAPT3(SSEL,NUM,OK)
      ******************************
C     *
C     search pattern for wall spec replacement in FNAMIW file
C     *
      CHARACTER*15 FIELD(100)
      CHARACTER*50 STRING(100),STR
      INTEGER OPERAT(100),ERROR,SSEL(100),NUM,JTILD
      LOGICAL OK
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      WRITE(*,*) 'SEAPT3 : ---------SEAPT3-------'
      OK=.TRUE.
      DO 100 I=1,100
        STRING(I)='_'
        FIELD(I)='_'
        OPERAT(I)=0
100   CONTINUE
C     *
      FIELD(1)='LEVEL_'                          ! LEVEL field
      CALL IROUND(SSEL(1),STR,J)
      STRING(1)=STR
      OPERAT(1)=1
      FIELD(2)='ELEMENT_'                        ! INTERNAL WALL
      STRING(2)='INTERNAL WALL_'
      OPERAT(2)=1
      FIELD(3)='ELEMENT_'                        ! EXTERNAL WALL
      STRING(3)='EXTERNAL WALL_'
      OPERAT(3)=1
C     *
      DO 2000 K=1,NUM
        KDUM=K+3
        FIELD(2*KDUM-4)='ROOM NUMBER 1_'
        CALL IROUND(SSEL(KDUM),STR,JTILD)
        STRING(2*KDUM-4)(1:JTILD)=STR(2:JTILD+1)
        OPERAT(2*KDUM-4)=1
C
        FIELD(2*KDUM-3)='ROOM NUMBER 2_'
        CALL IROUND(SSEL(KDUM),STR,JTILD)
        STRING(2*KDUM-3)(1:JTILD)=STR(2:JTILD+1)
        OPERAT(2*KDUM-3)=1
2000  CONTINUE
      KDUM=2*KDUM-3
      DO 90 J=1,KDUM
        WRITE(*,*) 'FIELD(',J,')= ',FIELD(J),'   STRING=',STRING(J)

90    CONTINUE
C     *
C     IF (KDUM.EQ.2) THEN
C        CALL SEAPAT(KDUM,1,FIELD,STRING,OPERAT,ERROR)
C     ELSE
        CALL SEAPAT(KDUM,7,FIELD,STRING,OPERAT,ERROR)
C     END IF
```

```
C       *
        IF (ERROR.NE.0) THEN
          CALL ERRMES('error in the search pattern format')
          OK=.FALSE.
          RETURN
        END IF
C       *
        RETURN
        END
```

```
          ************************
          SUBROUTINE SEARPT(SEL,OK)
          ************************
C         *
C         search set up for report on BEQ file
C         *
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSC.COM'
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
C         *
          INTEGER SEL,JTILD,OP,ERROR,OPERAT(9)
          CHARACTER*15 FIELDS(9)
          CHARACTER*50 STR,CHR,STRING(9)
          LOGICAL OK
C         *
          CALL IROUND(SEL,STR,J)
          CHR='('//STR(2:)
          JTILD=INDEX(CHR,'_')
          FIELDS(1)='CODE_'
          STRING(1)=CHR(1:JTILD)
          OPERAT(1)=1
          FIELDS(2)='ELEMENT_'
          STRING(2)='NOT USED_'
          OPERAT(2)=2
          IF (SEL.EQ.1) THEN
            OP=3
            FIELDS(3)='CODE_'
            STRING(3)='(10_'
            OPERAT(3)=2
          ELSE
            OP=2
          END IF
          CALL SEAPAT(OP,1,FIELDS,STRING,OPERAT,ERROR)
          IF (ERROR.NE.0) THEN
            CALL ERRMES('error in the search pattern_')
            RETURN
          END IF
C         *
          RETURN
          END
```

```
      **************************
      SUBROUTINE SELELM(SEL,OK)
      **************************
C     *
C     select element on BEM file
C     *
      CHARACTER*50 MENUT(20)
      INTEGER SEL,I
      LOGICAL OK
C     *
      MENUT(1)=' SELECT BUILDING ELEMENTAL MODEL '
      MENUT(2)=' Substructures '
      MENUT(3)=' Structure '
      MENUT(4)=' Completion '
      MENUT(5)=' Finishes '
      MENUT(6)=' Services '
      MENUT(7)=' Installation '
      MENUT(8)=' Fixed fitting '
      MENUT(9)=' Loose fittings '
      MENUT(10)=' Siteworks '
      MENUT(11)=' Preliminaries '
C     *
      CALL MENU(MENUT,1,10,I)
      SEL=I
      IF (I.EQ.0) SEL=-1
      OK=.TRUE.
      CALL DPAGE
C
      RETURN
      END
```

```
      ********************************************
      SUBROUTINE SFQCPY(FNAMI,FNAMB,FNAMIW,FNAMBW,OK)
      ********************************************
C     *
C     copy's ISAAC and LAYOUT files in working DMS files
C       - in FNAMI full path name of ISAAC dms file
C       - in FNAMB full path name of LAYOUT dms file
C       - out FNAMIW full path name of ISAAC file copy
C       - out FNAMBW full path name of LAYOUT file copy
C       - out OK
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      CHARACTER*50 FNAMI,FNAMB,FNAMIW,FNAMBW
      INTEGER ERROR
      LOGICAL DONE,OK,SET
C     *--------------------------------------
C     copy ISSAC file in FNAMIW working file ---------
C     *--------------------------------------
      CALL DMSTRA(FNAMI,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in ISAAC file_')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     *--------------------------------------------
C     copy file FNAMI in a work file  called FNAMIW
C     *--------------------------------------------
      CALL FILBLD(JOB,'DMS_','FNAMIW_',FNAMIW)
      CALL CHAROT('making a copy of the ISAAC dms file_')
      CALL CPYTRA(FNAMI,FNAMIW,ERROR)
C     *
      CALL DMSCLO
C     *--------------------------------------
C     copy LAYOUT file in FNAMBW working file ------------
C     *--------------------------------------
      CALL DMSTRA(FNAMB,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in LAYOUT file_')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     *--------------------------------------------
C     copy file FNAMB in a work file called FNAMBW
C     *--------------------------------------------
      CALL FILBLD(JOB,'DMS_','FNAMBW_',FNAMBW)
      CALL CHAROT('making a copy of the LAYOUT dms file_')
      CALL CPYTRA(FNAMB,FNAMBW,ERROR)
C     *
```

```
      CALL DMSCLO
C     *
      RETURN
      END
```

```
      ************************************
      SUBROUTINE SFQCRF(FNAMIW,FNAMBW,OK)
      ************************************
C     *
C     copy's ISAAC and LAYOUT files in working DMS files
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      CHARACTER*50 FNAMI,FNAMB,FNAMIW,FNAMBW,FILE
      INTEGER ERROR
      LOGICAL DONE,OK,SET
C     *---------------------------------------
C     copy ISAAC file in FNAMIW working file ------------
C     *---------------------------------------
      CALL DMSTRA(FNAMIW,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in ISAAC file_')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     *-----------------------------------------------
C     cross reference thickness from DMS SURFACES file
C     *-----------------------------------------------
      CALL FILBLD(JOB,'DMS_','SURFACES_',FILE)
      CALL CHAROT('cross reference in process_')
      CALL CRSREF('THICKNESS_','SPEC_',FILE,'THICKNESS_','SPEC_',
     1            .FALSE.,ERROR)
      IF (ERROR.NE.0) THEN
         CALL DMSCLO
         OK=.FALSE.
         RETURN
      END IF
C     *-------------------------------------------------
C     compute fields GIRTH*THICKNESS and store in WALL BED
C     *-------------------------------------------------
      SET=.FALSE.
      CALL CHAROT('computation in process_')
      CALL COMPUT('WALL BED_','GIRTH*(THICKNESS/1000)_',SET,ERROR)
      IF (ERROR.NE.0) THEN
        CALL DMSCLO
        OK=.FALSE.
        RETURN
      END IF
C     *
      CALL DMSCLO
C     *---------------------------------------
C     copy LAYOUT file in FNAMBW working file ------------
C     *---------------------------------------
      CALL DMSTRA(FNAMBW,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
```

```
            CALL ERRMES('no element in LAYOUT file')
            OK=.FALSE.
            CALL DMSCLO
            RETURN
          END IF
C         *-----------------------------------------------
C         cross reference thicknesses from dms SURFACE file
C         build first its full path name
C         *-----------------------------------------------
          CALL FILBLD(JOB,'DMS','SURFACES',FILE)
          CALL CRSREF('THICKNESS','SPEC',FILE,'THICKNESS','SPEC',
     1                .FALSE.,ERROR)
          IF(ERROR.NE.0) THEN
            CALL DMSCLO
            OK=.FALSE.
            RETURN
          END IF
C         *-----------------------------------------------------
C         compute fields GIRTH * THICKNESS and store in WALL BED
C         *-----------------------------------------------------
          SET=.FALSE.
          CALL COMPUT('WALL BED','GIRTH*(THICKNESS/1000)',SET,ERROR)
          IF (ERROR.NE.0) THEN
            CALL DMSCLO
            OK=.FALSE.
            RETURN
          END IF
C         *
          CALL DMSCLO
C         *
          RETURN
          END
```

