AEROSPACE REPORT NO.
ATR-74(7341)-7

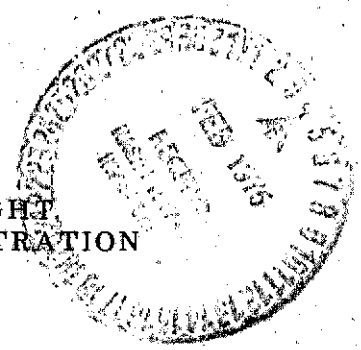
Operations Analysis (Study 2.1)

Program Listing for the LOVES Computer Code

DRA

Prepared by STANLEY T. WRAY, JR.
Information Processing Division

30 September 1974



Prepared for OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C.

Contract No. NASW-2575

Systems Engineering Operations

Aerospace Report No.
ATR-74(7341)-7

OPERATIONS ANALYSIS (STUDY 2.1)

Program Listing for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

30 September 1974

Systems Engineering Operations
THE AEROSPACE CORPORATION
El Segundo, California

Prepared for

OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW 2575

OPERATIONS ANALYSIS (STUDY 2.1)
Program Listing for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Staff Engineer
Evaluation and Planning Systems
Data Processing Subdivision

Approved by

L. Sashkin, Director
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

R.R. Wolfe
NASA Study 2.1 Director
Advanced Mission Analysis
Directorate
Advanced Orbital Systems Division

FOREWORD

This volume contains a listing of the LOVES program. The program is coded partially in SIMSCRIPT I.5 and FORTRAN (CDC FTN compiler). This version of LOVES is compatible with both the CDC 7600 and the UNIVAC 1108 computers. The code printed herein has been compiled, loaded, and executed successfully on the EXEC 8 system for the UNIVAC 1108 at Slidell, Louisiana. This was accomplished during the month of October (1974).

In addition to this volume, a separate report, Operations Analysis (Study 2.1), User's Guide and Programmer's Guide, ATR-74(7341)-6, has been prepared.

Study 2.1, DORCA Applications, is one of several study tasks connected under NASA Contract NASW-2575 in FY 1974. The NASA Study Director was Mr. V. N. Huff, NASA Headquarters, Code MTE.

By agreement with Mr. Huff, the LOVES program will be delivered directly to the NASA Computing Facility.

```
                                                                 DEFINE 2
 1UP       0      IC                                             DEFINE 3
 2DOWN     0      IC                                             DEFINE 4
 3OUT      0      IC                                             DEFINE 5
 4SHUT     0      IC                                             DEFINE 6
 5IUG      0      IC                                             DEFINE 7
 6SEPS     0      IC                                             DEFINE 8
 7BLANK    0      IC                                             DEFINE 9
 8G        0      F                                              DEFINE10
 9TIME3    0      F                                              DEFINE11
10TIMES    0      F                                              DEFINE12
11TIMEC    0      F                                              DEFINE13
12PDOWN    0      I                                              DEFINE14
13WINSJ    0      IC                                             DEFINE15
14WTSU     0      FC                                             DEFINE16
15LENSJ    0      FC                                             DEFINE17
16NMD      0      I                                              DEFINE18
17SU       0      F                                              DEFINE19
18WAIT1    0     SF                                              DEFINE20
19WAIT2    0     SF                                              DEFINE21
20WSATU    0      F                                              DEFINE22
21WSATV    0      F                                              DEFINE23
22WMODU    0      F                                              DEFINE24
23WMODN    0      F                                              DEFINE25
24TRIG     0      I                                              DEFINE26
25TRIG2    0      I                                              DEFINE27
26TRIGS    0      I                                              DEFINE28
27EXVEH    0      I                                              DEFINE29
28PREFT    0      F                                              DEFINE30
29TREFT    0      F                                              DEFINE31
30SREFT    0      F                                              DEFINE32
31PADT     0      F                                              DEFINE33
32DAYS     0      F                                              DEFINE34
33DV       0      F                                              DEFINE35
34FISP     0      F                                              DEFINE36
35WD       0      F                                              DEFINE37
36WPNU     0      F                                              DEFINE38
37WCONS    0      F                                              DEFINE39
38IORB     0      I                                              DEFINE40
39NQ       0      I                                              DEFINE41
40RA       0      F                                              DEFINE42
41VCO      0      F                                              DEFINE43
42RO       0      F                                              DEFINE44
43P1       0      F                                              DEFINE45
44RTFLG    0      I                                              DEFINE46
45PALEN    0      F                                              DEFINE47
46FLYT     0      F                                              DEFINE48
47WAIT3    0     SF                                              DEFINE49
48NEXIT    0      I                                              DEFINE50
49MSEP     0      I                                              DEFINE51
```

```
+N START4          N TIME1 3     F                                                  DEFINE52
+                  N PSAI  31/2  I                                                  DEFINE53
+                  N PMOD  32/2  I                                                  DEFINE54
+N TERM 4          N TIME2 4     F                                                  DEFINE55
+N NWSAT4          N VNAME 4     I                                                  DEFINE56
+                  N TIMEA 4     F                                                  DEFINE57
+N ARRIV4                                                                          DEFINE58
+N BACK 4                                                                          DEFINE59
+N FAIL 4                                                                          DEFINE60
+N LAUNC4                                                                          DEFINE61
+N NEWNE4                                                                          DEFINE62
+N REFMO4                                                                          DEFINE63
+N REFSA4                                                                          DEFINE64
+N REFVE4                                                                          DEFINE65
+N REMOV4                                                                          DEFINE66
+N RETRI4                                                                          DEFINE67
+N SATON4                                                                          DEFINE68
+N WARN 4                                                                          DEFINE69
+                                                                                  DEFINE70
+T NEW  2          T SNEWS 11/2  I 50FNEWS 0     I    NEWS L                     DEFINE71
+                  T SCHSY 12/2  I                                                  DEFINE72
+                  T SCHOT 2     F                                                  DEFINE73
+T FR   4          T SFRS  11/2  I 51FFRS  0     I    FRS    XSATNO L            DEFINE74
+                  T PFRS  12/2  I 52LFRS  0     I                                DEFINE75
+                  T TIMEF 2     F                                                  DEFINE76
+                  T SATNO 31/4  I                                                  DEFINE77
+                  T ST    32/4  I                                                  DEFINE78
+                  T SATSY 33/4  I                                                  DEFINE79
+                  T NPS   34/4  I                                                  DEFINE80
+                  T MOONO 41/2  I                                                  DEFINE81
+                  T NOSTA 42/2  I                                                  DEFINE82
+T MESET4          T SMES  11/2  I 53FMES  0     I    MES  L                     DEFINE83
+                  T MEDT  2     F                                                  DEFINE84
+T PAYLD8          T SLOAD 11/2  I 54FLOAD 0     I    LOAD   XLQTIM L            DEFINE85
+                  T PLOAD 12/2  I                                                  DEFINE86
+                  T ISAT  23/4  I 55LLOAD 0     I                                DEFINE87
+                  T IMOD  21/2  I                                                  DEFINE88
+                  T IRT   24/4  I                                                  DEFINE89
+                  T ANGLE 3    SF                                                  DEFINE90
+                  T PAYWT 4     F                                                  DEFINE91
+                  T PAYLN 5     F                                                  DEFINE92
+                  T GQTIM 6     F                                                  DEFINE93
+                  T LQTIM 7     F                                                  DEFINE94
+                  T CITEM 82/2  I                                                  DEFINE95
+                                                                                   DEFINE96
+                                  57EXMOD 0     I                                  DEFINE97
+                                  58DELTA 0     F                                  DEFINE98
+                                  59EXTUG 0     F                                  DEFINE99
+                                                                                   DEFIN100
+                                  60NORBS 0     IC                                 DEFIN101
```

```
+                                           61ORBID  1      I                                      DEFIN102
+                                           62ORBDV  1      F                                      DEFIN103
+                                           63ORBPD  1      F                                      DEFIN104
+                                           64ORBRA  1      F                                      DEFIN105
+                                           65ORBVC  1      F                                      DEFIN106
+                                           66ORBTM  1      F                                      DEFIN107
+                                           67RQUP   1      I                                      DEFIN108
+                                           68RQSEP  1      I                                      DEFIN109
+                                           69RQSUT  1      I                                      DEFIN110
+                                           70PQUE   1      I                                      DEFIN111
+                                           71NL     1      I                                      DEFIN112
+                                           72ANMD   1      F                                      DEFIN113
+                                           73W      1     SF                                      DEFIN114
+                                           74NMOEL  1      I                                      DEFIN115
+T ITORB4                T SORB   11/2 I 75FORB   1      I      ORB 1   XTORB  L               DEFIN116
+                        T PITEM  12/2 I 76LORB   1      I                                      DEFIN117
+                        T PORB   2    I                                                        DEFIN118
+                        T IORB   3    F                                                        DEFIN119
+                                           77DV1    1      F                                      DEFIN120
+                                           78EXORB  1      I                                      DEFIN121
+                                                                                                  DEFIN122
+                                           80IL     0      IC                                     DEFIN123
+                                           81FLTIM  1      F                                      DEFIN124
+                                           82ILOAD  1      I                                      DEFIN125
+                                           83PANGL  1     SF                                      DEFIN126
+                                                                                                  DEFIN127
+                                           85NVEH   0      IC                                     DEFIN128
+                                           86NAMEV  1      I                                      DEFIN129
+                                           87DAYSV  1      F                                      DEFIN130
+                                           88ISPV   1      F                                      DEFIN131
+                                           89WDV    1      F                                      DEFIN132
+                                           90WPNUV  1      F                                      DEFIN133
+                                           91WCONV  1      F                                      DEFIN134
+                                           92REFIV  1      F                                      DEFIN135
+                                           93EXPV   1      F                                      DEFIN136
+                                           94PAYLV  1      F                                      DEFIN137
+                                           95IDV    1      I                                      DEFIN138
+                                           96NSTAG  1      I                                      DEFIN139
+                                           97SOLID  1      I                                      DEFIN140
+                                                                                                  DEFIN141
+                                          100NYEAR  0      IC                                     DEFIN142
+                                          101SUTFY  1      I                                      DEFIN143
+                                          102SUM90  1      I                                      DEFIN144
+                                          103MAX90  1      I                                      DEFIN145
+                                          104MIN90  1      I                                      DEFIN146
+                                          105TUGFY  1      I                                      DEFIN147
+                                          106SUM39  1      I                                      DEFIN148
+                                          107MAX39  1      I                                      DEFIN149
+                                          108MIN39  1      I                                      DEFIN150
+                                          109SEPFY  1      I                                      DEFIN151
```

```
+
+  110SUM86  1     I                                  DEFIN152
+  111MAX86  1     I                                  DEFIN153
+  112MIN86  1     I                                  DEFIN154
+                                                     DEFIN155
+  120NSHUT  0     IC                                 DEFIN156
+  121VSHUT  1     I                                  DEFIN157
+  122ISHUT  0     I                                  DEFIN158
+  123MFSUT  0     I                                  DEFIN159
+  124IFSUT  0     I                                  DEFIN160
+  125NFSUT  0     I                                  DEFIN161
+                                                     DEFIN162
+  130NTUG   0     IC                                 DEFIN163
+  131VTUG   1     I                                  DEFIN164
+  132ITUG   0     I                                  DEFIN165
+  133NTFLT  0     I                                  DEFIN166
+  134ITFLT  0     I                                  DEFIN167
+  135MTFLT  0     I                                  DEFIN168
+                                                     DEFIN169
+  140NSEPS  0     IC                                 DEFIN170
+  141VSEPS  1     I                                  DEFIN171
+  142ISEPS  0     I                                  DEFIN172
+  143NFSEP  0     I                                  DEFIN173
+  144IFSEP  0     I                                  DEFIN174
+  145MFSEP  0     I                                  DEFIN175
+                                                     DEFIN176
+  150MITAB  0     IC                                 DEFIN177
+  151MNAME  1     I                                  DEFIN178
+  152ALPF   1     F                                  DEFIN179
+  153BETAF  1     F                                  DEFIN180
+  154ITEMD  1     F                                  DEFIN181
+  155ALPW   1     F                                  DEFIN182
+  156BETAW  1     F                                  DEFIN183
+  157TTWND  1     F                                  DEFIN184
+  158MODWT  1     F                                  DEFIN185
+  159MOVOL  1     F                                  DEFIN186
+  160MCLAS  1     I                                  DEFIN187
+  161MDCNT  1     I                                  DEFIN188
+  162S121   1     I                                  DEFIN189
+  163X121   1     I                                  DEFIN190
+  164N121   1     I                                  DEFIN191
+  165NOWAR  1     I                                  DEFIN192
+  166S125   1     I                                  DEFIN193
+  167X125   1     I                                  DEFIN194
+  168N125   1     I                                  DEFIN195
+  169NOFAL  1     I                                  DEFIN196
+  170S129   1     I                                  DEFIN197
+  171X129   1     I                                  DEFIN198
+  172N129   1     I                                  DEFIN199
+                                                     DEFIN200
+  180SITAB  0     IC                                 DEFIN201
```

```
+                                    181SNAME  1      I                                   DEFIN202
+                                    182SWI    1      F                                   DEFIN203
+                                    183SVOL   1      F                                   DEFIN204
+                                    184PRIOR  1      I                                   DEFIN205
+                                    185INCL   1      F                                   DEFIN206
+                                    186ORBIT  1      I                                   DEFIN207
+                                    187TTSAT  1      F                                   DEFIN208
+                                    188PTSAT  1      F                                   DEFIN209
+                                    189EXSAT  1      I                                   DEFIN210
+                                    190NRSAT  1      I                                   DEFIN211
+                                    191NMODS  1      I                                   DEFIN212
+                                    192POLON  1      I                                   DEFIN213
+                                    193SORTE  1      F                                   DEFIN214
+T MDSAT2              I SMOS   11/2 I194FMDS   1      I     MDS 1 F                       DEFIN215
+                                    195LMDS   1      I                                   DEFIN216
+                                                                                         DEFIN217
+                                    200STST3  0      IC                                  DEFIN218
+                                    201SYNAM  1      I                                   DEFIN219
+                                    202TTSYS  1      F                                   DEFIN220
+                                    203PTTSY  1      F                                   DEFIN221
+                                    204NSAT   1      I                                   DEFIN222
+                                    205FSAT   1      I                                   DEFIN223
+                                    206LSAT   1      I                                   DEFIN224
+                                    207STAT   1      I                                   DEFIN225
+                                    208NFUP   1      I                                   DEFIN226
+                                    209TGOSY  1      F                                   DEFIN227
+                                    210SYLF   1      F                                   DEFIN228
+                                    211XSYLF  1      F                                   DEFIN229
+                                    212NSYLF  1      F                                   DEFIN230
+                                    213BEGSY  1      F                                   DEFIN231
+                                    214HALSY  1      F                                   DEFIN232
+                                    215TLASY  1     SF                                   DEFIN233
+                                    216SDTSY  1      F                                   DEFIN234
+                                    217PERSY  1      F                                   DEFIN235
+                                    218X200   1      F                                   DEFIN236
+                                    219N200   1      F                                   DEFIN237
+                                    220DNTSY  1      F                                   DEFIN238
+                                    221C208   1      F                                   DEFIN239
+                                    222X208   1      F                                   DEFIN240
+                                    223N208   1      F                                   DEFIN241
+                                                                                         DEFIN242
+                                    230SYOR3  0      IC                                  DEFIN243