```
        **********************
        SUBROUTINE SFQMNR(OK)
        **********************
C       *
C       SFQ main routine with LEVEL file opened for LEVELN,DSTND,DATUM
C       *
        CHARACTER*80 MENUT(10)
        CHARACTER*50 FNAM,FNAMO,FNAML,FNAMI,FNAMB,FNAMR,FNAMIW,FNAMBW
        REAL*8 LVHT(0:99),DST(0:99)
        INTEGER LEVELN(0:99),SPANUM(0:100),LTOT,SEL
        LOGICAL OK,DONE,SET
C       *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
C       *
        OK=.TRUE.
C       *
C       selecte FFQ or SFQ
C       *
150     CONTINUE
        MENUT(1)='SPACE FUNDAMENTAL QUANTITIES'
        MENUT(2)='Create an SFQ file for the all building'
C       MENUT(3)='Create an SFQ file for a selected number of spaces'
C       *
        CALL MENU(MENUT,1,1,I)
        GOTO (1000) I
        RETURN
C       *
 1000   CONTINUE
C       *
C       creating SFQ****************************************
C       *
        CALL GET5FL(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
        IF (.NOT.OK) GOTO 150
        CALL SFQPTO(FNAML,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
        IF (.NOT.OK) GOTO 150
        CALL SFQCPY(FNAMI,FNAMB,FNAMIW,FNAMBW,OK)          !copy files
        IF (.NOT.OK) GOTO 150
        CALL SFQCRF(FNAMIW,FNAMBW,OK)          !cross-reference files
        IF (.NOT.OK) GOTO 150
        CALL SFQ124(FNAMR,FNAMO,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
        GOTO 150
C       *
 2000   CONTINUE
C       *
C       using SFQ file
C       *
C       CALL GET5FL(FNAML,FNAMB,FNAMR,FNAMI,FNAMO,OK)
C       IF (.NOT.OK) GOTO 150
C       CALL SFQCPY(FNAMI,FNAMB,FNAMIW,FNAMBW,OK) !cross-reference files

C       IF (.NOT.OK) GOTO 150
C       CALL SPASEL(FNAML,FNAMR,OK)
```

```
C       IF (.NOT.OK) GOTO 150
C       CALL SPASEL(FNAML,FNAMR,OK)
C       IF (.NOT.OK) GOTO 150
C       CALL SFQCRF(FNAMIW,FNAMBW,OK)
C       IF (.NOT.OK) GOTO 150
C       CALL SFQPT0(FNAML,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
C       IF (.NOT.OK) GOTO 150
C       CALL SFQ124(FNAMR,FNAMO,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
C       GOTO 150
C       *
        END
```

```
      ************************************************************
      SUBROUTINE SFQ124(FNAMR,FNAMO,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
      ************************************************************
      IMPLICIT NONE
C     *
C     SFQ main routine with LEVEL file opened for LEVELN,DSTND,DATUM
C     *
      CHARACTER*50 FNAM,FNAMR,FNAMIW,CHR,STR,NAME,FILE,FNAMO,FNAMBW
      LOGICAL OK,DONE,SET
      REAL*8 R,LVHT(0:99),DST(0:99)
      INTEGER I,J,N,IFL,IFR,IFG,IFD,IFS,IDUM,LTOT,K,LNUM,NSPA
     1  ,LEVELN(0:99),SPANUM(0:99),ERROR,L
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      OK=.TRUE.
C     *
      NAME='SFQ_'
      CALL GETOFL(NAME,FNAM,OK)
      IF (.NOT.OK) RETURN
      CALL CRESFQ(FNAM)
C     *----------------------------------------
C     SFQ main loop
C     *
      DO 700 J=0,LTOT+1              !main loop for SFQ
      IF (LEVELN(J).NE.999) THEN
C     ----------------------------------------
       DO 100 I=1,2000            ! spaces initialisation loop
        SPATIT(I)='_'
        SPATYP(I)=0
        ROONUM(I)=0
        GAREAS(I)=0D0
        EXWABD(I)=0D0
        INWABD(I)=0D0
        ESTHTS(I)=0D0
        ISTHTS(I)=0D0
        EWAARS(I)=0D0
        IWAARS(I)=0D0
        EWIARS(I)=0D0
        IWIARS(I)=0D0
        EDOARS(I)=0D0
        IDOARS(I)=0D0
        ROOFAR(I)=0D0
        CEILAR(I)=0D0
        GSLABA(I)=0D0
        SUCEAR(I)=0D0
100     CONTINUE
C     ----------------------------------------
      LNUM=LEVELN(J)
      NSPA=SPANUM(J)
C      ----------------------------------------
C      getting SFQ common block values
```

```
C       ------------------------------------------------
        CALL IROUND(LNUM,STR,L)
        CHR=' SFQ''s for level '//STR(2:L+1)
        CALL CHAROT(CHR)
        CALL SFQPT1(FNAMR,LNUM,NSPA,OK)              ! using ROOM
        IF (.NOT.OK) RETURN
        CALL SFQPT3(FNAMBW,LNUM,NSPA,LTOT,DST,OK) ! using LAYOUT
        IF (.NOT.OK) RETURN
        CALL SFQPT2(FNAMIW,LNUM,NSPA,LTOT,LVHT,OK)  ! using ISAAC
        IF (.NOT.OK) RETURN
        CALL SFQPT4(FNAMO,LNUM,NSPA,OK)              ! using OPENING
        IF (.NOT.OK) RETURN
        CALL ADDSFQ(FNAM,LNUM,NSPA,OK)
        IF (.NOT.OK) THEN
          CALL ERRMES ('failed in storring SFQs')
          CALL DMSCLO
          RETURN
        END IF
       END IF
700    CONTINUE
C      *
       CALL DMSDEL(FNAMIW,OK)          !delete temp file ( ISAAC copy )
       IF (.NOT.OK) RETURN
       CALL DMSDEL(FNAMBW,OK)          !delete temp file ( LAYOUT copy )
       IF (.NOT.OK) RETURN
C      *
       CHR='file SFQ type stored '//FNAM
       CALL CHAROT(CHR)
       CALL CRWAIT                                  !wait for new line
       RETURN
       END
```

```
      ********************************************************
      SUBROUTINE SFQPT0(FNAML,LEVELN,SPANUM,LVHT,LTOT,DST,OK)
      ********************************************************
      IMPLICIT NONE
C     *
C     start routine with LEVEL and ISAAC files
C        - in FNAML full path name of LEVEL file
C        - out LEVELN number of levels in building
C        - out SPANUM number of spaces in each floor
C        - out LVHT hight floor to ceiling for each floor
C        - out LTOT total number of levels in building
C        - out DST downstand
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
C     24-FEB-88 REV_1.16 TAB set LVHT to -1 fot single floor
C     *
      CHARACTER*50 FNAM,FNAML,CHR,STR,NAME,FILE,TXT
      LOGICAL OK,DONE,SET
      REAL*8 R,LVHT(0:99),DST(0:99)
      INTEGER I,J,N,IFL,IFR,IFG,IFD,IFS,IDUM,TOPLEV,LTOT,K,
     1  LNUM,NSPA,LEVELN(0:99),SPANUM(0:99),ERROR,BOTLEV
C     *
      OK=.TRUE.
C     *
C     *--------------------------
C     open the DMS LEVEL file
C     *--------------------------
      CALL DMSTRA(FNAML,DONE)
      IF(.NOT.DONE) RETURN
      IF(NREC.LT.1) THEN
        CALL ERRMES('no levels in LEVEL file')
        CALL DMSCLO
        RETURN
      END IF
C     *
      CALL FINFLD('LEVEL',IFL)
      CALL FINFLD('NO OF ROOMS',IFR)
      CALL FINFLD('DATUM',IFD)
      CALL FINFLD('DOWNSTAND',IFS)
      IF(IFL.EQ.0.OR.IFR.EQ.0.OR.IFD.EQ.0.OR.IFS.EQ.0)THEN
        CALL ERRMES('LEVEL file format error')
        CALL DMSCLO
        RETURN
      END IF
C     *
      DO 400 I=0,99
       LEVELN(I)=999                          ! arbitrary value
       SPANUM(I)=0
       LVHT(I)=0D0
       DST(I)=0D0
400   CONTINUE
C     *
```

```
      DO 200 I=1,NREC
        IDUM=I
        CALL GETFLD(IDUM,IFL,CHR,K,N,R)
        IF(N.GE.0.AND.N.LE.99) THEN
          LEVELN(N)=N                          ! valid level No
          CALL GETFLD(IDUM,IFD,CHR,K,J,R)
          LVHT(N)=R                            ! datums
          CALL GETFLD(IDUM,IFR,CHR,K,J,R)
          SPANUM(N)=J                          ! number of rooms/floor
          CALL GETFLD(IDUM,IFS,CHR,K,J,R)
          DST(N)=R                            ! downstand
        END IF
200   CONTINUE
      LTOT=NREC                               ! total number of levels
C     ------------------------------------------
C     get right content of LVHT from datums
C     ------------------------------------------
      BOTLEV=0
      TOPLEV=0
      DO 300 I=0,99
      IF (LEVELN(I).EQ.I) THEN
        TOPLEV=I
C       IF (BOTLEV.EQ.0) THEN
C        BOTLEV=I
C       END IF
      END IF
300   CONTINUE
      IF (TOPLEV.GT.BOTLEV) THEN
        DO 500 I=BOTLEV,TOPLEV-1
          LVHT(I)=LVHT(I+1)-LVHT(I)     !getting LVHT right
500     CONTINUE
        LVHT(TOPLEV)=LVHT(TOPLEV-1)  !in stade of INSTHT(TOPLEV)
      ELSE
        LVHT(TOPLEV)=-1              !just one floor level
      END IF
C     *
      RETURN
      END
```