+                                    231ITSAT  1      I                                   DEFIN244
+                                    232ITSYS  1      I                                   DEFIN245
+                                    233SSTAT  1      I                                   DEFIN246
+                                    234PHASE  1     SF                                   DEFIN247
+                                    235ATIME  1      F                                   DEFIN248
+                                    236DTIME  1      F                                   DEFIN249
+                                    237MARKS  1      I                                   DEFIN250
+                                    238MARKU  1      I                                   DEFIN251
```

```
+                                                239MARKD  1     I                                DEFIN252
+                                                240LFSAI  1     F                                DEFIN253
+                                                241SUMSL  1     F                                DEFIN254
+                                                242MAXSL  1     F                                DEFIN255
+                                                243MINSL  1     F                                DEFIN256
+                                                244IGO    1     F                                DEFIN257
+                                                245SEGST  1     F                                DEFIN258
+                                                246HALST  1     F                                DEFIN259
+                                                247TLAST  1     SF                               DEFIN260
+                                                248SOTST  1     F                                DEFIN261
+                                                249PERST  1     F                                DEFIN262
+                                                250X216   1     F                                DEFIN263
+                                                251N216   1     F                                DEFIN264
+                                                252DNIST  1     F                                DEFIN265
+                                                253C223   1     F                                DEFIN266
+                                                254X223   1     F                                DEFIN267
+                                                255N223   1     F                                DEFIN268
+                                                256SAILF  1     F                                DEFIN269
+                                                257S227   1     F                                DEFIN270
+                                                258X227   1     F                                DEFIN271
+                                                259N227   1     F                                DEFIN272
+                                                260NPOS   1     I                                DEFIN273
+                                                261NDEP   1     I                                DEFIN274
+T MODSY8          T SMOD    11/2 I262FMOD   1     I    MOD 1 F                                  DEFIN275
+                  T NOMOD   12/2 I263LMOD   1     I                                             DEFIN276
+                  T EFAIL   21/2 I                                                              DEFIN277
+                  T NUM     23/4 I                                                              DEFIN278
+                  T NRU     24/4 I                                                              DEFIN279
+                  T MAXNU   31/4 I                                                              DEFIN280
+                  T MINNU   32/4 I                                                              DEFIN281
+                  T MSTAT   33/4 I                                                              DEFIN282
+                  T SUMNU   34/4 I                                                              DEFIN283
+                  T LOADF   41/3 I                                                              DEFIN284
+                  T SUMLF   42/3 I                                                              DEFIN285
+                  T MAXLF   43/3 I                                                              DEFIN286
+                  T MINLF   51/3 I                                                              DEFIN287
+                  T EOO     52/3 I                                                              DEFIN288
+                  T EWARN   61/2 I                                                              DEFIN289
+                                                270NVS    0     IC                               DEFIN290
+                                                271CVA    1     F                                DEFIN291
+                                                272TCVA   1     F                                DEFIN292
+                                                273XCVA   1     F                                DEFIN293
+                                                274MCVA   1     F                                DEFIN294
+                                                275VDATE  1     F                                DEFIN295
+                                                276YTD    1     F                                DEFIN296
+                                                277XTD    1     F                                DEFIN297
+                                                278MTD    1     F                                DEFIN298
+                                                                                                 DEFIN299
+                                                280CTUG   1     F                                DEFIN300
+                                                281WTUG   1     F                                DEFIN301
```

```
+                                282CSHUT  1     F                                DEFIN302
+                                283WSHUT  1     F                                DEFIN303
+                                284CSEPS  1     F                                DEFIN304
+                                285WSEPS  1     F                                DEFIN305
      EVENTS                                                                     EVENTS  2
      1 EXOGENOUS                                                                EVENTS  3
          BEGIN (1)                                                              EVENTS  4
      15 ENDOGENOUS                                                              EVENTS  5
          START                                                                  EVENTS  6
          TERM                                                                   EVENTS  7
          NWSAT                                                                  EVENTS  8
          WARN                                                                   EVENTS  9
          FAIL                                                                   EVENTS10
          LAUNC                                                                  EVENTS11
          ARRIV                                                                  EVENTS12
          REFVE                                                                  EVENTS13
          REFMO                                                                  EVENTS14
          REFSA                                                                  EVENTS15
          REMOV                                                                  EVENTS16
          RETRI                                                                  EVENTS17
          SATDN                                                                  EVENTS18
          NEWME                                                                  EVENTS19
          BACK                                                                   EVENTS20
       END                                                                       EVENTS21
       SUBROUTINE ADMOD (IS,IM)                                                 ADMOD   2
C                                                                               ADMOD   3
CC     THIS ROUTINE CREATES THE FAILURE AND WARNING OF A MODULE                 ADMOD   4
C                                                                               ADMOD   5
       LET EDO(IM) = 0                                                          ADMOD   6
       LET MSIAT(IM) =    2                                                     ADMOD   7
       IF TIME GT TIMES, RETURN                                                 ADMOD   8
       LET I = NOMOD(IM)                                                        ADMOD   9
       CALL WEIBUL(ALPW(I),BETAW(I),TW,ALPF(I),BETAF(I),TF)                     ADMOD  10
C                                                                               ADMOD  11
CC  CAUSE WARNINGS                                                              ADMOD  12
C                                                                               ADMOD  13
       LET IEW = EWARN(IM)                                                      ADMOD  14
       IF IEW EQ 0, GO TO 2                                                     ADMOD  15
       IF TIMEF(IEW) NE 0., CANCEL WARN CALLED IEW                              ADMOD  16
       DESTROY WARN CALLED IEW                                                  ADMOD  17
     2 LET IEW = 0                                                              ADMOD  18
       IF TW EQ 0., GO TO 5                                                     ADMOD  19
       LET TX = TTFMO(I)-WMODU                                                  ADMOD  20
       IF TW GT TX, LET TW = TX                                                 ADMOD  21
       IF TIME + TW GT TGO(IS), GO TO 5                                         ADMOD  22
       CREATE WARN CALLED IEW                                                   ADMOD  23
       LET PSAT(IEW) = IS                                                       ADMOD  24
       LET PMOD(IEW) = IM                                                       ADMOD  25
       LET TIMEA(IEW) = ATIME(IS)                                               ADMOD  26
       CAUSE WARN CALLED IEW AT TIME + TW                                       ADMOD  27
```

```
C                                                                       ADMOD  28
CC  CAUSE FAILURES                                                      ADMOD  29
C                                                                       ADMOD  30
    5 LET EWARN(IM) = IEW                                               ADMOD  31
      LET IEF = EFAIL(IM)                                               ADMOD  32
      IF IF GT TTEMO(I), LET IF = TTEMO(I)                              ADMOD  33
      IF IEF EQ 0, GO TO 6                                              ADMOD  34
      IF TIMEF(IEF) NE 0., CANCEL FAIL CALLED IEF                       ADMOD  35
      DESTROY FAIL CALLED IEF                                           ADMOD  36
    6 LET IEF = 0                                                       ADMOD  37
      IF TF EQ 0., GO TO 10                                             ADMOD  38
      IF TIME + TF GT TGO(IS), GO TO 10                                 ADMOD  39
      CREATE FAIL CALLED IEF                                            ADMOD  40
      LET PSAT(IEF) = IS                                                ADMOD  41
      LET PMOD(IEF) = IM                                                ADMOD  42
      LET TIMEA(IEF) = ATIME(IS)                                        ADMOD  43
      CAUSE FAIL CALLED IEF AT TIME + TF                                ADMOD  44
   10 LET EFAIL(IM) = IEF                                               ADMOD  45
      RETURN                                                            ADMOD  46
      END                                                               ADMOD  47
      ENDOGENOUS EVENT ARRIV                                            ARRIV   2
C                                                                       ARRIV   3
CC    THIS IS THE ARRIVAL OF A SATELLITE IN ORBIT AFTER TIME OF FLIGHT. ARRIV   4
CC                                                                      ARRIV   5
CC    NOW ACTIVATE NEW SATELLITES                                       ARRIV   6
CC                                                                      ARRIV   7
CC         ATTEMPT TO REACTIVATE SATELLITES WITH REPLACED MODULES       ARRIV   8
C                                                                       ARRIV   9
      LET IS = PSAT(ARRIV)                                              ARRIV  10
      LET IM = PMOD(ARRIV)                                              ARRIV  11
      DESTROY ARRIV                                                     ARRIV  12
      IF IM NE 0, GO TO 100                                             ARRIV  13
      LET JST = ITSAT(IS)                                               ARRIV  14
      LET JSY = ITSYS(IS)                                               ARRIV  15
      LET NDEP(IS) = NDEP(IS) + 1                                       ARRIV  16
      LET NPOS(IS) = NPOS(IS) + 1                                       ARRIV  17
      IF BEGST(IS) EQ 0., LET BEGST(IS) = TIME                          ARRIV  18
      IF TLAST(IS) EQ 0., LET TLAST(IS) = -TIME                         ARRIV  19
      IF BEGSY(JSY) EQ 0., LET BEGSY(JSY) = TIME                        ARRIV  20
      IF TLASY(JSY) EQ 0., LET TLASY(JSY) = -TIME                       ARRIV  21
      CALL STATUS(IS,0,2)                                               ARRIV  22
      LET AITME(IS) = TIME                                              ARRIV  23
      LET DTIME(IS) = TIME                                              ARRIV  24
      IF TGOSY(JSY) EQ 0., LET TGOSY(JSY) = TIME +                      ARRIV  25
     * TTSYS(JSY)                                                       ARRIV  26
      IF TGOSY(JSY) GT TIMES, LET TGOSY(JSY)= TIMES                     ARRIV  27
      LET TGO(IS) = TIME + TTSAT(JST)                                   ARRIV  28
      IF TGO(IS) GT TGOSY(JSY), LET TGO(IS) =                           ARRIV  29
     * TGOSY(JSY)                                                       ARRIV  30
      CALL ADMOD(IS,MODSY), FOR ALL MODSY IN MOD( , IS)                 ARRIV  31
```

```
      LET IPOL = POLON(JST)                                          ARRIV 32
      IF IPOL EQ 0, GO TO 200                                        ARRIV 33
      LET T = TIME + TTSAT(JST) + WAIT1                              ARRIV 34
      CALL SAVER(T,IS)                                               ARRIV 35
C                                                                    ARRIV 36
CCC   SCHEDULE SATELLITE EVENT (SATON) AT TERMINATION TIME           ARRIV 37
C                                                                    ARRIV 38
  200 IF MARKS(IS) EQ 0, GO TO 1                                     ARRIV 39
      CANCEL SATON CALLED MARKS(IS)                                  ARRIV 40
      DESTROY SATON CALLED MARKS(IS)                                 ARRIV 41
      LET MARKS(IS) = 0                                              ARRIV 42
    1 LET T = TIME + TTSAT(JST)                                      ARRIV 43
      IF SORTE(TTSAT(IS)) NE 0., RETURN                              ARRIV 44
      IF T GT TGO(IS), LET T = TGO(IS)                               ARRIV 45
      IF T LT TIME, RETURN                                           ARRIV 46
      CREATE SATON CALLED MARKS(IS)                                  ARRIV 47
      LET PSAT(MARKS(IS)) = IS                                       ARRIV 48
      CAUSE SATON CALLED MARKS(IS) AT T                              ARRIV 49
      RETURN                                                         ARRIV 50
C                                                                    ARRIV 51
CCC   SINGLE MODULE IS REPLACED IN ORBIT                             ARRIV 52
C                                                                    ARRIV 53
  100 IF SSTAT(IS) EQ OUT, RETURN                                    ARRIV 54
      CALL ADMOD(IS,IM)                                              ARRIV 55
      CALL STATUS(IS,IM,2)                                           ARRIV 56
      LET MDCNT(NOMOD(IM)) = MDCNT(NOMOD(IM)) + 1                    ARRIV 57
      RETURN                                                         ARRIV 58
      END                                                            ARRIV 59
      ENDOGENOUS EVENT BACK                                          BACK   2
C                                                                    BACK   3
CCC   WHEN THIS EVENT OCCURS, THE SATELLITE IS REMOVED FROM ORBIT    BACK   4
C                                                                    BACK   5
      CALL STATUS(PSAT(BACK),0,6)                                    BACK   6
      DESTROY BACK                                                   BACK   7
      RETURN                                                         BACK   8
      END                                                            BACK   9
      EXOGENOUS EVENT BEGIN                                          BEGIN  2
      SAVE                                                           BEGIN  3
      READ TIMEB,TIMES                                               BEGIN  4
      FORMAT(2M5.2.2)                                                BEGIN  5
      CREATE START                                                   BEGIN  6
      CAUSE START AT 1.                                              BEGIN  7
      CALL LOAT                                                      BEGIN  8
C                                                                    BEGIN  9
CCC   INITIALIZATION                                                 BEGIN 10
C                                                                    BEGIN 11
      LET TREFT = TREFT/360.                                         BEGIN 12
      LET SREFT = SREFT/360.                                         BEGIN 13
      LET PREFT = PREFT/360.                                         BEGIN 14
      LET WAIT3 = WAIT3/360.                                         BEGIN 15
```

```
      LET PADT  = PADT /360.                                                BEGIN 16
      LET WAIT1 = WAIT1/360.                                                BEGIN 17
      LET WAIT2 = WAIT2/360.                                                BEGIN 18
      LET WSATU = WSATU/360.                                                BEGIN 19
      LET WSATN = WSATN/360.                                                BEGIN 20
      LET WMODU = WMODU/360.                                                BEGIN 21
      LET WMODN = WMODN/360.                                                BEGIN 22
      LET NTFLT = 1000                                                      BEGIN 23
      LET NFSEP = 1000                                                      BEGIN 24
      LET NFSUI = 1000                                                      BEGIN 25
      LET MIN39(I) = 1000, FOR I=(1)(NYEAR)                                 BEGIN 26
      LET MIN86(I) = 1000, FOR I=(1)(NYEAR)                                 BEGIN 27
      LET MIN90(I) = 1000, FOR I=(1)(NYEAR)                                 BEGIN 28
      LET MINSL(I) =1000. FOR I=(1)(SYORB)                                  BEGIN 29
      LET N227(I) = 1000, FOR I=(1)(SYORB)                                  BEGIN 30
      LET N208(I) = 1000., FOR I=(1)(STSTB)                                 BEGIN 31
      LET N200(I) = 1000., FOR I=(1)(STSTB)                                 BEGIN 32
      LET N223(I) = 1000., FOR I=(1)(SYORB)                                 BEGIN 33
      LET N216(I) = 1000., FOR I=(1)(SYORB)                                 BEGIN 34
      LET N121(I) = 1000, FOR I=(1)(MITAB)                                  BEGIN 35
      LET N125(I) = 1000, FOR I=(1)(MITAB)                                  BEGIN 36
      LET N129(I) = 1000. FOR I=(1)(MITAB)                                  BEGIN 37
      LET MTD(I) = 1000.,FOR I=(1)(3)                                       BEGIN 38
      LET MCVA(I) = 1000., FOR I=(1)(3)                                     BEGIN 39
      RETURN                                                                BEGIN 40
      END                                                                   BEGIN 41
      SUBROUTINE CSPAY                                                      CSPAY  2
C                                                                           CSPAY  3
CC    COMPUTE LAUNCH STATISTICS FOR PAYLOADS                                CSPAY  4
C                                                                           CSPAY  5
      LET B = 0.                                                            CSPAY  6
      DO TO 11, FOR I=(1)(NL(IORB))                                         CSPAY  7
      LET B = B + PAYWT(ILOAD(I))                                           CSPAY  8
      IF IMOD(ILOAD(I)) EQ 0, GO TO 11                                      CSPAY  9
      LET NX = IMOD(ILOAD(I))                                               CSPAY 10
      LET NUM(NX) = NUM(NX) + 1                                             CSPAY 11
   11 LOOP                                                                  CSPAY 12
      LET NMD = ANMD(IORB)                                                  CSPAY 13
      LET SU = (NMD+NINSU-1)/NINSU                                          CSPAY 14
      IF SU EQ 0., GO TO 13                                                 CSPAY 15
      LET X = SU*WTSU/ANMD(IORB)                                            CSPAY 16
      LET B = B + SU*WTSU                                                   CSPAY 17
      DO TO 12,       FOR I=(1)(NL(IORB))                                   CSPAY 18
      IF IMOD(ILOAD(I)) NE 0, LET PAYWT(ILOAD(I)) = PAYWT(ILOAD(I)) + X     CSPAY 19
      LOOP                                                                  CSPAY 20
   13 DO TO 14, FOR J=(1)(NL(IORB))                                         CSPAY 21
      LET NX = ISAT(ILOAD(J))                                               CSPAY 22
      LET NY = IMOD(ILOAD(J))                                               CSPAY 23
      LET LFSAT(NX) = LFSAT(NX) + PAYWT(ILOAD(J))/B                         CSPAY 24
      IF NY EQ 0, LET SATLF(NX) = SATLF(NX)+ PAYWT(ILOAD(J))/B              CSPAY 25
```

```
      IF NY EQ 0, GO TO 14                                                CSPAY 26
      LET M = 100.*PAYWT(ILOAD(J))/B + .5                                 CSPAY 27
      LET LOADF(NY) = LOADF(NY) + M                                       CSPAY 28
   14 LOOP                                                                CSPAY 29
      RETURN                                                              CSPAY 30
      END                                                                 CSPAY 31
      SUBROUTINE DROPQ(J,IO)                                              DROPQ  2
C                                                                         DROPQ  3
CCC   DROP PAYLOAD J FROM LOAD QUEUE ORB(IO)                              DROPQ  4
C                                                                         DROPQ  5
      DO TO 5, FOR ALL ITORB IN ORB(IO)                                   DROPQ  6
      IF PITEM(ITORB) NE J, GO TO 5                                       DROPQ  7
      REMOVE J FROM LOAD                                                  DROPQ  8
      DESTROY PAYLO CALLED J                                              DROPQ  9
      REMOVE ITORB FROM ORB(IO)                                           DROPQ 10
      DESTROY ITORB                                                       DROPQ 11
      RETURN                                                              DROPQ 12
    5 LOOP                                                                DROPQ 13
      RETURN                                                              DROPQ 14
      END                                                                 DROPQ 15
      ENDOGENOUS EVENT FAIL                                               FAIL   2
C                                                                         FAIL   3
CCCCC THIS ROUTINE WILL MARK OUTAGE OF A SATELLITE AND NOTE WHICH MODULE  FAIL   4
C                                                                         FAIL   5
CCCCC IS OUT (MAYBE MORE THAN ONE).                                       FAIL   6
C                                                                         FAIL   7
      LET IS = PSAT (FAIL)                                                FAIL   8
      LET IM = PMOD (FAIL)                                                FAIL   9
      LET T = TIMEA(FAIL)                                                 FAIL  10
      DESTROY FAIL                                                        FAIL  11
      LET EFAIL(IM) = 0                                                   FAIL  12
C                                                                         FAIL  13
CCC   BLOCK FAILURE EVENT (FOR LAUNCH) IF MODULE IS NOT REPLACEABLE       FAIL  14
C                                                                         FAIL  15
      IF SSTAT(IS) EQ OUT, RETURN                                         FAIL  16
      IF T LT ATIME(IS), RETURN                                           FAIL  17
      CALL STATUS(IS,IM,3)                                                FAIL  18
      LET NOFAL(NOMOD(IM)) = NOFAL(NOMOD(IM)) + 1                         FAIL  19
      IF SSTAT(IS) EQ OUT, RETURN                                         FAIL  20
      LET DELAY = WMODN                                                   FAIL  21
C                                                                         FAIL  22
CCC   BLOCK EVENT AFTER TIMES                                             FAIL  23
C                                                                         FAIL  24
      IF EWARN(IM) NE 0, RETURN                                           FAIL  25
      IF TIME + DELAY GT TGO(IS), RETURN                                  FAIL  26
      CALL SHIP(IS,IM)                                                    FAIL  27
      CALL REDUN(IS,IM)                                                   FAIL  28
      IF DELTA GT 0., RETURN                                              FAIL  29
      CREATE LAUNC                                                        FAIL  30
      LET PSAT(LAUNC) = IS                                                FAIL  31
```

```
      LET PMOD(LAUNC) = IM                                                FAIL  32
      CAUSE LAUNC AT TIME + DELAY                                         FAIL  33
      RETURN                                                              FAIL  34
      END                                                                 FAIL  35
      SUBROUTINE FILEO                                                    FILEO  2
C                                                                         FILEO  3
C  OUTPUT SATELLITE STATUS SUMMARY REPORT                                 FILEO  4
C                                                                         FILEO  5
      LET TRIG2 = 1                                                       FILEO  6
      WRITE ON 6                                                          FILEO  7
      FORMAT(*1*)                                                         FILEO  8
      DO TO 5, FOR ALL FR IN FRS                                          FILEO  9
      LET TIME = TIMEF(FR)                                                FILEO 10
      LET IS = SATNO(FR)                                                  FILEO 11
      LET I = SATSY(FR)                                                   FILEO 12
      LET NPOS(IS) = NPS(FR)                                              FILEO 13
      IF I EQ 1, LET K = UP                                               FILEO 14
      IF I EQ 2, LET K = DOWN                                             FILEO 15
      IF I EQ 3, LET K = OUT                                              FILEO 16
      LET STAT(ITSYS(IS)) = K                                             FILEO 17
      LET I = ST(FR)                                                      FILEO 18
      IF I EQ 1, LET K = UP                                               FILEO 19
      IF I EQ 2, LET K = DOWN                                             FILEO 20
      IF I EQ 3, LET K = OUT                                              FILEO 21
      LET SSTAT(IS) = K                                                   FILEO 22
      IF INOW NE IS, WRITE ON 6                                           FILEO 23
      FORMAT(*0          CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IFILEO 24
     *N ORBIT*/S5,*TIME              SYSTEM    STATUS    SATELLITE STATUS  FILEO 25
     * MODULE     STATUS*)                                                FILEO 26
      LET INOW = IS                                                       FILEO 27
      CALL STATUS(IS,MOONO(FR),NOSTA(FR))                                 FILEO 28
C                                                                         FILEO 29
CC    RELEASE MEMORY                                                      FILEO 30
C                                                                         FILEO 31
      REMOVE FR FROM FRS                                                  FILEO 32
      DESTROY FR                                                          FILEO 33
    5 LOOP                                                                FILEO 34
      LET TRIG2 = 2                                                       FILEO 35
      RETURN                                                              FILEO 36
VINOW 0                                                                   FILEO 37
      END                                                                 FILEO 38
      SUBROUTINE FILES(IS,IM,ISI)                                         FILES  2
C                                                                         FILES  3
CC  FILE SATELLITE SUMMARY IN FR                                          FILES  4
C                                                                         FILES  5
      CREATE FR                                                           FILES  6
      LET TIMEF(FR) = TIME                                                FILES  7
      LET SATNO(FR) = IS                                                  FILES  8
      LET I = STAT(ITSYS(IS))                                             FILES  9
      IF I EQ UP, LET K = 1                                               FILES 10
```

```
      IF I EQ DOWN, LET K = 2                                          FILES 11
      IF I EQ OUT, LET K = 3                                           FILES 12
      LET SATSY(FR) = K                                                FILES 13
      LET I = SSTAT(IS)                                                FILES 14
      IF I EQ UP, LET K = 1                                            FILES 15
      IF I EQ DOWN, LET K = 2                                          FILES 16
      IF I EQ OUT, LET K = 3                                           FILES 17
      LET ST(FR) = K                                                   FILES 18
      LET MODNO(FR) = IM                                               FILES 19
      LET NOSTA(FR) = IST                                              FILES 20
      LET NPS(FR) = NPOS(IS)                                           FILES 21
      FILE FR IN FRS                                                   FILES 22
      RETURN                                                           FILES 23
      END                                                              FILES 24
      SUBROUTINE GETV(IGO)                                             GETV   2
C                                                                      GETV   3
CCC  FIND NECESSARY VEHICLES                                           GETV   4
C                                                                      GETV   5
      LET ITUG = 0                                                     GETV   6
      LET ISEPS = 0                                                    GETV   7
      LET IGO = 0                                                      GETV   8
      LET ISHUT = 0                                                    GETV   9
      DO TO 5, FOR I=(1)(NSHUT)                                        GETV  10
      IF VSHUT(I) GT 0, LET ISHUT = I                                  GETV  11
    5 LOOP                                                             GETV  12
      IF ISHUT EQ 0, GO TO 20                                          GETV  13
      IF RQUP(IORB) EQ 0, RETURN                                       GETV  14
      DO TO 10, FOR I=(1)(NTUG)                                        GETV  15
      IF VTUG(I) GT 0, LET ITUG = I                                    GETV  16
   10 LOOP                                                             GETV  17
      IF ITUG EQ 0, GO TO 25                                           GETV  18
      IF RQSEP(IORB) EQ 0, RETURN                                      GETV  19
      DO TO 15, FOR I=(1)(NSEPS)                                       GETV  20
      IF VSEPS(I) GT 0, LET ISEPS = I                                  GETV  21
   15 LOOP                                                             GETV  22
      IF ISEPS EQ 0, GO TO 30                                          GETV  23
      RETURN                                                           GETV  24
   20 LET IGO = 1                                                      GETV  25
      RETURN                                                           GETV  26
   25 LET IGO = 2                                                      GETV  27
      RETURN                                                           GETV  28
   30 LET IGO = 3                                                      GETV  29
      RETURN                                                           GETV  30
      END                                                              GETV  31
      SUBROUTINE ISSUE                                                 ISSUE  2
C                                                                      ISSUE  3
CC   SET LAUNCH AND FLIGHT SEQUENCE EVENTS                             ISSUE  4
C                                                                      ISSUE  5
      DO TO 5, FOR I=(1)(3)                                            ISSUE  6
      IF VDATE(I) EQ 0., GO TO 5                                       ISSUE  7
```

```
      LET VDATE(I) = VDATE(I) + TIME                                    ISSUE  8
      IF VDATE(I) LT 0., GO TO 5                                        ISSUE  9
      LET VTD(I) = VTD(I) + VDATE(I)                                    ISSUE 10
      IF VDATE(I) GT XTD(I), LET XTD(I) = VDATE(I)                      ISSUE 11
      IF VDATE(I) LT MTD(I), LET MTD(I) = VDATE(I)                      ISSUE 12
      LET VDATE(I) = 0.                                                 ISSUE 13
    5 LOOP                                                              ISSUE 14
      LET FLYT = ORBTM(IORB)                                            ISSUE 15
      LET ILOAD(1) = PQUE(IORB)                                         ISSUE 16
      LET NQ = NL(IORB)                                                 ISSUE 17
      LET ILOAD(J+1) = CITEM(ILOAD(J)), FOR J=(1)(NQ-1)                 ISSUE 18
      LET NMD = ANMD(IORB)                                              ISSUE 19
      LET SU = (NMD+NINSU-1)/NINSU                                      ISSUE 20
      LET WGH = SU*WISU                                                 ISSUE 21
      LET WLEN = SU*LENSU                                               ISSUE 22
      DO TO 10, FOR I=(1)(NQ)                                           ISSUE 23
      IF IRT(ILOAD(I)) NE 0, GO TO 10                                   ISSUE 24
      LET WGH = WGH + PAYWT(ILOAD(I))                                   ISSUE 25
      LET WLEN = WLEN + PAYLN(ILOAD(I))                                 ISSUE 26
   10 LOOP                                                              ISSUE 27
      IF TRIG EQ 0, WRITE ON 6,ISHUT,ITUG,ISEPS,WGH,WLEN                ISSUE 28
      FORMAT(S5,*--LAUNCH NOW -- SHUTTLE NO.*,I4,* TUG NO.*,I4,* SEPS NOISSUE 29
     *.*,I4,* WEIGHT = *,D6,* LENGTH = *,D6.1*----------*)              ISSUE 30
      IF TIME GT TIMEB, CALL CSPAY                                      ISSUE 31
      DO TO 17, FOR J=(1)(NL(IORB))                                     ISSUE 32
      LET NX  = ISAT(ILOAD(J))                                          ISSUE 33
      LET NM = IMOD(ILOAD(J))                                           ISSUE 34
      IF SORTE(ITSAT(NX)) NE 0., LET FLYT = SORTE(ITSAT(NX))            ISSUE 35
      IF IRT(ILOAD(J)) NE 0, GO TO 16                                   ISSUE 36
      CALL STATUS(NX,NM,4)                                              ISSUE 37
      CREATE ARRIV                                                      ISSUE 38
      LET PSAT(ARRIV) = ISAT(ILOAD(J))                                  ISSUE 39
      LET PMOD(ARRIV) = IMOD(ILOAD(J))                                  ISSUE 40
      CAUSE ARRIV AT TIME + PADT + GOTIM(ILOAD(J))                      ISSUE 41
      IF SORTE(ITSAT(NX)) EQ 0., GO TO 15                               ISSUE 42
      LET GOTIM(ILOAD(J)) = SORTE(ITSAT(NX))                            ISSUE 43
   16 IF SORTE(ITSAT(NX)) NE 0., GO TO 160                              ISSUE 44
      CREATE BACK                                                       ISSUE 45
      LET PSAT(BACK) = NX                                               ISSUE 46
      CAUSE BACK AT TIME + PADT + FLYT                                  ISSUE 47
  160 CREATE REMOV                                                      ISSUE 48
      LET PSAT(REMOV) = NX                                              ISSUE 49
      CAUSE REMOV AT TIME + PADT + FLYT - 6./8640.                      ISSUE 50
   15 CALL OROPQ(ILOAD(J),IORB)                                         ISSUE 51
   17 LOOP                                                              ISSUE 52
      CREATE REFVE                                                      ISSUE 53
      LET VNAME(REFVE) = SHUT                                           ISSUE 54
      LET PMOD(REFVE) = ISHUT                                           ISSUE 55
      CAUSE REFVE AT TIME + PADT + SREFT + FLYT                         ISSUE 56
      LET VSHUT(ISHUT) = 0                                              ISSUE 57
```

```
      LET I = TIME - TIMEB + 1.                                          ISSUE  58
      IF I LE 0, GO TO 20                                                ISSUE  59
      LET SUTFY(I) = SUTFY(I) + 1                                        ISSUE  60
      IF ITUG NE 0, GO TO 20                                             ISSUE  61
      LET CSHUT(IORB) = CSHUT(IORB) + 1.                                 ISSUE  62
      LET WSHUT(IORB) = WSHUT(IORB) + WGH                                ISSUE  63
   20 IF ITUG EQ 0, GO TO 18                                             ISSUE  64
      CREATE REFVE                                                       ISSUE  65
      LET VNAME(REFVE) = TUG                                             ISSUE  66
      LET PMOD(REFVE) = ITUG                                             ISSUE  67
      CAUSE REFVE AT TIME + PADT + TREFT + FLYT                          ISSUE  68
      LET VTUG(ITUG) = 0                                                 ISSUE  69
        IF I LE 0, GO TO 18                                              ISSUE  70
      LET TUGFY(I) = TUGFY(I) + 1                                        ISSUE  71
      IF ISEPS NE 0, GO TO 18                                            ISSUE  72
      LET EXVEH = EXORB(IORB)                                            ISSUE  73
      LET EXORB(IORB) = 0                                                ISSUE  74
      IF EXVEH EQ 0, LET EXVEH = EXPV(ROUP(IORB))                        ISSUE  75
      IF EXVEH NE 0, LET EXTUG = EXTUG + 1.                              ISSUE  76
      LET WTUG(IORB) = WTUG(IORB) + WGH                                  ISSUE  77
      LET CTUG(IORB) = CTUG(IORB) + 1.                                   ISSUE  78
   18 IF ISEPS EQ 0, GO TO 19                                            ISSUE  79
      CREATE REFVE                                                       ISSUE  80
      LET VNAME(REFVE) = SEPS                                            ISSUE  81
      LET PMOD(REFVE) = ISEPS                                            ISSUE  82
      CAUSE REFVE AT TIME + PADT + PREFT + FLYT                          ISSUE  83
      LET VSEPS(ISEPS) = 0                                               ISSUE  84
      LET MSEP = 0                                                       ISSUE  85
      LET SEPFY(I) = SEPFY(I) +1                                         ISSUE  86
      LET CSEPS(IORB) = CSEPS(IORB) + 1.                                 ISSUE  87
      LET WSEPS(IORB) = WSEPS(IORB) + WGH                                ISSUE  88
   19 IF TRIG EQ 0, WRITE ON 6                                           ISSUE  89
      FORMAT(S5,*-------------------------------------------------------ISSUE  90
     *-----------------------------------------------*)                  ISSUE  91
      LET NL(IORB) = 0                                                   ISSUE  92
      RETURN                                                             ISSUE  93
      END                                                                ISSUE  94
      ENDOGENOUS EVENT LAUNC                                             LAUNC   2