```
      ************************************
      SUBROUTINE SFQPT1(FNAMR,LNUM,NSPA,OK)
      ************************************
      IMPLICIT NONE
C     *
C     open dms ROOM file and gets data for SFQ common
C     *
      CHARACTER*50 FNAMR,CHR
      REAL*8 R
      LOGICAL OK,DONE
      INTEGER ITILD,J,IDUM,NUROOM,IFN,IFR,IFT,IFG,IFL,IFV,LNUM,
     + NSPA,K,N,PT,JDUM
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      OK=.TRUE.
C     *
C     open dms ROOM file
C     *
      CALL DMSTRA(FNAMR,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in ROOM file')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     *
C     ----------------------------------------
C     NOW going through all spaces of floor LDUM
C     ----------------------------------------
      CALL FINFLD('LEVEL',IFL)
      CALL FINFLD('ROOM_NAME',IFN)
      CALL FINFLD('ROOM',IFR)
      CALL FINFLD('ROOM TYPE',IFT)
      CALL FINFLD('GROSS AREA',IFG)
      CALL FINFLD('EST VOLUME',IFV)
      IF(IFN.EQ.0.OR.IFR.EQ.0.OR.IFT.EQ.0.OR.IFG.EQ.0) THEN
         CALL ERRMES('ROOM file format error')
         CALL DMSCLO
         OK=.FALSE.
         RETURN
      END IF
C     *----------------------------------------
C     getting ROOM file data
C     *----------------------------------------
      IDUM=1
      DO 100 J=1,NREC
      JDUM=J
      CALL GETFLD(JDUM,IFL,CHR,K,N,R)
      IF (N.EQ.LNUM) THEN
C        *----------------------------------------
         CALL GETFLD(JDUM,IFR,CHR,K,N,R)          ! room
```

```
            IF (LNUM.GT.0) THEN
              PT=N-(LNUM*10000)
            ELSE
              PT=N
            END IF
            ROONUM(PT)=PT
C           *---------------------------------------
            CALL GETFLD(JDUM,IFN,CHR,K,N,R)
            SPATIT(PT)=CHR                              ! room name
            CALL GETFLD(JDUM,IFT,CHR,K,N,R)
            SPATYP(PT)=N                                ! room type
            CALL GETFLD(JDUM,IFG,CHR,K,N,R)
            GAREAS(PT)=R                                ! gross area
            CALL GETFLD(JDUM,IFV,CHR,K,N,R)
            ISTHTS(PT)=R/GAREAS(PT)*1000D0               ! int storey ht

C           *---------------------------------------
          END IF
          IDUM=IDUM+1
100       CONTINUE
          CALL DMSCLO
          RETURN
          END
```

```
      ************************************************
      SUBROUTINE SFQPT2(FNAMIW,LNUM,NSPA,LTOT,LVHT,OK)
      ************************************************
      IMPLICIT NONE
C     *
C     opens dms ISAAC file and extracts data for SFQ common
C     FNAMIW - file name of ISAAC + Thickness
C     LNUM - number of levels in building
C     NSPA - number of spaces in floor level
C     LTOT - total number of levels
C     LVHT - datum to datum height per floor exept top-level
C     OK - if .true. continue
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
C    10-FEB-88 REV_1.16 TAB max ISTHTS for top level default height
C     *
      CHARACTER*50 FNAMI,FNAMIW,STR,FILE,TXT,DSTR,CHR
      REAL*8 R,LVHT(0:99),HMAX,SPALEV
      LOGICAL OK,DONE
      INTEGER IDUM,I,IFL,IFR,IFB,IFP,IFG,LNUM,NSPA,PT,LTOT,N,J,JDUM
     +  ,ITILD,K,JTILD,L
C     *
C     *-------------------------------------------------------------
C     open the dms FNAMIW file ( copy of FNAMI plus field wall beds)
C     *-------------------------------------------------------------
      CALL FILBLD(JOB,'DMS_','FNAMIW_',FNAMIW)
      CALL DMSTRA(FNAMIW,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('can not trace FNAMIW _')
        OK=.FALSE.
        RETURN
      END IF
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in FNAMIW isaac file_')
        OK=.FALSE.
        CALL DMSCLO
        RETURN
      END IF
C     -----------------------------------------
C     get fields numbers in dms file records
C     -----------------------------------------
      CALL FINFLD('LEVEL_',IFL)
      CALL FINFLD('ROOM_',IFR)
      CALL FINFLD('GROSS AREA_',IFG)
      CALL FINFLD('ELEMENT_',IFP)
      CALL FINFLD('WALL BED_',IFB)
      IF(IFL.EQ.0.OR.IFR.EQ.0.OR.IFG.EQ.0.OR.IFP.EQ.0.OR.IFB.EQ.0)THEN

        CALL ERRMES('ROOM file format error_')
        CALL DMSCLO
        OK=.FALSE.
        RETURN
```

```
      END IF
C     *-----------------------------------------
C     NOW going through all existing rooms in building
C     *-----------------------------------------
      HMAX=0                          !maximum internal space height
C     ---------------------------------------
      DO 200 L=1,NSPA
        IF (ISTHTS(L).GT.HMAX) THEN
          HMAX=ISTHTS(L)
        END IF
200   CONTINUE
      SPALEV=HMAX
C     *
      IF(LNUM.EQ.LTOT-1) THEN
       CALL DROUND(HMAX,STR,0)
300    CONTINUE
       CHR='give external height for top level [ '//STR(2:)
       ITILD=INDEX(CHR,'_')
       CHR(ITILD:)='mms ] _'
       CALL CHARIN(0,CHR,0,TXT,K,N,R)
       IF (K.EQ.1) THEN
         CALL ERRMES('improper input_')
         GOTO 300
       ELSE IF (K.GE.2) THEN
         SPALEV=R
       END IF
      END IF
C     ***********************************
      DO 100 J=1,NREC
       JDUM=J
       CALL GETFLD(JDUM,IFR,CHR,K,N,R)
       IF (LNUM.GT.0) THEN
         PT=N-(LNUM*10000)
       ELSE
         PT=N
       END IF
C      *----------------------------------------
       CALL GETFLD(JDUM,IFL,CHR,K,N,R)
C      *----------------------------------------
       IF (N.EQ.LNUM) THEN
C       *---------------------------------------
        CALL GETFLD(JDUM,IFP,CHR,K,N,R)
        JTILD=INDEX(CHR,'_')
        STR=CHR(1:JTILD)
        IF (STR.EQ.'INTERNAL WALL_') THEN
          CALL GETFLD(JDUM,IFB,CHR,K,N,R)
          INWABD(PT)=INWABD(PT)+(R/2)              !internal wall bed

          CALL GETFLD(JDUM,IFG,CHR,K,N,R)
          IWAARS(PT)=IWAARS(PT)+R
        END IF
C       *----------------------------------------
        ESTHTS(PT)=SPALEV              ! what ever in the space
C       *
        IF (STR.EQ.'CEILING WITH A ROOM ABOVE_') THEN
```

```
            CALL GETFLD(JDUM,IFG,CHR,K,N,R)
            CEILAR(PT)=R
            IF (LVHT(LNUM).EQ.-1) THEN
              ESTHTS(PT)=SPALEV
            ELSE
              ESTHTS(PT)=LVHT(LNUM)              ! level storey hight
            END IF
          END IF
C         *-----------------------------------------
          IF(STR.EQ.'CEILING WITH A ROOF ABOVE ')THEN
           CALL GETFLD(JDUM,IFG,CHR,K,N,R)
           CEILAR(PT)=R
           IF(LNUM.EQ.LTOT-1) THEN
             ESTHTS(PT)=SPALEV
           ELSE
             ESTHTS(PT)=LVHT(LNUM)
           END IF
          END IF
C         *-----------------------------------------
          IF(STR.EQ.'CEILING WITHIN A ROOM ') THEN
           CALL GETFLD(JDUM,IFG,CHR,K,N,R)
           SUCEAR(PT)=SUCEAR(PT)+R
          END IF
C         *-----------------------------------------
          IF (STR.EQ.'FLOOR WITH GROUND BELOW ') THEN
           CALL GETFLD(JDUM,IFG,CHR,K,N,R)
           GSLABA(PT)=GSLABA(PT)+R+INWABD(PT)+EXWABD(PT)
          END IF
C         *-----------------------------------------
         END IF
100     CONTINUE
        CALL DMSCLO
        RETURN
        END
```