C                                                                        LAUNC   3
C     THIS EVENT OCCURS WITH AN ACTUAL LAUNCH SCHEDULED WITH DELAYS.     LAUNC   4
C                                                                        LAUNC   5
C     IT SCHEDULES ARRIVAL IN ORBIT, VEHICLE REFURB CYCLE, MODULE AND    LAUNC   6
C                                                                        LAUNC   7
C     SATELLITE RETRIEVAL WITH REFURB CYCLE                              LAUNC   8
C                                                                        LAUNC   9
C     PREDICT ABORTED LAUNCHES AND LOST PAYLOADS                         LAUNC  10
C                                                                        LAUNC  11
      LET IS = PSAT(LAUNC)                                               LAUNC  12
      LET IM = PMOD(LAUNC)                                               LAUNC  13
      DESTROY LAUNC                                                      LAUNC  14
```

```
      LET IORB = ORBIT(ITSAT(IS))                                    LAUNC 15
      IF ORB(IORB) IS EMPTY, RETURN                                  LAUNC 16
      DO TO 5, FOR ALL ITORB IN ORB(IORB)                            LAUNC 17
      IF IS NE ISAT(PITEM(ITORB)), GO TO 5                           LAUNC 18
      IF IM NE IMOD(PITEM(ITORB)), GO TO 5                           LAUNC 19
      CALL GETV(IGO)                                                 LAUNC 20
      IF W(IORB) GT 0., LET W(IORB) = -W(IORB)                       LAUNC 21
      IF RQSEP(IORB) NE 0, GO TO 15                                  LAUNC 22
      IF IGO NE 0, GO TO 10                                          LAUNC 23
      CALL SHIP(0,0)                                                 LAUNC 24
      RETURN                                                         LAUNC 25
    5 LOOP                                                           LAUNC 26
      RETURN                                                         LAUNC 27
   10 IF TRIG EQ 0, WRITE ON 6,TIME                                  LAUNC 28
      FORMAT(S5,M5.2.2,S60,*PAYLOAD DUE TO GO - NO VEHICLE*)         LAUNC 29
      LET CVA(IGO) = CVA(IGO) + 1.                                   LAUNC 30
      LET VDATE(IGO) = VDATE(IGO) - TIME                             LAUNC 31
      RETURN                                                         LAUNC 32
   15 IF IGO EQ 1, RETURN                                            LAUNC 33
      IF IGO EQ 2, RETURN                                            LAUNC 34
      LET I = RQSEP(IORB)                                            LAUNC 35
      IF IGO EQ 3, LET RQSEP(IORB) = 0                               LAUNC 36
      LET NL(IORB) = 0                                               LAUNC 37
      CALL SHIP(0,0)                                                 LAUNC 38
      LET RQSEP(IORB) = I                                            LAUNC 39
      RETURN                                                         LAUNC 40
      END                                                            LAUNC 41
      SUBROUTINE LDAT                                                LDAT   2
C                                                                    LDAT   3
C  LOAD DATA SUBROUTINE                                              LDAT   4
C                                                                    LDAT   5
      WRITE ON 6                                                     LDAT   6
      FORMAT(*1      INPUT DATA*//)                                  LDAT   7
      LET IRFLG = 0                                                  LDAT   8
      CALL LDVEH(IRFLG)                                              LDAT   9
      CALL LDORB(IRFLG)                                              LDAT  10
      CALL LDMOD(IRFLG)                                              LDAT  11
      CALL LDSAT(IRFLG)                                              LDAT  12
      CALL LDSYS(IRFLG)                                              LDAT  13
      CALL LDSCH(IRFLG)                                              LDAT  14
      CALL LDME(IRFLG)                                               LDAT  15
      CALL LDPUR                                                     LDAT  16
      IF IRFLG EQ 0, RETURN                                          LDAT  17
      WRITE ON 6                                                     LDAT  18
      FORMAT(*0-------- RUN STOPPED DUE TO DATA ERROR --------*)     LDAT  19
      STOP                                                           LDAT  20
      END                                                            LDAT  21
      SUBROUTINE LDME(IRFLG)                                         LDME   2
C                                                                    LDME   3
C     MISSION EQUIPMENT UPGRADE INPUT ROUTINE                        LDME   4
```

```
C                                                                       LOME    5
      DIMENSION IA(5),A(4)                                              LOME    6
      WRITE ON 6                                                        LOME    7
      FORMAT(*  ME UPGRADE SCHEDULES INPUT *)                           LOME    8
C                                                                       LOME    9
C   LOAD MISSION EQUIPMENT UPGRADE SCHEDULE                             LOME   10
C                                                                       LOME   11
  100 READ FROM 5,IA(1),IA(2),IA(3),IA(4),B,IA(5)                       LOME   12
      FORMAT(A6,I4,A6,I4,M4.2.2,A6)                                     LOME   13
C                                                                       LOME   14
C     PRINT SCHEDULES                                                   LOME   15
C                                                                       LOME   16
      WRITE ON 6,IA(1),IA(2),IA(3),IA(4),B,IA(5)                        LOME   17
      FORMAT(S10,A6,I6,S3,A6,I6,S3,M4.2.2,S3,A6)                        LOME   18
            IF IA(1) EQ BLANK, GO TO 200                                LOME   19
      LET MEOLD = 0                                                     LOME   20
      LET MENEW = 0                                                     LOME   21
      DO TO 110, FOR I=(1)(MIIAB)                                       LOME   22
      IF IA(3) EQ MNAME(I), LET MEOLD = I                               LOME   23
      IF IA(5) EQ MNAME(I), LET MENEW = I                               LOME   24
  110 LOOP                                                              LOME   25
      IF MEOLD + MENEW NE 0, GO TO 115                                  LOME   26
C                                                                       LOME   27
C     ERROR DETECTED                                                    LOME   28
C                                                                       LOME   29
  111 WRITE ON 6                                                        LOME   30
      FORMAT(* BAD ME DATA - ENTRY REJECTED *)                          LOME   31
      LET RTFLG = 1                                                     LOME   32
      GO TO 100                                                         LOME   33
  115 IF MCLAS(MEOLD) NE ME, GO TO 111                                  LOME   34
      IF MCLAS(MENEW) NE ME, GO TO 111                                  LOME   35
      DO TO 120, FOR I=(1)(STSTB)                                       LOME   36
      IF IA(1) NE SYNAM(I), GO TO 120                                   LOME   37
      LET ISY = I                                                       LOME   38
      GO TO 125                                                         LOME   39
  120 LOOP                                                              LOME   40
      GO TO 111                                                         LOME   41
  125 IF FSAT(ISY) EQ 0, GO TO 111                                      LOME   42
      LET ISY = FSAT(ISY)+IA(2) - 1                                     LOME   43
      IF MOD(ISY) IS EMPTY, GO TO 111                                   LOME   44
      DO TO 130, FOR ALL MODSY IN MOD(ISY)                              LOME   45
      IF NOMOD(MODSY) EQ MEOLD, LET IA(4) = IA(4) - 1                   LOME   46
      IF IA(4) EQ 0, GO TO 135                                          LOME   47
  130 LOOP                                                              LOME   48
      GO TO 111                                                         LOME   49
C                                                                       LOME   50
C  SAVE ME UPGRADE IN MENEW                                             LOME   51
C                                                                       LOME   52
  135 CREATE MESET                                                      LOME   53
      LET PSAT(MESET) = ISY                                             LOME   54
```

```
      LET PMOD(MESET) = NOMOD(MODSY)                                    LDME  55
      LET  MEOI(MESET)= B                                               LDME  56
      LET NOMOD(MESET) = MENEW                                          LDME  57
      FILE MESET IN MES                                                 LDME  58
      GO TO 100                                                         LDME  59
  200 RETURN                                                            LDME  60
UME   ME                                                                LDME  61
      END                                                               LDME  62
      SUBROUTINE LDMOD(IRFLG)                                           LDMOD  2
C                                                                       LDMOD  3
CCC   MODULE INPUT ROUTINE                                              LDMOD  4
C                                                                       LDMOD  5
      READ FROM 5,NUMMOD,FACT                                           LDMOD  6
      FORMAT(I3,01.3)                                                   LDMOD  7
      IF NUMMOD LE MITAB, GO TO 5                                       LDMOD  8
      WRITE ON 6,NUMMOD,MITAB                                           LDMOD  9
      FORMAT(*  ERROR - NUMBER OF MODULES INPUT(*,I6,*) EXCEEDS CAPACITYLDMOD 10
     *(*,I6,*)*)                                                        LDMOD 11
      LET IRFLG = 1                                                     LDMOD 12
      RETURN                                                            LDMOD 13
    5 WRITE ON 6,NUMMOD                                                 LDMOD 14
      FORMAT(I11,* MODULES INPUT*/*     NAME*,S7* ALPHA F   BETA F    TTLDMOD 15
     *      ALPHA W   BETA W    WT        VOL        CLASS*)           LDMOD 16
      DO TO 10, FOR I=(1)(NUMMOD)                                       LDMOD 17
C                                                                       LDMOD 18
CCC   LOAD MODULE DATA                                                  LDMOD 19
C                                                                       LDMOD 20
      READ FROM 5, MNAME(I)                                             LDMOD 21
     * ,ALPF(I),BETAF(I),TTFMD(I),MODWT(I),MDVOL(I),                    LDMOD 22
     *  MCLAS(I)                                                        LDMOD 23
     *,ALPW(I),BETAW(I)                                                 LDMOD 24
     *,R,TAU                                                            LDMOD 25
      FORMAT(A6,S4,2D2.2,D3,D5,D3.1,A6,S3,2D2.2,D1,D2.2)                LDMOD 26
      IF ALPF(I) NE 0., GO TO 7                                         LDMOD 27
      IF R EQ 0., GO TO 7                                               LDMOD 28
      LET BETAF(I) = 1.                                                 LDMOD 29
      LET ALPF(I) = -TAU/ALOG(R)                                        LDMOD 30
    7 IF ALPW(I) EQ 0., LET ALPW(I) = FACT*ALPF(I)                      LDMOD 31
      IF BETAW(I) EQ 0., LET BETAW(I) = 1.                              LDMOD 32
      IF TTFMD(I) EQ 0., LET TTFMD(I)=.5*ALPF(I)                        LDMOD 33
C                                                                       LDMOD 34
CCC   PRINT MODULE DATA                                                 LDMOD 35
C                                                                       LDMOD 36
      WRITE ON 6,MNAME(I),ALPF(I),BETAF(I),TTFMD(I),ALPW(I),BETAW(I),   LDMOD 37
     *  MODWT(I),MDVOL(I),MCLAS(I)                                      LDMOD 38
      FORMAT(S5,A6,S4,7D7.2,S4,A6)                                      LDMOD 39
   10 LOOP                                                              LDMOD 40
      RETURN                                                            LDMOD 41
      END                                                               LDORB  2
      SUBROUTINE LDORB(IRFLG)                                           LDORB  3
C
```

```
C   LOAD ORBIT DATA                                                    LDORB   4
C                                                                      LDORB   5
      READ FROM 5,NORB                                                 LDORB   6
      FORMAT(I3)                                                       LDORB   7
      IF NORB LE NORBS, GO TO 1                                        LDORB   8
      WRITE ON 6,NORB,NORBS                                            LDORB   9
      FORMAT(*  ERROR - NUMBER OF ORBITS INPUT(*,I6,*) EXCEEDS CAPACITY(LDORB  10
     **,I6,*)*)                                                        LDORB  11
      LET IRFLG = 1                                                    LDORB  12
    1 WRITE ON 6,NORB                                                  LDORB  13
      FORMAT(I8,* ORBITS INPUT*)                                       LDORB  14
      WRITE ON 6                                                       LDORB  15
      FORMAT(*    NAME     DV     PERIOD     RA     VC    UPPER  SEPS   SHLDORB  16
     *UTTLE*)                                                          LDORB  17
      DO TO 10, FOR I=(1)(NORB)                                        LDORB  18
      READ FROM 5,ORBID(I),ORBDV(I),ORBPD(I),ORBRA(I),ORBVC(I),RQUP(I), LDORB  19
     * RQSEP(I),RQSUT(I),DV1(I)                                        LDORB  20
      FORMAT(A6,4D5.1,3A6,D5.1)                                        LDORB  21
      WRITE ON 6 ,ORBID(I),ORBDV(I),ORBPD(I),ORBRA(I),ORBVC(I),RQUP(I), LDORB  22
     * RQSEP(I),RQSUT(I),DV1(I)                                        LDORB  23
      FORMAT(S3,A6,4D7.1,S1,A6,S1,A6,S1,A6,D7.1)                       LDORB  24
      DO TO 5, FOR J=(1)(NVEH)                                         LDORB  25
      IF RQUP(I) EQ NAMEV(J), GO TO 9                                  LDORB  26
    5 LOOP                                                             LDORB  27
      LET J = 0                                                        LDORB  28
      IF RQUP(I) EQ BLANK, GO TO 9                                     LDORB  29
      LET IRFLG = 1                                                    LDORB  30
      WRITE ON 6                                                       LDORB  31
      FORMAT(*  NO SUCH UPPER STAGE*)                                  LDORB  32
      GO TO 2                                                          LDORB  33
    9 LET RQUP(I) = J                                                  LDORB  34
    2 DO TO 3, FOR J=(1)(NVEH)                                         LDORB  35
      IF RQSEP(I) EQ NAMEV(J), GO TO 4                                 LDORB  36
    3 LOOP                                                             LDORB  37
      LET J= 0                                                         LDORB  38
      IF RQSEP(I) EQ BLANK, GO TO 4                                    LDORB  39
      LET IRFLG = 1                                                    LDORB  40
      WRITE ON 6                                                       LDORB  41
      FORMAT(*  NO SUCH SEPS VEHICLE FOUND *)                          LDORB  42
      GO TO 6                                                          LDORB  43
    4 LET RQSEP(I) = J                                                 LDORB  44
    6 DO TO 7, FOR J=(1)(NVEH)                                         LDORB  45
      IF RQSUT(I) EQ NAMEV(J), GO TO 8                                 LDORB  46
    7 LOOP                                                             LDORB  47
      LET IRFLG = 1                                                    LDORB  48
      WRITE ON 6                                                       LDORB  49
      FORMAT(*  NO SUCH SHUTTLE FOUND *)                               LDORB  50
      GO TO 10                                                         LDORB  51
    8 LET RQSUT(I) = J                                                 LDORB  52
   10 LOOP                                                             LDORB  53
```

```
      RETURN                                                            LDORB 54
      END                                                               LDORB 55
      SUBROUTINE LDPUR                                                  LDPUR  2
C                                                                       LDPUR  3
CC  PURGE MEMORY OF UNUSED MODULES                                      LDPUR  4
C                                                                       LDPUR  5
      WRITE ON 6                                                        LDPUR  6
      FORMAT(*1      SYNOPSIS OF INPUT*)                                LDPUR  7
      LET K = 0                                                         LDPUR  8
      LET M = 0                                                         LDPUR  9
      DO TO 80, FOR I=(1)(STSTB)                                        LDPUR 10
      LET NSYLF(I) = 1000.                                              LDPUR 11
      LET J = 0                                                         LDPUR 12
      IF FSAT(I) EQ 0, GO TO 80                                         LDPUR 13
      DO TO 79, FOR L=(FSAT(I))(LSAT(I))                                LDPUR 14
      IF MARKS(L) EQ 0, GO TO 79                                        LDPUR 15
      LET MARKS(L) = 0                                                  LDPUR 16
      LET J = 1                                                         LDPUR 17
      LET NMODS(ITSAT(L)) = 1                                           LDPUR 18
      LET MDCNT(NOMOD(MDSAT)) = 1, FOR ALL MDSAT IN MDS(ITSAT(L))       LDPUR 19
   79 LOOP                                                              LDPUR 20
      IF J NE 0, GO TO 78                                               LDPUR 21
      WRITE ON 6,SYNAM(I)                                               LDPUR 22
      FORMAT(* UNUSED SYSTEM - *,A6)                                    LDPUR 23
      LET SYNAM(I) = 0                                                  LDPUR 24
      GO TO 80                                                          LDPUR 25
   78 LET K = K + 1                                                     LDPUR 26
      LET M = M + LSAT(I) - FSAT(I) + 1                                 LDPUR 27
   80 LOOP                                                              LDPUR 28
      LET I = M/4                                                       LDPUR 29
      IF I*4 NE M, LET I = I+1                                          LDPUR 30
      LET M = I*4                                                       LDPUR 31
      WRITE ON 6,M,SYORB                                                LDPUR 32
      FORMAT(* PROBLEM USED *,I3,* SATELLITE/SYSTEM POSITIONS OUT OF AVALDPUR 33
     *ILABLE *,I3)                                                      LDPUR 34
      WRITE ON 6,K,STSTB                                                LDPUR 35
      FORMAT(* PROBLEM USED *,I3,* SYSTEMS OUT OF AVAILABLE *,I3)       LDPUR 36
      LET K=0                                                           LDPUR 37
      DO TO 85, FOR I=(1)(SITAB)                                        LDPUR 38
      IF NMODS(I) NE 0, LET K = K+1                                     LDPUR 39
      IF NMODS(I) NE 0, GO TO 85                                        LDPUR 40
      IF MDS(I) IS EMPTY, GO TO 84                                      LDPUR 41
      DO TO 83, FOR ALL MDSAT IN MDS(I)                                 LDPUR 42
      REMOVE FIRST MDSAT FROM MDS(I)                                    LDPUR 43
      DESTROY MDSAT                                                     LDPUR 44
   83 LOOP                                                              LDPUR 45
   84 IF SNAME(I) EQ 0, GO TO 85                                        LDPUR 46
      WRITE ON 6,SNAME(I)                                               LDPUR 47
      FORMAT(*   UNUSED SATELLITE - *,A6)                              LDPUR 48
      LET SNAME(I) = 0                                                  LDPUR 49
```

```
 85 LOOP                                                                LDPUR  50
    WRITE ON 6,K,SITAB                                                  LDPUR  51
    FORMAT(* PROBLEM USED *,I3,* SATELLITES OUT OF AVAILABLE *,I3)      LDPUR  52
    LET K = 0                                                           LDPUR  53
    DO TO 90, FOR I=(1)(MITAB)                                          LDPUR  54
    IF MNAME(I) EQ 0, GO TO 90                                          LDPUR  55
    IF MDCNT(I) NE 0, LET K = K + 1                                     LDPUR  56
    IF MDCNT(I) EQ 0, WRITE ON 6,MNAME(I)                               LDPUR  57
    FORMAT(*    UNUSED MODULE - *,A6)                                   LDPUR  58
    IF MDCNT(I) EQ 0, LET MNAME(I) = 0                                  LDPUR  59
    LET MDCNT(I) = 0                                                    LDPUR  60
 90 LOOP                                                                LDPUR  61
    DO TO 6, FOR I=(1)(SYORB)                                           LDPUR  62
    IF ITSAT(I) EQ BLANK, GO TO 6                                       LDPUR  63
    IF ITSAT(I) EQ 0, GO TO 6                                           LDPUR  64
    IF MDS(ITSAT(I)) IS EMPTY, GO TO 6                                  LDPUR  65
    DO TO 4, FOR ALL MDSAT IN MDS(ITSAT(I))                             LDPUR  66
    CREATE MODSY                                                        LDPUR  67
    LET NOMOD(MODSY) = NOMOD(MDSAT)                                     LDPUR  68
    LET NUM  (MODSY) = 0                                                LDPUR  69
    LET SUMNU(MODSY) = 0                                                LDPUR  70
    LET MAXNU(MODSY) = 0                                                LDPUR  71
    LET MINNU(MODSY) = 500                                              LDPUR  72
    LET LOADF(MODSY) = 0                                                LDPUR  73
    LET SUMLF(MODSY) = 0                                                LDPUR  74
    LET MAXLF(MODSY) = 0                                                LDPUR  75
    LET MINLF(MODSY) = 1000                                             LDPUR  76
    LET MSTAT(MODSY) = 0                                                LDPUR  77
    LET NRU  (MODSY) = NRU(MDSAT)                                       LDPUR  78
    FILE MODSY IN MOD(I)                                                LDPUR  79
  4 LOOP                                                                LDPUR  80
  6 LOOP                                                                LDPUR  81
    WRITE ON 6,K,MITAB                                                  LDPUR  82
    FORMAT(* PROBLEM USED *,I3,* MODULES OUT OF AVAILABLE *,I3)         LDPUR  83
    RETURN                                                              LDPUR  84
    END                                                                 LDPUR  85
    SUBROUTINE LDSAT(IRFLG)                                             LDSAT   2
C                                                                       LDSAT   3
C   SATELLITE INPUT ROUTINE                                             LDSAT   4
C                                                                       LDSAT   5
    DIMENSION IA(25),MODUL(25)                                          LDSAT   6
    READ FROM 5,NUMSAT                                                  LDSAT   7
    FORMAT(I3)                                                          LDSAT   8
    IF NUMSAT LE SITAB, GO TO 6                                         LDSAT   9
    WRITE ON 6,NUMSAT,SITAB                                             LDSAT  10
    FORMAT(*  ERROR - NUMBER OF SATELLITES INPUT(*,I6,*) EXCEEDS CAPACLDSAT  11
   *ITY(*,I6*)*)                                                        LDSAT  12
    LET IRFLG = 1                                                       LDSAT  13
  6 WRITE ON 6,NUMSAT                                                   LDSAT  14
    FORMAT(/S1,I10,*  SATELLITES INPUT*/*     NAME    HT     VOL     PRIORLDSAT  15
```

```
*     INCLINATION  ORBIT   MODULES   SAT TT  POLICY*)                    LDSAT 16
      DO TO 25,  FOR I=(1)(NUMSAT)                                       LDSAT 17
C                                                                        LDSAT 18
CCC   LOAD SATELLITE DATA                                                LDSAT 19
C                                                                        LDSAT 20
      READ FROM 5,SNAME(I),SWT(I),SVOL(I),                               LDSAT 21
     * PRIOR(I),INCL(I),ORBIT(I),NO                                       LDSAT 22
     *,TTSAT(I),POLDN(I)                                                 LDSAT 23
     *,SORTE(I)                                                          LDSAT 24
      FORMAT(A6,S3,D5,D2.2,2D4,A6,S34,I5,D4,/I1,D4)                      LDSAT 25
      IF TTSAT(I) EQ 0., LET TTSAT(I)=10.                                LDSAT 26
C                                                                        LDSAT 27
CCC   PRINT SATELLITE DATA                                               LDSAT 28
C                                                                        LDSAT 29
      WRITE ON 6,SNAME(I),SWT(I),SVOL(I),PRIOR(I),INCL(I),ORBIT(I),NO    LDSAT 30
     *,TTSAT(I),POLDN(I)                                                 LDSAT 31
     *,SORTE(I)                                                          LDSAT 32
      FORMAT(S2,A6,S1,4D6,S8,A6,I5,S7,D6,I8,D6)                          LDSAT 33
      LET SORTE(I) = SORTE(I)/360.                                       LDSAT 34
      DO TO 1, FOR J=(1)(NORBS)                                          LDSAT 35
      IF ORBIT(I) NE ORBID(J), GO TO 1                                   LDSAT 36
      LET ORBIT(I) = J                                                   LDSAT 37
      GO TO 2                                                            LDSAT 38
    1 LOOP                                                               LDSAT 39
      LET IRFLG = 1                                                      LDSAT 40
      WRITE ON 6                                                         LDSAT 41
      FORMAT(*  ERROR - UNKNOWN ORBIT  *)                                LDSAT 42
    2 DO TO 15, FOR J=(1)(NO)(7)                                         LDSAT 43
                                                                         LDSAT 44
C                                                                        LDSAT 45
CCC   READ MODULE LIST FOR SATELLITE                                     LDSAT 46
C                                                                        LDSAT 47
      READ FROM 5,MODUL(J),IA(J),MODUL(J+1),IA(J+1),MODUL(J+2),IA(J+2),  LDSAT 48
     *      MODUL(J+3),IA(J+3),MODUL(J+4),IA(J+4),MODUL(J+5),IA(J+5),    LDSAT 49
     *      MODUL(J+6),IA(J+6)                                           LDSAT 50
      FORMAT(S10,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4,A6,A4)              LDSAT 51
C                                                                        LDSAT 52
CCC   PRINT MODULE LIST                                                  LDSAT 53
C                                                                        LDSAT 54
      WRITE ON 6,MODUL(J),IA(J),MODUL(J+1),IA(J+1),MODUL(J+2),IA(J+2),   LDSAT 55
     *      MODUL(J+3),IA(J+3),MODUL(J+4),IA(J+4),MODUL(J+5),IA(J+5),    LDSAT 56
     *      MODUL(J+6),IA(J+6)                                           LDSAT 57
      FORMAT(S10,14A6)                                                   LDSAT 58
   15 LOOP                                                               LDSAT 59
      DO TO 10, FOR J = (1)(NO)                                          LDSAT 60
      DO TO 20, FOR L=(1)(MITAB)                                         LDSAT 61
      IF MODUL(J) EQ MNAME(L), GO TO 5                                   LDSAT 62
   20 LOOP                                                               LDSAT 63
C                                                                        LDSAT 64
CCC   ERROR DETECTED                                                     LDSAT 65
C
```

```
      WRITE ON 6,MODUL(J)                                                LDSAT 66
      FORMAT(S3,* ERROR  MODULE - *,A6,* - NOT FOUND IN MODULE TABLE*)   LDSAT 67
      LET IRFLG = 1                                                      LDSAT 68
      GO TO 10                                                           LDSAT 69
C                                                                        LDSAT 70
CCC   PUT MODULE IN SET MDS BELONGING TO SATELLITE I                     LDSAT 71
C                                                                        LDSAT 72
    5 CREATE MDSAT                                                       LDSAT 73
      CALL CON(IA(J),K)                                                  LDSAT 74
      LET NRU(MDSAT) = K                                                 LDSAT 75
      LET NOMOD(MDSAT) = L                                               LDSAT 76
      FILE MDSAT IN MDS(I)                                               LDSAT 77
   10 LOOP                                                               LDSAT 78
   25 LOOP                                                               LDSAT 79
      RETURN                                                             LDSAT 80
      END                                                                LDSAT 81
      SUBROUTINE LDSCH(IRFLG)                                            LDSCH  2
C                                                                        LDSCH  3
CCC   SATELLITE SCHEDULE INPUT ROUTINE                                   LDSCH  4
C                                                                        LDSCH  5
      DIMENSION IA(4),A(4),IB(4)                                         LDSCH  6
      WRITE ON 6                                                         LDSCH  7
      FORMAT(*      SCHEDULES INPUT*)                                    LDSCH  8
C                                                                        LDSCH  9
CCC   LOAD SCHEDULES                                                     LDSCH 10
C                                                                        LDSCH 11
   60 READ FROM 5,IA(1),IB(1),A(1),IA(2),IB(2),A(2),IA(3),IB(3),A(3),    LDSCH 12
     *IA(4),IB(4),A(4)                                                   LDSCH 13
      FORMAT(I1,A5,S3,D4.5,I1,A6,S3,D4.5,I1,A6,S3,D4.5,I1,A6,S3,D4.5)    LDSCH 14
C                                                                        LDSCH 15
CCC   PRINT SCHEDULES                                                    LDSCH 16
C                                                                        LDSCH 17
      WRITE ON 6,IA(1),IB(1),A(1),IA(2),IB(2),A(2),IA(3),IB(3),A(3),     LDSCH 18
     $IA(4),IB(4),A(4)                                                   LDSCH 19
      FORMAT(I6S2,,A6,S3,D4.5,I2,S2,A6,S3,D4.5,I2,S2,A6,S3,D4.5,I2,S2,A6LDSCH 20
     *,S3,D4.5)                                                          LDSCH 21
      IF IA(1) EQ 0, GO TO 70                                            LDSCH 22
C                                                                        LDSCH 23
CCC FIND SYSTEM AND SAVE NEW SATELLITE LAUNCH IN NEWS                    LDSCH 24
C                                                                        LDSCH 25
      DO TO 65, FOR K = (1)(4)                                           LDSCH 26
      IF IA(K) EQ 0, GO TO 65                                            LDSCH 27
      IF A(K) GT TIMES, GO TO 65                                         LDSCH 28
      DO TO 66, FOR I=(1)(SISTB)                                         LDSCH 29
      IF IB(K) NE SYNAM(I),GO TO 66                                      LDSCH 30
      LET J = LSAT(I) - FSAT(I) + 1                                      LDSCH 31
      IF IA(K) GT J, GO TO 64                                            LDSCH 32
      LET MARKS(FSAT(I)-1+IA(K)) = 1                                     LDSCH 33
      CREATE NEW                                                         LDSCH 34
      LET SCHDT(NEW) = A(K)                                              LDSCH 35
```

```
      LET SCHSY(NEW) = FSAT(I)-1+IA(K)                                  LDSCH 36
      FILE NEW IN NEWS                                                  LDSCH 37
      GO TO 65                                                          LDSCH 38
   64 LET IRFLG = 1                                                     LDSCH 39
      WRITE ON 6,IA(K),IB(K)                                            LDSCH 40
      FORMAT(*   ERROR - MEMBER NO.*,I3,* IS NOT IN SYSTEM - *,A6)     LDSCH 41
      GO TO 65                                                          LDSCH 42
   66 LOOP                                                              LDSCH 43
C                                                                       LDSCH 44
CCC   ERROR DETECTED                                                    LDSCH 45
C                                                                       LDSCH 46
      LET IRFLG = 1                                                     LDSCH 47
      WRITE ON 6,IB(K)                                                  LDSCH 48
      FORMAT(S3,* ERROR  SYSTEM NOT FOUND - *,A6)                       LDSCH 49
   65 LOOP                                                              LDSCH 50
      GO TO 60                                                          LDSCH 51
   70 RETURN                                                            LDSCH 52
      END                                                               LDSCH 53
      SUBROUTINE LDSYS(IRFLG)                                           LDSYS  2
C                                                                       LDSYS  3
CCC   SYSTEMS INPUT ROUTINE                                             LDSYS  4
C                                                                       LDSYS  5
      READ FROM 5,NUMSYS                                                LDSYS  6
      FORMAT(I3)                                                        LDSYS  7
      IF NUMSYS LE STSTB, GO TO 1                                       LDSYS  8
      WRITE ON 6,NUMSYS,STSTB                                           LDSYS  9
      FORMAT(*  ERROR - NUMBER OF SYSTEMS INPUT(*,I6*) EXCEEDS CAPACITY(LDSYS 10
     **,I6,*)*)                                                         LDSYS 11
      LET IRFLG = 1                                                     LDSYS 12
    1 WRITE ON 6,NUMSYS                                                 LDSYS 13
      FORMAT(/I11,*SYSTEMS INPUT*/*   NAME   NUP  NTOT   SYS TT    SAT LDSYS 14
     *    PHASE    SAT     PHASE    SAT     PHASE*)                    LDSYS 15
      LET J = 0                                                         LDSYS 16
      DO TO 60, FOR I=(1)(NUMSYS)                                       LDSYS 17
      LET FSAT(I) = J + 1                                               LDSYS 18
C                                                                       LDSYS 19
CCC   LOAD SATELLITE SYSTEMS DATA                                       LDSYS 20
C                                                                       LDSYS 21
      READ FROM 5,SYNAM(I),NFUP(I),NO,TTSYS(I),                         LDSYS 22
     * ITSAT(J+1),PHASE(J+1),ITSAT(J+2),                               LDSYS 23
     * PHASE(J+2),ITSAT(J+3),PHASE(J+3)                                LDSYS 24
      FORMAT(A6,2I5,D2.1,A6,S4,D4.5,A6,S4,D4.5,A6,S4,D4.5)              LDSYS 25
      IF TTSYS(I) EQ 0., LET TTSYS(I)=15.                               LDSYS 26
      IF NFUP(I) LE 0, LET NFUP(I) = 1                                  LDSYS 27
      IF NO LE 0, LET NO = 1                                            LDSYS 28
C                                                                       LDSYS 29
CCC   PRINT SATELLITE SYSTEMS DATA                                      LDSYS 30
C                                                                       LDSYS 31
      WRITE ON 6,SYNAM(I),NFUP(I),NO,TTSYS(I),                          LDSYS 32
     * ITSAT(J+1),PHASE(J+1)                                           LDSYS 33
```

```
     * ,ITSAT(J+2),PHASE(J+2),ITSAT(J+3),PHASE(J+3)                    LDSYS  34
      FORMAT(S2,A6,2I5,D6.2,S4,A6,D6.1,S4,A6,D6.1,S4,A6,D6.1)          LDSYS  35
      LET NSAT(I) = NO                                                 LDSYS  36
      IF NO LT 4, GO TO 5                                              LDSYS  37
      READ FROM 5,ITSAT(J+4),PHASE(J+4),ITSAT(J+5),                    LDSYS  38
     * PHASE(J+5),ITSAT(J+6),PHASE(J+6)                                LDSYS  39
      FORMAT(S20,A6,S4,D4.5,A6,S4,D4.5,A6,S4,D4.5)                     LDSYS  40
      WRITE ON 6,ITSAT(J+4),PHASE(J+4),ITSAT(J+5),PHASE(J+5)           LDSYS  41
     *   ,ITSAT(J+6),PHASE(J+6)                                        LDSYS  42
      FORMAT(S31,A6,D6.1,S4,A6,D6.1,S4,A6,D6.1)                        LDSYS  43
C                                                                      LDSYS  44
CC   FIND SATELLITE                                                    LDSYS  45
C                                                                      LDSYS  46
    5 LET J = J + NO                                                   LDSYS  47
      LET LSAT(I) = J                                                  LDSYS  48
      DO TO 55, FOR L = (FSAT(I))(LSAT(I))                             LDSYS  49
      IF PHASE(L) LT 0., LET PHASE(L) = PHASE(L) + 360.                LDSYS  50
      LET ITSYS(L) = I                                                 LDSYS  51
      DO TO 45, FOR K = (1)(SITAB)                                     LDSYS  52
      IF SNAME(K) EQ ITSAT(L), GO TO 50                                LDSYS  53
   45 LOOP                                                             LDSYS  54
C                                                                      LDSYS  55
CC   ERROR DETECTED                                                    LDSYS  56
C                                                                      LDSYS  57
      LET IRFLG = 1                                                    LDSYS  58
      WRITE ON 6,ITSAT(L),SYNAM(I)                                     LDSYS  59
      FORMAT(S3,* ERROR  SATELLITE -*,A6,*- NOT FOUND, SYSTEM - *,A6)  LDSYS  60
      GO TO 55                                                         LDSYS  61
   50 LET ITSAT(L) = K                                                 LDSYS  62
   55 LOOP                                                             LDSYS  63
   60 LOOP                                                             LDSYS  64
      RETURN                                                           LDSYS  65
      END                                                              LDSYS  66
      SUBROUTINE LDVEH(IRFLG)                                          LDVEH   2
C                                                                      LDVEH   3
CC   LOAD VEHICLE DATA                                                 LDVEH   4
C                                                                      LDVEH   5
      READ FROM 5,NOVEH                                                LDVEH   6
      FORMAT(I3)                                                       LDVEH   7
      IF NOVEH LE NVEH, GO TO 1                                        LDVEH   8
      WRITE ON 6,NOVEH,NVEH                                            LDVEH   9
      FORMAT(*  ERROR - NUMBER OF VEHICLES INPUT(*,I6,*) EXCEEDS CAPACITLDVEH  10
     *Y(*,I6,*)*)                                                      LDVEH  11
      LET IRFLG = 1                                                    LDVEH  12
    1 WRITE ON 6,NOVEH                                                 LDVEH  13
      FORMAT(I8,* VEHICLES INPUT*)                                     LDVEH  14
      WRITE ON 6                                                       LDVEH  15
      FORMAT(*    NAME    DAYS    ISP    WDV     WPNU    WCONV    REFT    EXP  LDVEH  16
     *   LENGTH  NSATGE  SOLID   ID  *)                                LDVEH  17
      DO TO 5, FOR I=(1)(NOVEH)                                        LDVEH  18
```

```
      READ FROM 5,NAMEV(I),DAYSV(I),ISPV(I),WDV(I),WPNUV(I),WCONV(I),    LOVEH 19
     * REFIV(I),EXPV(I),PAYLV(I)                                         LOVEH 20
     *,NSTAG(I),SOLID(I),IDV(I)                                         LOVEH 21
      FORMAT(A6,8D5.1,2I2,A6)                                           LOVEH 22
      WRITE ON 6 ,NAMEV(I),DAYSV(I),ISPV(I),WDV(I),WPNUV(I),WCONV(I),    LOVEH 23
     * REFIV(I),EXPV(I),PAYLV(I)                                         LOVEH 24
     *,NSTAG(I),SOLID(I),IDV(I)                                         LOVEH 25
      FORMAT(S3,A6,8D7.1,2I6,S1,A6)                                     LOVEH 26
      IF NAMEV(I) EQ SEPS, CALL LOSEP(WDV(I),PAYLV(I),WCONV(I),ISPV(I),  LOVEH 27