```
      ***********************************************
      SUBROUTINE SFQPT3(FNAMBW,LNUM,NSPA,LTOT,DST,OK)
      ***********************************************
      IMPLICIT NONE
C     *
C     opens dms LAYOUT file and extracts data for SFQ common
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*50 STR,TOT,FNAMBW,CHR
      REAL*8 R,DST(0:99)
      LOGICAL OK,DONE,SET
      INTEGER I,N,ERROR,IDUM,ILOW,LNUM,NSPA,LTOT,JDUM,K
      INTEGER IFL,IFE,IFB,IFG,IF1,IF2,PT,JREC,JTILD,J,IFT
C     *
C     open dms LAYOUT file ( work fiel ) -------
C     *
      CALL FILBLD(JOB,'DMS_','FNAMBW_',FNAMBW)
      CALL DMSTRA(FNAMBW,DONE)          !working layout file
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
         CALL ERRMES('no element in LAYOUT working file_')
         OK=.FALSE.
         CALL DMSCLO
         RETURN
      END IF
C     *----------------------------------------
C     get fields number in dms file records
C     *----------------------------------------
      CALL FINFLD('LEVEL_',IFL)
      CALL FINFLD('ELEMENT_',IFE)
      CALL FINFLD('WALL BED_',IFB)
      CALL FINFLD('GROSS_AREA_',IFG)
      CALL FINFLD('GIRTH_',IFT)
      CALL FINFLD('ROOM NUMBER 1_',IF1)
      CALL FINFLD('ROOM NUMBER 2_',IF2)

IF(IFL.EQ.0.OR.IFE.EQ.0.OR.IFG.EQ.0.OR.IF1.EQ.0.OR.IF2.EQ.0)THEN
         CALL ERRMES('ROOM file format error_')
         CALL DMSCLO
         OK=.FALSE.
         RETURN
      END IF
C     ------------------------------------------------------
C     NOW going through all  rooms in building ( level )
C     ------------------------------------------------------
      DO 100 J=1,NREC
      JDUM=J
      CALL GETFLD(JDUM,IFL,CHR,K,N,R)
      IF (N.EQ.LNUM) THEN
C       *---------------------------------------
        CALL GETFLD(JDUM,IFE,CHR,K,N,R)          !element
```

```
            JTILD=INDEX(CHR,'_')
            STR=CHR(1:JTILD)
            IF (STR.EQ.'EXTERNAL WALL_') THEN
              CALL GETFLD(JDUM,IF1,CHR,K,N,R)          !room 1
              IF (N.EQ.0) THEN
                CALL GETFLD(JDUM,IF2,CHR,K,N,R)        !room 2
              END IF
              IF (LNUM.GT.0) THEN
                PT=N-(LNUM*10000)
              ELSE
                PT=N
              END IF
              CALL GETFLD(JDUM,IFG,CHR,K,N,R)            !gross area
              EWAARS(PT)=EWAARS(PT)+R
              CALL GETFLD(JDUM,IFT,CHR,K,N,R)            !girth
              EWAARS(PT)=EWAARS(PT)+(R*(DST(LNUM)/1000D0))
              CALL GETFLD(JDUM,IFB,CHR,K,N,R)            !wall bed
              EXWABD(PT)=EXWABD(PT)+R
            END IF
          END IF
100       CONTINUE
          CALL DMSCLO
C         *
          RETURN
          END
```

```
      ************************************
      SUBROUTINE SFQPT4(FNAMO,LNUM,NSPA,OK)
      ************************************
      IMPLICIT NONE
C     *
C     opens dms OPENING file and extracts data for SFQ common
C     *
      CHARACTER*50 FNAMO,CHR,STR
      REAL*8 R,AREA
      LOGICAL OK,DONE
      INTEGER JDUM,IFL,IFG,IFE,IFP,IFT,IF1,IF2,LNUM,NSPA,JTILD,IDUM

      INTEGER PT1,PT2,J,K,N
C     *
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
C     open the dms OPENING file
C     *
      CALL DMSTRA(FNAMO,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES('no element in OPENING file')
C       OK=.FALSE.
C       CALL DMSCLO
        RETURN
      END IF
C     *
C     ---------------------------------------------
C     get fields numbers in dms file records
C     ---------------------------------------------
      CALL FINFLD('LEVEL',IFL)
      CALL FINFLD('ELEMENT',IFE)
      CALL FINFLD('GROSS AREA',IFG)
      CALL FINFLD('ROOM NUMBER 1',IF1)
      CALL FINFLD('ROOM NUMBER 2',IF2)
      CALL FINFLD('INT EXT',IFT)

IF(IFL.EQ.0.OR.IFG.EQ.0.OR.IF1.EQ.0.OR.IF2.EQ.0.OR.IFT.EQ.0)THEN
         CALL ERRMES('ROOM file format error')
         CALL DMSCLO
         OK=.FALSE.
         RETURN
      END IF
C     *---------------------------------------------
C     NOW going through all existing rooms in building
C     *---------------------------------------------
      AREA=0D0
      DO 100 J=1,NREC
      JDUM=J
      CALL GETFLD(JDUM,IFL,CHR,K,N,R)
      IF (N.EQ.LNUM) THEN
C         *---------------------------------------
          CALL GETFLD(JDUM,IFG,CHR,K,N,R)          !gross area
          AREA=R
```

```
            CALL GETFLD(JDUM,IF1,CHR,K,N,R)          !room 1
            IF (LNUM.GT.0) THEN
              PT1=N-(LNUM*10000)
            ELSE
              PT1=N
            END IF
            CALL GETFLD(JDUM,IFT,CHR,K,N,R)          !int ext
            JTILD=INDEX(CHR,'_')
            STR=CHR(1:JTILD)
            IF (STR.EQ.'INTERNAL_') THEN
              CALL GETFLD(JDUM,IF2,CHR,K,N,R)        !room 2
              IF (LNUM.GT.0) THEN
                PT2=N-(LNUM*10000)
              ELSE
                PT2=N
              END IF
              CALL GETFLD(JDUM,IFE,CHR,K,N,R)        !element
              JTILD=INDEX(CHR,'_')
              STR=CHR(1:JTILD)
              IF (STR.EQ.'DOOR_') THEN
               IDOARS(PT2)=IDOARS(PT2)+AREA
               IDOARS(PT1)=IDOARS(PT1)+AREA
              ELSE
               IWIARS(PT2)=IWIARS(PT2)+AREA
               IWIARS(PT1)=IWIARS(PT1)+AREA
              END IF
            ELSE
              CALL GETFLD(JDUM,IFE,CHR,K,N,R)        !element
              JTILD=INDEX(CHR,'_')
              STR=CHR(1:JTILD)
              IF (STR.EQ.'WINDOW_') THEN
               EWIARS(PT1)=EWIARS(PT1)+AREA
              ELSE
               EDOARS(PT1)=EDOARS(PT1)+AREA
              END IF
            END IF
          END IF
C           IDUM=IDUM+1                          ! NOT NEEDED
C       END IF
100     CONTINUE
        CALL DMSCLO
        RETURN
        END
```

```
      ************************
      SUBROUTINE SFTOFF(SET,OK)
      ************************
      IMPLICIT NONE
C     *
C     get SFQ data from a DMS file
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
C     *
C     18-FEB-88 REV_1.16 just for a selected set of spaces
C     05-APR-88 REV_1.16 TAB include number of spaces selected
C     *
      CHARACTER*50 CHR,STR
      REAL*8 R,INWABD(100),EXWABD(100)
      INTEGER ILV,GRA,SPT,SPN,STY,EWB,IWB,EWH,IWH,EWA,IWA,EWI,
     1  IWI,EDA,IDA,CEA,SUP,GSA,ROA,I,K,N,IDUM,LNUM,M
      LOGICAL DONE,OK,SET,SPASEL(2000),LEVSEL(100)
      COMMON/SELECT/LEVSEL,SPASEL
C     *
      OK=.TRUE.
C     *
      IF (.NOT.SET) THEN
       DO 110 I=1,100
          LEVSEL(I)=.TRUE.
110    CONTINUE
       DO 120 I=1,2000
          SPASEL(I)=.TRUE.          !initialising selected spaces
120    CONTINUE
      END IF
C     *
      DO 100 I=1,100
        LEVELN(I)=0
        GRAREA(I)=0D0
        GSAREA(I)=0D0
        LUAREA(I)=0D0
        EXSTHT(I)=0D0
        INSTHT(I)=0D0
        EXWABD(I)=0D0
        INWABD(I)=0D0
        EXWAAR(I)=0D0
        INWAAR(I)=0D0
        EXWIAR(I)=0D0
        INWIAR(I)=0D0
        EXDOAR(I)=0D0
        INDOAR(I)=0D0
        ROAREA(I)=0D0
        CEAREA(I)=0D0
        EXWAGI(I)=0D0
        DSTND(I)=0D0
        NUMSPA(I)=0
        PILEHD(I)=0                  ! number of spaces/floor
        STRFLT(I)=0
```

```
          LFTFLT(I)=0
          SUSCLG(I)=0D0
          SANFIT(I)=0
100       CONTINUE
C         *
C         get SFQ file fields numbers
C         *
          OK=.TRUE.
C         *
          CALL FINFLD('LEVEL ',ILV)
          CALL FINFLD('GROSS AREA ',GRA)
          CALL FINFLD('SPACE TITLE ',SPT)
          CALL FINFLD('SPACE NUMBER ',SPN)
          CALL FINFLD('SPACE TYPE ',STY)
          CALL FINFLD('EXT WALL BEDS ',EWB)
          CALL FINFLD('INT WALL BEDS ',IWB)
          CALL FINFLD('EXT STOREY HT ',EWH)
          CALL FINFLD('INT STOREY HT ',IWH)
          CALL FINFLD('EXT WALL AREA ',EWA)
          CALL FINFLD('INT WALL AREA ',IWA)
          CALL FINFLD('EXT WIND AREA ',EWI)
          CALL FINFLD('IXT WIND AREA ',IWI)
          CALL FINFLD('EXT DOOR AREA ',EDA)
          CALL FINFLD('INT DOOR AREA ',IDA)
          CALL FINFLD('GR. SLAB AREA ',GSA)
          CALL FINFLD('SUSP. CEILING ',SUP)
          CALL FINFLD('ROOF AREA ',ROA)
          CALL FINFLD('CEILING AREA ',CEA)
C         *
          IF (ILV.EQ.0.OR.GRA.EQ.0.OR.EWB.EQ.0.OR.CEA.EQ.0) THEN
            CALL ERRMES('SFQ file format error ')
            CALL DMSCLO
            OK=.FALSE.
            RETURN
          END IF
200       CONTINUE
C         *
          DO 1000 I=1,NREC
           IDUM=I
           CALL GETFLD(IDUM,ILV,CHR,K,N,R)
           LNUM=N+1
           IF(LEVSEL(LNUM)) THEN
             LEVELN(LNUM)=LNUM
             CALL GETFLD(IDUM,SPN,CHR,K,N,R)
             IF(SPASEL(N)) THEN
              NUMSPA(LNUM)=NUMSPA(LNUM)+1          !incrementing spaces
              CALL GETFLD(IDUM,GRA,CHR,K,N,R)
              GRAREA(LNUM)=GRAREA(LEVELN(LNUM))+R
              CALL GETFLD(IDUM,IWB,CHR,K,N,R)
              INWABD(LNUM)=INWABD(LEVELN(LNUM))+R
              CALL GETFLD(IDUM,EWB,CHR,K,N,R)
              EXWABD(LNUM)=EXWABD(LEVELN(LNUM))+R
              CALL GETFLD(IDUM,IWH,CHR,K,N,R)
              INSTHT(LNUM)=INSTHT(LEVELN(LNUM))+R
```

```
            CALL GETFLD(IDUM,EWH,CHR,K,N,R)
            EXSTHT(LNUM)=EXSTHT(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,IWA,CHR,K,N,R)
            INWAAR(LNUM)=INWAAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,EWA,CHR,K,N,R)
            EXWAAR(LNUM)=EXWAAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,IWI,CHR,K,N,R)
            INWIAR(LNUM)=INWIAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,EWI,CHR,K,N,R)
            EXWIAR(LNUM)=EXWIAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,IDA,CHR,K,N,R)
            INDOAR(LNUM)=INDOAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,EDA,CHR,K,N,R)
            EXDOAR(LNUM)=EXDOAR(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,ROA,CHR,K,N,R)
            ROAREA(LNUM)=ROAREA(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,SUP,CHR,K,N,R)
            SUSCLG(LNUM)=SUSCLG(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,GSA,CHR,K,N,R)
            GSAREA(LNUM)=GSAREA(LEVELN(LNUM))+R
            CALL GETFLD(IDUM,CEA,CHR,K,N,R)
            CEAREA(LNUM)=CEAREA(LEVELN(LNUM))+R
          END IF
        END IF
1000    CONTINUE
C       *-----------------------------------------------
        DO 2000 M=1,100
         IF(LEVELN(M).EQ.M) THEN
           LUAREA(M)=GRAREA(M)
           GRAREA(M)=GRAREA(M)+INWABD(M)
C          IF (LEVELN(M)-1.EQ.0) THEN
C            GSAREA(M)=GRAREA(M)+INWABD(M)+EXWABD(M)
C          END IF
           IF(INWIAR(M).GT.0) INWIAR(M)=INWIAR(M)/2
           IF(INDOAR(M).GT.0) INDOAR(M)=INDOAR(M)/2   !

           IF (NUMSPA(M).GT.0) THEN
            EXSTHT(M)=EXSTHT(M)/NUMSPA(M)
            INSTHT(M)=INSTHT(M)/NUMSPA(M)
           END IF
           NUMSPA(M)=-1*NUMSPA(M)        !for better identification
         END IF
2000    CONTINUE
C       *
        CALL COMFFQ(SET)              !complementary FFQs
C       *
        CALL DMSCLO
        OK=.TRUE.
C       *
        RETURN
        END
```

```
      **********************
      SUBROUTINE SGRMNR(OK)
      **********************
C     *
C     space grouping routine
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
C     *
      CHARACTER*50 NAME,FNAM,CHR,STR,PROMPT,STRL
      REAL*8 R
      INTEGER LVN,SPN,SPT,GSR,JNREC,N,I,SPACES,LEVEL,SPANUM
      LOGICAL OK,GETYN,DONE,SPASEL(2000),LEVSEL(100),SET
      COMMON/SELECT/LEVSEL,SPASEL
C     *
      DO 110 I=1,100
       LEVSEL(I)=.FALSE.          !initialising selected levels
110   CONTINUE
      DO 120 I=1,2000
       SPASEL(I)=.FALSE.          !initialising selected spaces
120   CONTINUE
C     *
      OK=.TRUE.
      SPACES=0
C     *
C     get SFQ file name
C     *
      NAME='SFQ_'
      CALL GETIFL(NAME,FNAM,OK)
      IF(.NOT.OK) THEN
       CALL ERRMES('can not get SFQ file_')
       RETURN
      END IF
C     *
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) THEN
        CALL ERRMES('can not trace SFQ file_')
        OK=.FALSE.
        RETURN
      END IF
C     *
      IF(NREC.LT.1) THEN
       CALL ERRMES('no element in SFQ file_')
       RETURN
      END IF
C     *
C     getting fiels numbers
C     *
      CALL FINFLD('LEVEL_',LVN)
      CALL FINFLD('SPACE NUMBER_',SPN)
      CALL FINFLD('SPACE TITLE_',SPT)
      CALL FINFLD('GROSS AREA_',GSR)
      IF (LVN.EQ.0.OR.SPN.EQ.0.OR.SPT.EQ.0.OR.GSR.EQ.0) THEN
        CALL ERRMES('SFQ file format error_')
```

```
          CALL DMSCLO
          OK=.FALSE.
          RETURN
        END IF
C       *
        CALL DPAGE
        CALL CHAROT('SFQ SPACE SELECTION ')
200     CONTINUE
        CALL CHAROT(' ')
        STR='enter 0-99 for one level in building  '
        CALL CHARIN(0,STR,0,CHR,K,LEVEL,R)
        IF (K.EQ.0) THEN
          OK=.FALSE.
          RETURN
        END IF
C       *
        IF (K.EQ.1.OR.LEVEL.LT.0.OR.LEVEL.GT.99) THEN
          CALL ERRMES('must be in range 0-99 ')
          GOTO 200                !try again
        END IF
        LEVSEL(LEVEL+1)=.TRUE.
C       *
        DO 100 I=1,NREC
         PROMPT=' '
         JNREC=I
         CALL GETFLD(JNREC,LVN,CHR,K,N,R)
         IF (N.EQ.LEVEL) THEN           !selected space
          CALL IROUND(N,STRL,J)
          JTILD=INDEX(STRL,' ')-1
          PROMPT(35:)=STRL(1:JTILD)            !put level in prompt
          CALL GETFLD(JNREC,SPN,CHR,K,N,R)
          SPANUM=N
          CALL IROUND(N,STR,J)
          JTILD=INDEX(STR,' ')-1
          PROMPT(1:)=STR(1:JTILD)         !put space No in prompt
          CALL GETFLD(JNREC,SPT,CHR,K,N,R)
          JTILD=INDEX(CHR,' ')-1
          PROMPT(11:)=CHR(1:JTILD)        !put space title in prompt
          CALL GETFLD(JNREC,GSR,CHR,K,N,R)
          CALL DROUND(R,STR,J)
          JTILD=INDEX(STR,' ')-1
          PROMPT(25:)=STR(1:JTILD)        !put area in prompt + ' '
          PROMPT(39:39)=' '
          IF (.NOT.GETYN(PROMPT,'N')) THEN
            SPACES=SPACES+1
            SPASEL(SPANUM)=.TRUE.
          END IF
         END IF
100     CONTINUE
C       *
        CALL CHAROT(' ')
        CALL IROUND(SPACES,STR,J)
        CHR='number of spaces selected '//STR(2:)
        JTILD=INDEX(CHR,' ')
```

```
      CHR(JTILD:)='..confirm ¯'
      IF (.NOT.GETYN(CHR,'Y')) THEN
        OK=.FALSE.
        RETURN
      END IF
C     *
      SET=.TRUE.
      CALL SFTOFF(SET,OK)
      IF (.NOT.OK) RETURN
C     *
      RETURN
      END
```