     * WPNUV(I),EXPV(I),DAYSV(I),REFIV(I))                               LOVEH 28
    5 LOOP                                                              LOVEH 29
      RETURN                                                            LOVEH 30
      END                                                               LOVEH 31
      SUBROUTINE MARKQ                                                  MARKQ  2
C                                                                       MARKQ  3
C   MARK ALL PAYLOADS FOR LAUNCH IN ORBIT QUEUE IORB                    MARKQ  4
C                                                                       MARKQ  5
      LET NQ = 0                                                        MARKQ  6
      IF ORB(IORB) IS EMPTY, RETURN                                     MARKQ  7
      DO TO 5, FOR ALL ITORB IN ORB(IORB)                               MARKQ  8
      IF LQTIM(PITEM(ITORB)) GT 3000., GO TO 5                          MARKQ  9
      LET NQ = NQ + 1                                                   MARKQ 10
      LET ILOAD(NQ) = PITEM(ITORB)                                      MARKQ 11
                                                                        MARKQ 12
      IF NQ EQ IL, RETURN                                               MARKQ 13
    5 LOOP                                                              MARKQ 14
      RETURN                                                            MARKQ 15
      END                                                               MARKQ 16
      SUBROUTINE MCMOD                                                  MCMOD  2
C                                                                       MCMOD  3
C  STATISTICS FOR MODULES                                               MCMOD  4
C                                                                       MCMOD  5
      DO TO 5, FOR I=(1)(MITAB)                                         MCMOD  6
      IF MDCNT(I) + S121(I) EQ 0, GO TO 1                               MCMOD  7
      LET S121(I) = S121(I) + MDCNT(I)                                  MCMOD  8
      IF X121(I) LT MDCNT(I), LET X121(I) = MDCNT(I)                    MCMOD  9
      IF N121(I) GT MDCNT(I), LET N121(I) = MDCNT(I)                    MCMOD 10
      IF TRIG NE TRIGS, GO TO 1                                         MCMOD 11
      IF TRIG EQ 1, GO TO 1                                             MCMOD 12
      IF N121(I) EQ X121(I), LET N121(I) = 0                            MCMOD 13
    1 IF NOWAR(I) + S125(I) EQ 0, GO TO 2                               MCMOD 14
      LET S125(I) = S125(I) + NOWAR(I)                                  MCMOD 15
      IF X125(I) LT NOWAR(I), LET X125(I) = NOWAR(I)                    MCMOD 16
      IF N125(I) GT NOWAR(I), LET N125(I) = NOWAR(I)                    MCMOD 17
      IF TRIG NE TRIGS, GO TO 2                                         MCMOD 18
      IF TRIG EQ 1, GO TO 2                                             MCMOD 19
      IF N125(I) EQ X125(I), LET N125(I) = 0                            MCMOD 20
    2 IF NOFAL(I) + S129(I) EQ 0, GO TO 5                               MCMOD 21
      LET S129(I) = S129(I) + NOFAL(I)                                  MCMOD 22
      IF X129(I) LT NOFAL(I), LET X129(I) = NOFAL(I)                    MCMOD 23
```

```
      IF N129(I) GT NOFAL(I), LET N129(I) = NOFAL(I)                    MCMOD 24
      IF TRIG NE TRIGS, GO TO 5                                         MCMOD 25
      IF TRIG EQ 1, GO TO 5                                             MCMOD 26
      IF N129(I) EQ X129(I), LET N129(I) = 0                            MCMOD 27
    5 LOOP                                                              MCMOD 28
      RETURN                                                            MCMOD 29
      END                                                               MCMOD 30
      SUBROUTINE MCSAT                                                  MCSAT  2
C                                                                       MCSAT  3
CCC STATISTICS FOR SATELLITES                                           MCSAT  4
C                                                                       MCSAT  5
      DO TO 3, FOR I=(1)(SYORB)                                         MCSAT  6
      IF MOD(I) IS EMPTY, GO TO 3                                       MCSAT  7
      LET S227(I) = S227(I) + SATLF(I)                                  MCSAT  8
      IF X227(I) LT SATLF(I), LET X227(I) = SATLF(I)                    MCSAT  9
      IF N227(I) GT SATLF(I), LET N227(I) = SATLF(I)                    MCSAT 10
      LET A = LFSAT(I)                                                  MCSAT 11
      LET SUMSL(I) = SUMSL(I) + A                                       MCSAT 12
      IF MAXSL(I) LT A, LET MAXSL(I) = A                                MCSAT 13
      IF MINSL(I) GT A, LET MINSL(I) = A                                MCSAT 14
      DO TO 2, FOR ALL MODSY IN MOD(I)                                  MCSAT 15
      LET SUMNU(MODSY) = SUMNU(MODSY) + NUM(MODSY)                      MCSAT 16
      IF MAXNU(MODSY) LT NUM(MODSY), LET MAXNU(MODSY) = NUM(MODSY)      MCSAT 17
      IF MINNU(MODSY) GT NUM(MODSY), LET MINNU(MODSY) = NUM(MODSY)      MCSAT 18
      LET  SUMLF(MODSY) = SUMLF(MODSY) + LOADF(MODSY)                   MCSAT 19
      IF MAXLF(MODSY) LT LOADF(MODSY), LET MAXLF(MODSY) = LOADF(MODSY)  MCSAT 20
      IF MINLF(MODSY) GT LOADF(MODSY), LET MINLF(MODSY) = LOADF(MODSY)  MCSAT 21
    2 LOOP                                                              MCSAT 22
      LET A = HALST(I)-BEGST(I)                                         MCSAT 23
      IF A EQ 0., GO TO 3                                               MCSAT 24
      LET P = 100.*SDTST(I)/A                                           MCSAT 25
      LET PERST(I) = PERST(I) + P                                       MCSAT 26
      IF N216(I) GT P, LET N216(I) = P                                  MCSAT 27
      IF X216(I) LT P, LET X216(I) = P                                  MCSAT 28
    3 LOOP                                                              MCSAT 29
      RETURN                                                            MCSAT 30
      END                                                               MCSAT 31
      SUBROUTINE MCVEH                                                  MCVEH  2
C                                                                       MCVEH  3
CCC STATISTICS FOR VEHICLES                                             MCVEH  4
C                                                                       MCVEH  5
      DO TO 1, FOR I=(1)(NYEAR)                                         MCVEH  6
      LET SUM39(I) = SUM39(I)+TUGFY(I)                                  MCVEH  7
      IF MAX39(I) LT TUGFY(I), LET MAX39(I) = TUGFY(I)                  MCVEH  8
      IF MIN39(I) GT TUGFY(I),LET MIN39(I) = TUGFY(I)                   MCVEH  9
      LET SUM86(I) = SUM86(I) + SEPFY(I)                                MCVEH 10
      IF MAX86(I) LT SEPFY(I), LET MAX86(I) = SEPFY(I)                  MCVEH 11
      IF MIN86(I) GT SEPFY(I), LET MIN86(I) = SEPFY(I)                  MCVEH 12
      LET SUM90(I) = SUM90(I) + SUTFY(I)                                MCVEH 13
      IF MIN90(I) GT SUTFY(I), LET MIN90(I) = SUTFY(I)                  MCVEH 14
```

```
      IF MAX90(I) LT SUTFY(I), LET MAX90(I) = SUTFY(I)                  MCVEH 15
    1 LOOP                                                              MCVEH 16
      LET IT= 0                                                         MCVEH 17
      LET IT = IT + TUGFY(I), FOR I=(1)(NYEAR)                          MCVEH 18
      IF MTFLT LT IT, LET MTFLT = IT                                    MCVEH 19
      IF NTFLT GT IT, LET NTFLT = IT                                    MCVEH 20
      LET ITFLT = ITFLT + IT                                            MCVEH 21
      LET IT = 0                                                        MCVEH 22
      LET IT = IT + SUTFY(I), FOR I=(1)(NYEAR)                          MCVEH 23
      LET IFSUT = IFSUT + IT                                            MCVEH 24
      IF MFSUT LT IT, LET MFSUT = IT                                    MCVEH 25
      IF NFSUT GT IT, LET NFSUT = IT                                    MCVEH 26
      LET IT = 0                                                        MCVEH 27
      LET IT = IT + SEPFY(I), FOR I=(1)(NYEAR)                          MCVEH 28
      LET IFSEP = IFSEP + IT                                            MCVEH 29
      IF MFSEP LT IT, LET MFSEP = IT                                    MCVEH 30
      IF NFSEP GT IT, LET NFSEP = IT                                    MCVEH 31
      DO TO 2, FOR I=(1)(3)                                             MCVEH 32
      LET TCVA(I) = TCVA(I) + CVA(I)                                    MCVEH 33
      IF CVA(I) GT XCVA(I), LET XCVA(I) = CVA(I)                        MCVEH 34
      IF CVA(I) LT MCVA(I), LET MCVA(I) = CVA(I)                        MCVEH 35
    2 LOOP                                                              MCVEH 36
      RETURN                                                            MCVEH 37
      END                                                               MCVEH 38
      SUBROUTINE MCSYS                                                  MCSYS  2
C                                                                       MCSYS  3
CC  STATISTICS FOR SYSTEMS                                              MCSYS  4
C                                                                       MCSYS  5
      DO TO 4, FOR I=(1)(STSTB)                                         MCSYS  6
      IF SYNAM(I) EQ 0, GO TO 4                                         MCSYS  7
      LET A = 0.                                                        MCSYS  8
      DO TO 6, FOR J=(FSAT(I))(LSAT(I))                                 MCSYS  9
      LET A = A + LFSAT(J)                                              MCSYS 10
    6 LOOP                                                              MCSYS 11
      LET SYLF(I) = SYLF(I) + A                                         MCSYS 12
      IF XSYLF(I) LT A, LET XSYLF(I) = A                                MCSYS 13
      IF NSYLF(I) GT A, LET NSYLF(I) = A                                MCSYS 14
      LET A = HALSY(I) - BEGSY(I)                                       MCSYS 15
      IF A EQ 0., GO TO 4                                               MCSYS 16
      LET P = 100.*SDTSY(I)/A                                           MCSYS 17
      LET PERSY(I) = PERSY(I) + P                                       MCSYS 18
      IF N200(I) GT P, LET N200(I) = P                                  MCSYS 19
      IF X200(I) LT P, LET X200(I) = P                                  MCSYS 20
    4 LOOP                                                              MCSYS 21
      RETURN                                                            MCSYS 22
      END                                                               MCSYS 23
      ENDOGENOUS EVENT NEWME                                            NEWME  2
C                                                                       NEWME  3
CC  REPLACEMENT OR UPGRADING OF ME                                      NEWME  4
C                                                                       NEWME  5
```

```
C  FIX UP AND TEST ************************************************************ NEWME   6
      IF MSTAT(PMOD(NEWME)) EQ UP, CALL SHIP(PSAT(NEWME),PMOD(NEWME))            NEWME   7
      IF MSTAT(PMOD(NEWME)) NE UP, CALL STATUS(IX,IY,5)                          NEWME   8
      DESTROY NEWME                                                              NEWME   9
      RETURN                                                                     NEWME  10
      END                                                                        NEWME  11
      ENDOGENOUS EVENT NWSAT                                                     NWSAT   2
                                                                                 NWSAT   3
C     THIS ROUTINE WILL ATTEMPT TO SCHEDULE THE LAUNCHING OF A PAYLOAD           NWSAT   4
                                                                                 NWSAT   5
C     ON A VEHICLE.                                                              NWSAT   6
                                                                                 NWSAT   7
C     IT WILL INCLUDE FIRST LAUNCH CHECK TO SET FINAL 6 MONTH LATER GO.          NWSAT   8
                                                                                 NWSAT   9
      LET IS = PSAT(NWSAT)                                                       NWSAT  10
      DESTROY NWSAT                                                              NWSAT  11
      CALL STATUS(IS,0,1)                                                        NWSAT  12
      LET T = IGOSY (ITSYS(IS))                                                  NWSAT  13
      IF T EQ 0., GO TO 1                                                        NWSAT  14
      IF TIME GT T, RETURN                                                       NWSAT  15
    1 CALL SHIP(IS,0)                                                            NWSAT  16
      LET IM =0                                                                  NWSAT  17
      LET DELAY = WSATN                                                          NWSAT  18
      IF SSTAT(IS) EQ UP, LET DELAY = WSATU                                      NWSAT  19
      IF DELAY GT TIMES - TIME, LET DELAY = TIMES - TIME                         NWSAT  20
      IF DELAY LT 0., LET DELAY = 0.                                             NWSAT  21
      LET DTIME(IS) =  TIME + DELAY                                              NWSAT  22
                                                                                 NWSAT  23
C  SCHEDULE MANDATORY LAUNCH                                                     NWSAT  24
                                                                                 NWSAT  25
      IF SORTE(ITSAT(IS)) NE 0., RETURN                                          NWSAT  26
    5 CREATE LAUNC                                                               NWSAT  27
      LET PSAT(LAUNC) = IS                                                       NWSAT  28
      LET PMOD(LAUNC) = IM                                                       NWSAT  29
      CAUSE LAUNC AT TIME + DELAY                                                NWSAT  30
      RETURN                                                                     NWSAT  31
      END                                                                        NWSAT  32
      SUBROUTINE PASER                                                           PASER   2
                                                                                 PASER   3
C     PHASING ALGORITHM                                                          PASER   4
                                                                                 PASER   5
      LET KSAT = 0                                                               PASER   6
      DO TO 5, FOR J=(1)(NQ)                                                     PASER   7
      IF IMOD(ILOAD(J)) + IRT(ILOAD(J)) EQ 0, LET KSAT = 1                       PASER   8
    5  LOOP                                                                      PASER   9
      LET MARK = 0                                                               PASER  10
    6 DO TO 9, FOR K=(1)(NQ-1)                                                   PASER  11
      DO TO 11, FOR J=(K+1)(NQ)                                                  PASER  12
      IF ANGLE(ILOAD(K)) LE ANGLE(ILOAD(J)), GO TO 11                            PASER  13
      LET L = ILOAD(K)                                                           PASER  14
```

```
      LET ILOAD(K) = ILOAD(J)                                                  PASER 15
      LET ILOAD(J) = L                                                         PASER 16
   11 LOOP                                                                     PASER 17
    9 LOOP                                                                     PASER 18
      LET CX = 0                                                               PASER 19
      LET JSAT = NQ                                                            PASER 20
      DO TO 12, FOR J=(2)(NQ)                                                  PASER 21
      IF ANGLE(ILOAD(J))-ANGLE(   ILOAD(J-1)) LT CX, GO TO 12                  PASER 22
      LET CX = ANGLE(ILOAD(J)) - ANGLE(ILOAD(J-1))                             PASER 23
      LET JSAT = J                                                             PASER 24
   12 LOOP                                                                     PASER 25
      IF 360.-ANGLE(ILOAD(NQ))+ANGLE(ILOAD(1)) GT CX, LET JSAT = 1             PASER 26
      IF JSAT EQ 1, GO TO 14                                                   PASER 27
      IF MARK GT 1, GO TO 14                                                   PASER 28
      LET MARK = MARK + 1                                                      PASER 29
   13 LET ANGLE(ILOAD(J)) = ANGLE(ILOAD(J)) - 360., FOR J=(JSAT)(NQ)           PASER 30
      GO TO 6                                                                  PASER 31
   14 IF KSAT EQ 0, GO TO 50                                                   PASER 32
      IF RQSEP(IORB) NE 0, RETURN                                              PASER 33
      IF IMOD(ILOAD(1))+IRT(ILOAD(1)) EQ 0, GO TO 50                           PASER 34
      IF NQ GT 2, GO TO 20                                                     PASER 35
      LET L = ILOAD(2)                                                         PASER 36
      LET ILOAD(2) = ILOAD(1)                                                  PASER 37
      LET ILOAD(1) = L                                                         PASER 38
      GO TO 50                                                                 PASER 39
   20 DO TO 25, FOR J=(1)(NQ)                                                  PASER 40
      IF IMOD(ILOAD(J))+IRT(ILOAD(J)) EQ 0, GO TO 21                           PASER 41
   25 LOOP                                                                     PASER 42
      GO TO 50                                                                 PASER 43
   21 IF ANGLE(ILOAD(J))-ANGLE(ILOAD(1)) GT                                    PASER 44
     *    ANGLE(ILOAD(NQ))-ANGLE(ILOAD(J)), GO TO 22                          PASER 45
   23 DO TO 26, FOR K=(1)(J/2)                                                 PASER 46
      LET L = ILOAD(J-K+1)                                                     PASER 47
      LET ILOAD(J-K+1) = ILOAD(K)                                              PASER 48
      LET ILOAD(K) = L                                                         PASER 49
   26 LOOP                                                                     PASER 50
      GO TO 50                                                                 PASER 51
   22 IF J EQ NQ, GO TO 23                                                     PASER 52
      LET IJ =(NQ-J+1)/2                                                       PASER 53
      DO TO 27, FOR K=(J)(J+IJ-1)                                              PASER 54
      LET L = ILOAD(NQ-K+J)                                                    PASER 55
      LET ILOAD(NQ-K+J) = ILOAD(K)                                             PASER 56
      LET ILOAD(K) = L                                                         PASER 57
   27 LOOP                                                                     PASER 58
      LET J = NQ                                                               PASER 59
      GO TO 23                                                                 PASER 60
   50 RETURN                                                                   PASER 61
      END                                                                      PASER 62
      SUBROUTINE PAYLQ(IS,IM)                                                  PAYLQ  2
C                                                                              PAYLQ  3
```

```
C  ENTER PAYLOAD INTO LOADING QUEUE AND ORBIT QUEUE                    PAYLQ   4
C                                                                      PAYLQ   5
      CREATE ITORB                                                     PAYLQ   6
      CREATE PAYLD CALLED IX                                           PAYLQ   7
      LET PITEM(ITORB) = IX                                            PAYLQ   8
      LET ISAT(IX) = IS                                                PAYLQ   9
      LET IMOD(IX) = IM                                                PAYLQ  10
      IF IM EQ 0, LET PAYWT(IX) = SWT(ITSAT(IS))                       PAYLQ  11
      IF IM NE 0, LET PAYWT(IX) = MODWT(NOMOD(IM))                     PAYLQ  12
      LET ANGLE(IX) = PHASE(IS)                                        PAYLQ  13