```
      ********************************
      SUBROUTINE SPASEL(FNAML,FNAMR,OK)
      ********************************
C     *
C     select a number of spaces
C     *
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
C     *
      CHARACTER*50 PROMPT,FNAML,FNAMR,CHR,STR,STRL,FNAMI,FNAMB
     +     ,FNAMIW,FNAMBW,FILE
      INTEGER IFL,RMT,RMN,GSR,IFR,JREC,SPACES,SPANUM,LEVEL,SSEL(100)
     +          ,LVN,NUM,ERROR,SPEC
      REAL*8 R
      LOGICAL FOUND,DONE,OK,GETYN,SET
C     *
      CALL WALINI                          !initialise BMSSP1
C     *
      SET=.FALSE.
      DO 10 L=1,100
        SSEL(L)=0
10    CONTINUE
      SPACES=0
C     *--------------------------
C     open the DMS LEVEL file
C     *--------------------------
      CALL DMSTRA(FNAML,DONE)
      IF(.NOT.DONE) RETURN
      IF(NREC.LT.1) THEN
        CALL ERRMES('no levels in LEVEL file_')
        CALL DMSCLO
        RETURN
      END IF
C     *
      CALL FINFLD('LEVEL_',IFL)
      CALL FINFLD('NO OF ROOMS_',IFR)
      IF(IFL.EQ.0.OR.IFR.EQ.0) THEN
        CALL ERRMES('LEVEL file format error_')
        CALL DMSCLO
        RETURN
      END IF
C     *-- ---------------------------------
C     get LEVEL number for space selection
C     *-----------------------------------
      CALL DPAGE                             !clear screen
      CALL CHAROT('SELECT BUILDING LEVEL_')
200   CONTINUE
      CALL CHAROT(' _')
      STR='entre 0-99 for on level in building _'
      CALL CHARIN(0,STR,0,CHR,K,LEVEL,R)
      IF (K.EQ.0) THEN
        OK=.FALSE.
        RETURN
```

```
      END IF
      IF (K.EQ.1.OR.LEVEL.LT.0.OR.LEVEL.GT.99) THEN
        CALL ERRMES('must be in range 0-99_')
        GOTO 200
      END IF
C     *-----------------------------------
C     check if selected floor level exist in LEVEL file
C     *-----------------------------------
      FOUND=.FALSE.
      DO 500 I=1,NREC
       JREC=I
       CALL GETFLD(JREC,IFL,CHR,K,N,R)
       IF (N.EQ.LEVEL) THEN
         FOUND=.TRUE.                ! level exist
         SSEL(1)=LEVEL               ! first test string
         SSEL(2)=0                   ! INTERNAL WALL field
         SSEL(3)=0                   ! EXTERNAL WALL field
       END IF
500   CONTINUE
C     *
      IF (.NOT.FOUND) THEN
         CALL ERRMES('level does not exist_')
         GOTO 200
      END IF
C     *
      CALL DMSCLO                    ! close LEVEL dms file
C     *--------------------------
C     open the DMS ROOM file
C     *--------------------------
      CALL DMSTRA(FNAMR,DONE)
      IF(.NOT.DONE) RETURN
      IF(NREC.LT.1) THEN
        CALL ERRMES('no levels in LEVEL file_')
        CALL DMSCLO
        RETURN
      END IF
C     *
      CALL FINFLD('LEVEL_',IFL)
      CALL FINFLD('ROOM ',RMN)                    !room number
      CALL FINFLD('ROOM NAME_',RMT)               !room name
      CALL FINFLD('GROSS AREA_',GSR)              !gross area
      IF(IFL.EQ.0.OR.IFR.EQ.0) THEN
        CALL ERRMES('LEVEL file format error_')
        CALL DMSCLO
        RETURN
      END IF
C     *-----------------------------------
C     get selection of spaces in floor level
C     *-----------------------------------
      CALL DPAGE                          !clear screen
      CALL CHAROT('SFQ SPACE SELECTION_')
      CALL CHAROT(' _')
C     *-----------------------------------
      NUM=3
```

```
C       *-------------------------------------
C       trace copy of ISAAC file
C       *-------------------------------------
        CALL FILBLD(JOB,'DMS_','FNAMIW_',FNAMIW)
        CALL DMSTRA(FNAMIW,OK)            !trace copy of ISAAC file
        IF (.NOT.DONE) RETURN
        IF (NREC.LT.1) THEN
          CALL ERRMES('no element in ISAAC file_')
          OK=.FALSE.
          CALL DMSCLO
          RETURN
        END IF
C       *-----------------------------------------
C       set up search pattern for replacing wall spec
C       *-----------------------------------------
        CALL SEAPT2(SSEL,SPACES,OK)
        IF (.NOT.OK) THEN
          CALL ERRMES('error in search pattern_')
          RETURN
        END IF
C       *
        CALL ISAREP(SPEC,DONE)  !replacing in ISAAC file
        IF (.NOT.DONE) THEN
           OK=.FALSE.
           RETURN
        END IF
C       *
        CALL DMSCLO
C       *-------------------------------------
C       trace copy of LAYOUT file
C       *-------------------------------------
        CALL FILBLD(JOB,'DMS_','FNAMBW_',FNAMBW)
        CALL DMSTRA(FNAMBW,OK)            !trace copy of LAYOUT file
        IF (.NOT.DONE) RETURN
        IF (NREC.LT.1) THEN
          CALL ERRMES('no element in LAYOUT file_')
          OK=.FALSE.
          CALL DMSCLO
          RETURN
        END IF
C       *-----------------------------------------
C       set up search pattern for replacing wall spec
C       *-----------------------------------------
        CALL SEAPT3(SSEL,SPACES,OK)
        IF (.NOT.OK) THEN
          CALL ERRMES('error in search pattern_')
          RETURN
        END IF
C       *
        CALL LAYREP(SPEC,DONE)  !replacing in LAYOUT file
        IF (.NOT.DONE) THEN
          OK=.FALSE.
          RETURN
        END IF
```

```
      DO 100 I=1,NREC
       PROMPT='                                                    ¯'
       JNREC=I
       CALL GETFLD(JNREC,IFL,CHR,K,N,R)
       IF (N.EQ.LEVEL) THEN                !selected space
        CALL IROUND(N,STRL,J)
        JTILD=INDEX(STRL,'¯')-1
        PROMPT(35:)=STRL(1:JTILD)          !put level in prompt
        CALL GETFLD(JNREC,RMN,CHR,K,N,R)
        IF (LEVEL.GT.0) THEN
          SPANUM=N-(LEVEL*10000)
        ELSE
          SPANUM=N
        END IF
        CALL IROUND(SPANUM,STR,J)
        JTILD=INDEX(STR,'¯')-1
        PROMPT(1:)=STR(1:JTILD)            !put space No in prompt
        CALL GETFLD(JNREC,RMT,CHR,K,N,R)
        JTILD=INDEX(CHR,'¯')-1
        PROMPT(11:)=CHR(1:JTILD)           !put space title in prompt
        CALL GETFLD(JNREC,GSR,CHR,K,N,R)
        CALL DROUND(R,STR,J)
        JTILD=INDEX(STR,'¯')-1
        PROMPT(25:)=STR(1:JTILD)              !put area in prompt + '

        PROMPT(39:39)='¯'
        IF (GETYN(PROMPT,'N')) THEN
          SPACES=SPACES+1
          NUM=NUM+1
          SSEL(NUM)=SPANUM
        END IF
       END IF
100    CONTINUE
C      *
       CALL CHAROT(' ¯')
       NUM=NREC-SPACES
       CALL IROUND(NUM,STR,J)
       CHR='number of spaces selected '//STR(2:)
       JTILD=INDEX(CHR,'¯')
       CHR(JTILD:)='..confirm ¯'
       IF (.NOT.GETYN(CHR,'Y')) THEN
         OK=.FALSE.
         RETURN
       END IF
C      *
       IF (SPACES.EQ.0) RETURN
C      *
       CALL GTSURF(SPEC,OK)       !get surface spec to replace with
       IF (.NOT.OK) RETURN
       IF (SPEC.LT.1.OR.SPEC.GT.99) THEN
          CALL ERRMES('error in wall specification¯')
          RETURN
       END IF
C      *
       CALL DMSCLO
```

```
C       *
        CALL DMSCLO
C       *
        RETURN
        END
```

```
      ******************
      SUBROUTINE TOTOUT
      ******************
C     *
C     printout total of ELEMENTAL COST of BEQ file
C     *
%INCLUDE ':GABLE:V31:DMS:DMSR.COM'
C     *
C     22-MAR-88 REV_1.16 TAB include percentage in report total
C     *
      LOGICAL SET,MESSAGE
      INTEGER LINES
C     *
      RAWDAT=.FALSE.
      NFLD=2
      FIELD=.FALSE.
      SPACE=.FALSE.
      HEAD=.TRUE.
      HEADTX='BUILDING ELEMENTAL COST MODEL'
      IRPFLD(1)=5
      IRPFLD(2)=10
      IRPCOL(1)=62
      IRPCOL(2)=73
      IRPW=78
      TOTAL(1)=.TRUE.
      TOTAL(2)=.TRUE.
C     *
      SET=.FALSE.
      LINES=0
      MESSAGE=.FALSE.
      CALL REPOUT(SET,LINES,MESSAGE)
C     *
      RETURN
      END
```

```
      **********************
      SUBROUTINE UPDRAT(OK)
      **********************
C     *
C     make an update of rates from COST DATA ( in job COST_MOD )
C     file in current BEM file.
C     *
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMSGT.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
      CHARACTER*50 CHR,STR,SNAME,SOURCE,NAME,FNAM,SNAMI
      CHARACTER*15 INDNAM,FLDJNM(200)
      INTEGER JFLDTP(200),KYJFLD(200),LKJFLD(200),JSTA(200),
     + JFIN(200),JTILD,JICHAN,JDCHAN,JINDX,IINDX,JSOUR
      LOGICAL EXIST,DONE,OK,SET
      COMMON/SOURCE/JFLDTP,KYJFLD,LKJFLD,JSTA,JFIN
C     *
      OK=.TRUE.
C     *
      NAME='BEM_'
      CALL GETIFL(NAME,FNAM,OK)
      IF(.NOT.OK) THEN
       CALL ERRMES('can not get BEM file_')
       RETURN
      END IF
C     *
C     trace DMS file
C     *
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) THEN
         CALL ERRMES('can not trace file_')
         OK=.FALSE.
         RETURN
      END IF
C     *
      IF (NREC.LT.1) THEN
        CALL ERRMES(' no space fundamental quantitites_')
        CALL DMSCLO
        OK=.FALSE.
        RETURN
      END IF
C     *
C     verify source file conditions to importing from
C     *
      SNAME='RATES_'           !dms_file name were cost data are
      CALL FILBLD('COST_MOD_','DMS_',SNAME,FNAM)
      CALL INQUIR(FNAM,EXIST)
      IF(.NOT.EXIST) THEN
        CALL ERRMES('ff462ffile not found_')
        OK=.FALSE.
        RETURN
      END IF
```

```
C       *
C       set up index field
C       *
        IINDX=1                 !index field in BEQ file (CODE)
C       *
C       read index file of source file
C       *
        SOURCE=FNAM
        ITILD=INDEX(SOURCE,'_')-1
        SNAMI=SOURCE(1:ITILD)//'.IND_'
        CALL FILOPY(SNAMI,1,0,JICHAN)
        IF(JICHAN.EQ.0) THEN
          CALL ERRMES('££349£££file index missing_')
          OK=.FALSE.
          RETURN
        ELSE
          JINDX=0
          INDNAM=FLDNAM(IINDX)
          JTILD=INDEX(INDNAM,'_')-1
          READ(JICHAN) JF9,JFLDW,JRFLD,JNREC
          DO 1200 I=1,JF9
            READ(JICHAN) FLDJNM(I),JFLDTP(I),KYJFLD(I),LKJFLD(I)
            IF(FLDJNM(I)(1:JTILD).EQ.INDNAM(1:JTILD)) THEN
              JINDX=I
            END IF
1200      CONTINUE
          CALL CLOIOC(JICHAN)
        END IF
C       *
C       open DMS source date file
C       *
        CALL FILOPY(SOURCE,0,JFLDW,JDCHAN)
        IF(JDCHAN.EQ.0) THEN
          CALL ERRMES('££350£££file error_')
          OK=.FALSE.
          RETURN
        END IF
        IF(JINDX.EQ.0) THEN
          CALL CHAROT('££3044£££INDEX field not found in SOURCE file_')

          CALL ERRMES('££3045£££update abandoned_')
          CALL CLOIOC(JDCHAN)
          RETURN
        END IF
        IF(JFLDTP(JINDX).NE.IFLDTP(IINDX)) THEN
          CALL CHAROT('££3046£££INDEX field wrong in SOURCE file_')
          CALL ERRMES('££3045£££update abandoned_')
          CALL CLOIOC(JDCHAN)
          RETURN
        END IF
C       *
C       set up target field
C       *
        ITARG=8                 !field targeted in BEQ file ( BASE RATE)

C       *
```

```
C       set up source field
        JSOUR=2              !source field in RATES file
C       ------------------------------------
C       perform actual update
C       ------------------------------------
        SET=.FALSE.
        CALL UPDTRC(ITARG,IINDX,JSOUR,JINDX,JDCHAN,JNREC,JFLDW,SET)
C       *
        CALL CLOIOC(JDCHAN)
        LRDREC=0
C       *
        CALL COMRAT             ! compute RATE*QUANTITY
C       *
        RETURN
        END
```

```
      ***************************
      SUBROUTINE USEFFQ(FNAM,OK)
      ***************************
      IMPLICIT NONE
C     *
C     select FFQ elements
C     *
%INCLUDE ':GABLE:V31:MOD385:FFQ.COM'
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:DMS:DMSA.COM'
%INCLUDE ':GABLE:V31:DMS:DMSC.COM'
%INCLUDE ':GABLE:V31:COMMON:JOB.COM'
C     *
C     *
      CHARACTER*50 FNAM,CHR,TXT
      REAL*8 R
      INTEGER ILV,GRA,GSA,LUA,EWH,IWH,EWA,IWA,EWI,IWI,EDA,IDA,
     1  CEA,ROA,EWG,DST,PIL,STF,LIF,SUC,SAF,JREC,I,IDUM,K,N
      LOGICAL OK,EXIST,DONE
C     *
      DO 100 I=1,100
        LEVELN(I)=0
        GRAREA(I)=0D0
        GSAREA(I)=0D0
        LUAREA(I)=0D0
        EXSTHT(I)=0D0
        INSTHT(I)=0D0
        EXWAAR(I)=0D0
        INWAAR(I)=0D0
        EXWIAR(I)=0D0
        INWIAR(I)=0D0
        EXDOAR(I)=0D0
        INDOAR(I)=0D0
        ROAREA(I)=0D0
        CEAREA(I)=0D0
        EXWAGI(I)=0D0
        DSTND(I)=0D0
        PILEHD(I)=0
        STRFLT(I)=0
        LFTFLT(I)=0
        SUSCLG(I)=0D0
        SANFIT(I)=0
100   CONTINUE
C     *-----------------------------------------
C     check for FFQ file name
C     *-----------------------------------------
      OK=.TRUE.
C     *
C     trace DMS file
C     *
      CALL DMSTRA(FNAM,DONE)          !open DMS file
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
```

```
          CALL ERRMES('no floor fundamental quantities file ')
          CALL DMSCLO
          OK=.FALSE.
          RETURN
        END IF
C       *
        CALL FINFLD('LEVEL ',ILV)
        CALL FINFLD('GROSS AREA ',GRA)
        CALL FINFLD('GRND SLAB AREA ',GSA)
        CALL FINFLD('LET USE AREA ',LUA)
        CALL FINFLD('EXT STOREY HT ',EWH)
        CALL FINFLD('INT STOREY HT ',IWH)
        CALL FINFLD('EXT WALL AREA ',EWA)
        CALL FINFLD('INT WALL AREA ',IWA)
        CALL FINFLD('EXT WIND AREA ',EWI)
        CALL FINFLD('IXT WIND AREA ',IWI)
        CALL FINFLD('EXT DOOR AREA ',EDA)
        CALL FINFLD('INT DOOR AREA ',IDA)
        CALL FINFLD('ROOF AREA ',ROA)
        CALL FINFLD('CEILING AREA ',CEA)
        CALL FINFLD('PILE HEADS ',PIL)
        CALL FINFLD('STRCASE FLIGHT ',STF)
        CALL FINFLD('LIFT FLIGHTS ',LIF)
        CALL FINFLD('SUSPENDED CEAR ',SUC)
        CALL FINFLD('SANITARY FITNS ',SAF)
C       *
        IF (ILV.EQ.0.OR.GRA.EQ.0.OR.ROA.EQ.0.OR.CEA.EQ.0) THEN
          CALL ERRMES('FFQ file format error ')
          CALL DMSCLO
          OK=.FALSE.
          RETURN
        END IF
        IDUM=1
400     CONTINUE
C       *-------------------------------------------------
        DO 1000 I=1,NREC
         IDUM=I
         CALL GETFLD(IDUM,ILV,CHR,K,N,R)
         LEVELN(I)=N
         CALL GETFLD(IDUM,GRA,CHR,K,N,R)
         GRAREA(I)=R
         CALL GETFLD(IDUM,GSA,CHR,K,N,R)
         GSAREA(I)=R
         CALL GETFLD(IDUM,LUA,CHR,K,N,R)
         LUAREA(I)=R
         CALL GETFLD(IDUM,EWH,CHR,K,N,R)
         EXSTHT(I)=R
         CALL GETFLD(IDUM,IWH,CHR,K,N,R)
         INSTHT(I)=R
         CALL GETFLD(IDUM,EWA,CHR,K,N,R)
         EXWAAR(I)=R
         CALL GETFLD(IDUM,IWA,CHR,K,N,R)
         INWAAR(I)=R
         CALL GETFLD(IDUM,IWI,CHR,K,N,R)
```

```
            INWIAR(I)=R
            CALL GETFLD(IDUM,EWI,CHR,K,N,R)
            EXWIAR(I)=R
            CALL GETFLD(IDUM,EDA,CHR,K,N,R)
            EXDOAR(I)=R
            CALL GETFLD(IDUM,IDA,CHR,K,N,R)
            INDOAR(I)=R
            CALL GETFLD(IDUM,ROA,CHR,K,N,R)
            ROAREA(I)=R
            CALL GETFLD(IDUM,CEA,CHR,K,N,R)
            CEAREA(I)=R
C           CALL GETFLD(IDUM,EWG,CHR,K,N,R)
C           CEAREA(I)=R
C           CALL GETFLD(IDUM,DST,CHR,K,N,R)
C           DSTND(I)=R
            CALL GETFLD(IDUM,PIL,CHR,K,N,R)
            PILEHD(I)=N
            CALL GETFLD(IDUM,STF,CHR,K,N,R)
            STRFLT(I)=N
            CALL GETFLD(IDUM,LIF,CHR,K,N,R)
            LFTFLT(I)=N
            CALL GETFLD(IDUM,SUC,CHR,K,N,R)
            SUSCLG(I)=R
            CALL GETFLD(IDUM,SAF,CHR,K,N,R)
            SANFIT(I)=N
1000        CONTINUE
C           *-----------------------------------------------
            CALL DMSCLO
            OK=.TRUE.
            RETURN
            END
```

```
      ******************************
      SUBROUTINE USESFQ(SET,FNAM,OK)
      ******************************
      IMPLICIT NONE
C     *
C     get SFQ data from a DMS file
C     if SET =.true.   just a selected number of spaces is going
C                      to be used. else all spaces
C     *
%INCLUDE ':GABLE:V31:DMS:DMSB.COM'
%INCLUDE ':GABLE:V31:DMS:DMS.COM'
%INCLUDE ':GABLE:V31:MOD385:SFQ.COM'
C     *
C     18-FEB-88 REV_1.16 TAMI add suspended ceiling area
C     *
      CHARACTER*50 FNAM,CHR
      REAL*8 R
      INTEGER ILV,GRA,SPT,SPN,STY,EWB,IWB,EWH,IWH,EWA,IWA,EWI,
     1  IWI,EDA,IDA,SEA,CEA,ROA,I,K,N,IDUM
      LOGICAL DONE,OK,SET
C     *
      DO 100 I=1,2000
       SPATIT(I)='  '
       SPATYP(I)=0
       ROONUM(I)=0
       GAREAS(I)=0D0
       INWABD(I)=0D0
       EXWABD(I)=0D0
       ISTHTS(I)=0D0
       ESTHTS(I)=0D0
       IWAARS(I)=0D0
       EWAARS(I)=0D0
       IWIARS(I)=0D0
       EWIARS(I)=0D0
       IDOARS(I)=0D0
       EDOARS(I)=0D0
       CEILAR(I)=0D0
       ROOFAR(I)=0D0
100    CONTINUE
C     *
C     check for FFQ file name
C     *
      OK=.TRUE.
C     *
C     trace DMS file
C     *
      CALL DMSTRA(FNAM,DONE)
      IF (.NOT.DONE) RETURN
      IF (NREC.LT.1) THEN
        CALL ERRMES(' no space fundamental quantitites  ')
        CALL DMSCLO
        OK=.FALSE.
        RETURN
      END IF
```

```
C       *
        CALL FINFLD('LEVEL ',ILV)
        CALL FINFLD('GROSS AREA ',GRA)
        CALL FINFLD('SPACE TITLE ',SPT)
        CALL FINFLD('SPACE NUMBER ',SPN)
        CALL FINFLD('SPACE TYPE ',STY)
        CALL FINFLD('EXT WALL BEDS ',EWB)
        CALL FINFLD('INT WALL BEDS ',IWB)
        CALL FINFLD('EXT STOREY HT ',EWH)
        CALL FINFLD('INT STOREY HT ',IWH)
        CALL FINFLD('EXT WALL AREA ',EWA)
        CALL FINFLD('INT WALL AREA ',IWA)
        CALL FINFLD('EXT WIND AREA ',EWI)
        CALL FINFLD('IXT WIND AREA ',IWI)
        CALL FINFLD('EXT DOOR AREA ',EDA)
        CALL FINFLD('INT DOOR AREA ',IDA)
        CALL FINFLD('SUSPENDED CEILING ',SEA)
        CALL FINFLD('ROOF AREA ',ROA)
        CALL FINFLD('CEILING AREA ',CEA)
C       *
        IF (ILV.EQ.0.OR.GRA.EQ.0.OR.EWB.EQ.0.OR.CEA.EQ.0) THEN
          CALL ERRMES('SFQ file format error ')
          CALL DMSCLO
          OK=.FALSE.
          RETURN
        END IF
200     CONTINUE
C       *
        DO 1000 I=1,NREC
         IDUM=I
C        IF (SET) THEN                          ! just selected spaces

         CALL GETFLD(IDUM,ILV,CHR,K,N,R)
         LEVELS(I)=N
         CALL GETFLD(IDUM,GRA,CHR,K,N,R)
         GAREAS(I)=R
         CALL GETFLD(IDUM,SPT,CHR,K,N,R)
         SPATIT(I)=CHR
         CALL GETFLD(IDUM,SPN,CHR,K,N,R)
         ROONUM(I)=N
         CALL GETFLD(IDUM,STY,CHR,K,N,R)
         SPATYP(I)=N
         CALL GETFLD(IDUM,IWB,CHR,K,N,R)
         INWABD(I)=R
         CALL GETFLD(IDUM,EWB,CHR,K,N,R)
         EXWABD(I)=R
         CALL GETFLD(IDUM,IWH,CHR,K,N,R)
         ISTHTS(I)=R
         CALL GETFLD(IDUM,EWH,CHR,K,N,R)
         ESTHTS(I)=R
         CALL GETFLD(IDUM,IWA,CHR,K,N,R)
         IWAARS(I)=R
         CALL GETFLD(IDUM,EWA,CHR,K,N,R)
         EWAARS(I)=R
         CALL GETFLD(IDUM,IWI,CHR,K,N,R)
```

```
            IWIARS(I)=R
            CALL GETFLD(IDUM,EWI,CHR,K,N,R)
            EWIARS(I)=R
            CALL GETFLD(IDUM,EDA,CHR,K,N,R)
            EDOARS(I)=R
            CALL GETFLD(IDUM,IDA,CHR,K,N,R)
            IDOARS(I)=R
            CALL GETFLD(IDUM,ROA,CHR,K,N,R)
            ROOFAR(I)=R
            CALL GETFLD(IDUM,SEA,CHR,K,N,R)
            SUCEAR(I)=R
            CALL GETFLD(IDUM,CEA,CHR,K,N,R)
            CEILAR(I)=R
1000        CONTINUE
C           *-----------------------------------------------
            CALL DMSCLO
            OK=.TRUE.
C           *
            RETURN
            END
```

APPENDIX 10.8.2 : ROUTINES INDEX.

```
**********************************
INDEX OF FILES IN LIBRARY MOD385
**********************************

ADDBEM MOD385   store data in file BEM
ADDFFQ MOD385   stores data in file FFQ
ADDSFQ MOD385   stores data in file SFQ
ADJRAT MOD385   rate adjustement routine
BEMMNR MOD385   building elemental quantities main routine
BEMPT1 MOD385   workout content of BEM and store it
BTSELT MOD385   building type selection for COST SELECTION
CALTOT MOD385   calculat total of FFQs and complementary values
COMRAT MOD385   compute rate and quantity in BEM file
COSMOD MOD385   main cost model routine
COSMNR MOD385   cost modelling main routine
COSTSL MOD385   cost selection main routine
CMVER3 MOD385   second version of main cost model routine
COMFFQ MOD385   complementary FFQs from the user
CREFFQ MOD385   creates dms file FFQ for fundamental quantities
CREBEM MOD385   creates dms file BEM for fundamental quantities
CRESFQ MOD385   creates dms file SFQ for space fund. quantities
DISBEM MOD385   display BEM file in a report format
FILREP MOD385   replace particular field of record
FFQMNR MOD385   Floor Fundamental Quantites main routine
FFQPT1 MOD385   extract fields from LEVEL file for FFQ
FFQPT2 MOD385   extract fields from LAYOUT file for FFQ
FFQPT3 MOD385   extract fields from ISAAC
FFQPT4 MOD385   extract fields from ROOM        !not used
FFQPT5 MOD385   extract fields from OPENING
GETOFL MOD385   gets output filenames from user
GETIFL MOD385   gets input filenames from user
GET5FL MOD385   gets 5 630 complete building survey file names
GTSURF MOD385   gets external wall spec for cost modeller
ISAREP MOD385   replace ISAAC data (FNAMIW)
LAYREP MOD385   replace LAYOUT data (FNAMBW)
M385   MOD385   HOST MODULE ROUTINE
PJSELT MOD385   project selection routine
PRCENT MOD385   calculates percentages in BEM file
REPREC MOD385   replace content of a record ut to 9 fields
SCSURF MOD385   search in SURFACES file for project related surface
SEAPT1 MOD385   search pattern subroutine
SEAPT2 MOD385   set up search pattern for ISAAC data replacement
SEAPT3 MOD385   set up search pattern for LAYOUT data replacement
SEARPT MOD385   search set up for report on BEM file
SELELM MOD385   select elements of BEM file
SETSTA MOD385   set status of project rate if adjusted
SFQCPY MOD385   copy ISAAC and LAYOUT in FNAMIW and FNAMBW
SFQCRF MOD385   cross-reference from SURFACES file
SFQMNR MOD385   Space Fundamental Quantites main routine
SFQ124 MOD385   routine calling SFQPT1,SFQPT3,SFQPT3 and SFQPT4
```

SFQPT0	MOD385	extract fields from LEVEL file for SFQ
SFQPT1	MOD385	extract fields from ROOM file for SFQ
SFQPT2	MOD385	extract fields from ISAAC file for SFQ (FNAMIW)
SFQPT3	MOD385	extract fields from LAYOUT file for SFQ (FNAMBW)
SFQPT4	MOD385	extract fields from OPENING file for SFQ
SFTOFF	MOD385	transfer data from SFQ to FFQ format
SGRMNR	MOD385	Space Grouping main routine
SPASEL	MOD385	selects a range of spaces in building model
TOTOUT	MOD385	print out total of ELEMENTAL COST of BEM file
UPDRAT	MOD385	make an update of rates from cost data file to BEM
USEFFQ	MOD385	get records from FFQ file
USESFQ	MOD385	get records from SFQ file