      LET IRT(IX) = RTFLG                                              PAYLQ  14
      LET GOTIM(IX) = 0.                                               PAYLQ  15
      IF IM NE 0, LET PAYLN(IX) = 0.                                   PAYLQ  16
      IF IM EQ 0, LET PAYLN(IX) = SVOL(ITSAT(IS))                      PAYLQ  17
      CALL REDUN(IS,IM)                                                PAYLQ  18
      IF DELTA LT 0., LET DELTA = 0.                                   PAYLQ  19
      LET LQTIM(IX) = TIME + DELTA                                     PAYLQ  20
      FILE IX IN LOAD                                                  PAYLQ  21
      LET TORB(ITORB) = TIME + DELTA                                   PAYLQ  22
      FILE ITORB IN ORB(IORB)                                          PAYLQ  23
      RETURN                                                           PAYLQ  24
      END                                                              PAYLQ  25
      SUBROUTINE PROP(IGO)                                             PROP   2
C                                                                      PROP   3
CCCCC COMPUTE PROPELLANT REQUIRED TO DELIVER NQ ITEMS IN CLOAD ARRAY   PROP   4
C                                                                      PROP   5
C                                                                      PROP   6
C  VOLUME(LENGTH) CONSTRAINT                                           PROP   7
C                                                                      PROP   8
      DIMENSION PLEG(20),DVLEG(20),THETA(20),A(20)                     PROP   9
      LET NMD = 0                                                      PROP  10
      LET KX = 0                                                       PROP  11
      DO TO 10, FOR J=(1)(NQ)                                          PROP  12
      IF SQRTE(ITSAT(ISAT(ILOAD(J)))) NE 0, LET KX = J                 PROP  13
      IF IMOD(ILOAD(J)) NE 0, LET NMD = NMD + 1                        PROP  14
   10 LOOP                                                             PROP  15
      IF KX EQ 0, GO TO 1                                              PROP  16
      IF KX GT 1, GO TO 70                                             PROP  17
      IF NQ GT 1, GO TO 70                                             PROP  18
      GO TO 90                                                         PROP  19
    1 LET KX = 0                                                       PROP  20
      LET JK = RQUP(IORB)                                              PROP  21
C TEMP FIX                                                             PROP  22
      IF JK EQ 0, LET JK = 1                                           PROP  23
      LET PALEN = PAYLV(RQSUT(IORB))                                   PROP  24
      IF RQUP(IORB) NE 0, LET PALEN = PAYLV(RQUP(IORB))                PROP  25
      LET  SU =(NMD+NINSU-1)/NINSU                                     PROP  26
      LET PAY = SU*LENSU                                               PROP  27
      DO TO 20, FOR L=(1)(NQ)                                          PROP  28
      IF IMOD(ILOAD(L))+IRT(ILOAD(L)) EQ 0,                            PROP  29
```

```
*   LET PAY = PAY + SVOL(ITSAT(ISAT(ILOAD(L))))                    PROP  30
20 LOOP                                                            PROP  31
   IF PAY GT PALEN, GO TO 70                                       PROP  32
   LET DAYS = DAYSV(JK)                                            PROP  33
   LET WCONS = WCONV(RQSUT(IORB))                                  PROP  34
   LET DV = ORBDV(IORB)                                            PROP  35
   LET RA = ORBRA(IORB)                                            PROP  36
   LET VCO = 25936.                                                PROP  37
   LET P1 = ORBPD(IORB)                                            PROP  38
   LET WRET = 0.                                                   PROP  39
   LET WDEP = 0.                                                   PROP  40
   LET WSERV = 0.                                                  PROP  41
   DO TO 5, FOR J=(1)(NQ)                                          PROP  42
   CALL QUAD(ANGLE(ILOAD(J)))                                      PROP  43
   IF IMOD(ILOAD(J)) EQ 0, LET WDEP = WDEP + PAYWT(ILOAD(J))       PROP  44
   IF IMOD(ILOAD(J)) NE  0, LET WSERV = WSERV + PAYWT(ILOAD(J))    PROP  45
   IF IRT(ILOAD(J)) EQ 0, GO TO 5                                  PROP  46
   LET WDEP = WDEP - PAYWT(ILOAD(J))                               PROP  47
   LET WRET = WRET + PAYWT(ILOAD(J))                               PROP  48
 5 LOOP                                                            PROP  49
                                                                   PROP  50
C  COMPUTE PERFORMANCE - UP/DOWN PAYLOADS                          PROP  51
C                                                                  PROP  52
   LET WSERV = WSERV + WTSU* SU                                    PROP  53
   LET WUPL = WDEP + WSERV                                         PROP  54
   LET WSPL = WRET + WSERV                                         PROP  55
   IF PDOWN NE 0, LET WSPL = WRET                                  PROP  56
   IF RQUP(IORB) EQ 0, GO TO 100                                   PROP  57
   LET NS = NSTAG(JK)                                              PROP  58
   IF NS EQ 0, LET NS = 1                                          PROP  59
   DO TO 40, FOR NK=(1)(NS)                                        PROP  60
   LET JX = JK + NK - 1                                            PROP  61
   IF EXVEH EQ 0, LET EXVEH = EXPV(JX)                             PROP  62
   LET XVEH = EXVEH                                                PROP  63
   CALL LINKT(NK,ISPV(JX),WDV(JX),WPNUV(JX),WCONV(JX),XVEH,        PROP  64
  * SOLID(JX),WCONV(RQSUT(IORB)))                                  PROP  65
40 LOOP                                                            PROP  66
   IF NS GT 1, CALL TWOBR(DV,DV1(IORB))                            PROP  67
 6 LET NLEG = 1                                                    PROP  68
   LET PLEG(1) = WUPL                                              PROP  69
   LET DVLEG(1) = DV                                               PROP  70
   IF NQ EQ 1, GO TO 1000                                          PROP  71
   LET GDAY = DAYS - .5                                            PROP  72
C                                                                  PROP  73
C  COMPUTE PROPELLANT FOR SERVICING                                PROP  74
C                                                                  PROP  75
C  VOLUME(LENGTH) CONSTRAINT                                       PROP  76
   CALL PASER                                                      PROP  77
50 LET PANGL(1) = 0.                                               PROP  78
   LET PANGL(J) = ANGLE(ILOAD(J)) - ANGLE(ILOAD(J-1)),             PROP  79
```

```
      * FOR J=(2)(NQ)                                                    PROP  80
       DO TO 60, FOR MELT = (2)(NQ)                                     PROP  81
       LET X = WSERV                                                    PROP  82
       LET TO = 0.                                                      PROP  83
       LET NFF = NQ +2 -MELT                                            PROP  84
       LET TO = TO + ABS(PANGL(J)), FOR J=(2)(NFF)                      PROP  85
       DO TO 55, FOR J=(NFF)(NQ)                                        PROP  86
       IF IMOD(ILOAD(J)) EQ 0, LET X = X + PAYWT(ILOAD(J))              PROP  87
    55 LOOP                                                             PROP  88
C                                                                       PROP  89
CC   COMPUTE PHASING PROPELLANT                                         PROP  90
C                                                                       PROP  91
       LET FLTIM(NFF) = 0.                                              PROP  92
       IF PANGL(NFF) EQ 0., GO TO 60                                    PROP  93
       LET IETA = ABS(PANGL(NFF))/TO*GDAY + .2                          PROP  94
       IF IETA EQ 0, LET IETA = 1                                       PROP  95
       LET ETA = IETA                                                   PROP  96
       LET DP1 = PANGL(NFF)                                             PROP  97
       LET PO = P1*(1.-PANGL(NFF)/(360.*ETA))                           PROP  98
       LET TO = TO - ABS(PANGL(NFF))                                    PROP  99
       LET FLTIM(NFF) = (PO*ETA)/(P1*30.*12.)                           PROP 100
       LET GDAY = GDAY - PO/P1*ETA                                      PROP 101
       IF GDAY LT -.5, GO TO 70                                         PROP 102
       IF PO LT .3535*P1, GO TO 70                                      PROP 103
       LET RP = RA*(2.*(PO/P1)**(2./3.)-1.)                             PROP 104
       LET VCP = VCO * SQRT(RO/RP)                                      PROP 105
       LET DVO = 2.*VCP*(SQRT(1./(RA/RP))-SQRT(2./((RA/RP)*(1.+RA/RP)))) PROP 106
       LET NLEG = NLEG + 1                                              PROP 107
       LET PLEG(NFF) = X                                                PROP 108
       LET DVLEG(NFF) = DVO                                             PROP 109
    60 LOOP                                                             PROP 110
  1000 LET NLEG = NLEG + 1                                              PROP 111
       LET THETA(J-1) = PANGL(J), FOR J=(2)(NQ)                         PROP 112
       IF RQSEP(IORB) NE 0, CALL SEPSV(NLEG-2,P1,VCO,THETA(1),PLEG(2))  PROP 113
       LET PLEG(NLEG) = WSPL                                            PROP 114
       LET DVLEG(NLEG) = DV                                             PROP 115
       LET DVLEG(J) = DVLEG(J)*1.01, FOR J=(1)(NLEG)                    PROP 116
       IF EXVEH NE 0, LET PLEG(NLEG) = 0                                PROP 117
       IF EXVEH NE 0, LET DVLEG(NLEG) = 0                               PROP 118
       LET JKO = 0                                                      PROP 119
       CALL CONEC(NS,JK,JKO)                                            PROP 120
       IF IORBO NE 0, GO TO 64                                          PROP 121
       CALL PRFORM(DVLEG,PLEG,NLEG,WP,**NEXIT,**MSEP,NT)                PROP 122
       LET ISEPS = 0                                                    PROP 123
       IF WP GT 0., GO TO 65                                            PROP 124
       IF RQSEP(IORB) EQ 0, GO TO 65                                    PROP 125
       LET WP = 10.                                                     PROP 126
       CALL GETV(IGO)                                                   PROP 127
       IF IGO EQ 3, GO TO 65                                            PROP 128
    64 LET WP = 10.                                                     PROP 129
```

```
      LET JKO = 1                                                        PROP 130
      CALL CONEC(NS,JK,JKO)                                              PROP 131
      CALL PRFORM(DVLEG,PLEG,NLEG,WP,**NEXIT,**MSEP,NT)                  PROP 132
      LET MSEP = 1                                                       PROP 133
      IF NT GT 1, LET WP = -10.                                          PROP 134
      WRITE ON 6,NT,NEXIT,WP                                             PROP 135
      FORMAT(2I5,D4.2)                                                   PROP 136
      IF NT GT 1, WRITE ON 6,NT                                          PROP 137
      FORMAT(*   NO TUGS/SEPS  *,I4)                                    PROP 138
      IF NEXIT EQ 6, LET ISEPS = 0                                       PROP 139
      IF NEXIT GT 6, GO TO 110                                           PROP 140
      IF NEXIT EQ 3, GO TO 110                                           PROP 141
      IF NEXIT EQ 4, GO TO 110                                           PROP 142
   65 LET W(IORB) = WP                                                   PROP 143
      IF W(IORB) LT 0., RETURN                                           PROP 144
C                                                                        PROP 145
C  SAVE PREVIOUS GOOD LAUNCH SETUP FOR NEXT FLIGHT (IF SEQUENCE ENDS )   PROP 146
C                                                                        PROP 147
      LET FLTIM(1) = 6./(24.*12.*30.)                                    PROP 148
      LET FLTIM(J) = FLTIM(J) + FLTIM(J-1), FOR J=(2)(NQ)                PROP 149
      IF ISEPS NE 0, CALL TPHAS(A,NLEG)                                  PROP 150
      LET FLTIM(J) = A(J), FOR J=(1)(NLEG)                               PROP 151
      LET FLY = FLTIM(NQ) + 6./(24.*12.*30.)                             PROP 152
      LET NL(IORB) = NQ                                                  PROP 153
      LET GOTIM(ILOAD(J)) = FLTIM(J), FOR J=(1)(NQ)                      PROP 154
      LET ORBTM(IORB) = FLY                                              PROP 155
      LET ANMD(IORB) = NMD                                               PROP 156
      LET CITEM(ILOAD(J)) = ILOAD(J+1), FOR J=(1)(NQ-1)                  PROP 157
      LET PQUE(IORB) = ILOAD(1)                                          PROP 158
      RETURN                                                             PROP 159
   70 LET W(IORB) = -10.                                                 PROP 160
      RETURN                                                             PROP 161
C                                                                        PROP 162
CCC    SINGLE SORTIE OPTION                                              PROP 163
C                                                                        PROP 164
   90 LET W(IORB) = -50.                                                 PROP 165
      LET NL(IORB) = 1                                                   PROP 166
      LET GOTIM(ILOAD(1))= 6./8640.                                      PROP 167
      LET ORBTM(IORB) = SORTE(ITSAT(ISAT(ILOAD(1))))                     PROP 168
      LET ANMD(IORB) = 0                                                 PROP 169
      LET PQUE(IORB) = ILOAD(1)                                          PROP 170
      RETURN                                                             PROP 171
C                                                                        PROP 172
CCC    SHUTTLE ONLY OPTION                                               PROP 173
C                                                                        PROP 174
  100 IF WUPL GT WCONS, GO TO 70                                         PROP 175
      IF WSPL GT WCONS, GO TO 70                                         PROP 176
      LET NL(IORB) = NQ                                                  PROP 177
      LET W(IORB) = 100.*(1.-WUPL/WCONS)                                 PROP 178
      LET GOTIM(ILOAD(J)) = 6./8640., FOR J=(1)(NQ)                      PROP 179
```

```
      LET ANMD(IORB) = SU                                                PROP 180
      LET ORBIM(IORB) = 24./8540.                                        PROP 181
      LET CITEM(ILOAD(J)) = ILOAD(J+1), FOR J=(1)(NQ-1)                  PROP 182
      LET PQUE(IORB) = ILOAD(1)                                          PROP 183
      RETURN                                                             PROP 184
  110 IF IORBO NE 0., GO TO 120                                          PROP 185
      LET IORBO = 1                                                      PROP 186
      LET NL(IORB) = 0                                                   PROP 187
      LET W(IORB) = -10.                                                 PROP 188
      RETURN                                                             PROP 189
  120 LET IORBO = 0                                                      PROP 190
      LET W(IORB) = -10.                                                 PROP 191
      RETURN                                                             PROP 192
VJKO  0                                                                  PROP 193
VIORBOO                                                                  PROP 194
      END                                                                PROP 195
      SUBROUTINE QDMP(IS,IM,ILL)                                         QDMP   2
C                                                                        QDMP   3
CCCC  REMOVES EARLIER DUPLICATE PAYLOAD FROM LOADING QUEUE               QDMP   4
CCCC  ALSO BLOCKS MODULES FROM ENTERING QUEUE                            QDMP   5
C                                                                        QDMP   6
      LET IORB = ORBIT(ITSAT(IS))                                        QDMP   7
      IF SORTE(ITSAT(IS)) NE 0., RETURN                                  QDMP   8
      IF RTFLG EQ 0, GO TO 1                                             QDMP   9
      IF NPOS(IS) GT 1, RETURN                                           QDMP  10
    1 IF ORB(IORB) IS EMPTY, GO TO 3                                     QDMP  11
      DO TO 5, FOR ALL ITORB IN ORB(IORB)                                QDMP  12
      IF ISAT(PITEM(ITORB)) NE IS, GO TO 5                               QDMP  13
      IF IMOD(PITEM(ITORB)) EQ IM, GO TO 2                               QDMP  14
      IF IRT(PITEM(ITORB)) NE 0, GO TO 5                                 QDMP  15
      IF IM EQ 0, GO TO 2                                                QDMP  16
      IF IMOD(PITEM(ITORB)) EQ 0, GO TO 7                                QDMP  17
    5 LOOP                                                               QDMP  18
      GO TO 3                                                            QDMP  19
    2 CALL DROPQ(PITEM(ITORB),IORB)                                      QDMP  20
      LET NL(IORB) = 0                                                   QDMP  21
      GO TO 1                                                            QDMP  22
    3 LET ILL = 0                                                        QDMP  23
      RETURN                                                             QDMP  24
    7 LET ILL = 1                                                        QDMP  25
      RETURN                                                             QDMP  26
      END                                                                QDMP  27
      SUBROUTINE QUAD(A)                                                 QUAD   2
    5 IF A GT 0., GO TO 10                                               QUAD   3
      LET A = A + 360.                                                   QUAD   4
      GO TO 5                                                            QUAD   5
   10 IF A LT 360., RETURN                                               QUAD   6
      LET A = A - 360.                                                   QUAD   7
      GO TO 10                                                           QUAD   8
      END                                                                QUAD   9
```

```
      SUBROUTINE REDUN(IS,IM)                                          REDUN   2
      LET DELTA = 0                                                    REDUN   3
      IF IM EQ 0, RETURN                                               REDUN   4
      LET EDO(IM) = 1                                                  REDUN   5
      LET IX = 0                                                       REDUN   6
      DO TO 5, FOR ALL MODSY IN MOD(IS)                                REDUN   7
      IF NRU(MODSY) EQ 0, GO TO 5                                      REDUN   8
      IF NRU(MODSY) EQ 100, GO TO 5                                    REDUN   9
      IF NRU(MODSY) EQ 1, GO TO 3                                      REDUN  10
      IF IX NE 0, GO TO 4                                              REDUN  11
      LET IB = 0                                                       REDUN  12
      LET IX = NRU(MODSY)                                              REDUN  13
      LET IY = 0                                                       REDUN  14
      LET IK = MODSY                                                   REDUN  15
      DO TO 1, FOR I=(1)(IX)                                           REDUN  16
      IF IM EQ IK, LET IY = 1                                          REDUN  17
      IF EDO(IK) NE 0, LET IB = IB + 1                                 REDUN  18
      LET IK = SMOD(IK)                                                REDUN  19
    1 LOOP                                                             REDUN  20
      IF IY EQ 0, GO TO 5                                              REDUN  21
      LET IB = IX - NRU(SMOD(MODSY)) - IB                              REDUN  22
      IF IB GT 0, LET DELTA = 3000.                                    REDUN  23
      IF IB EQ 0, LET DELTA = -3000.                                   REDUN  24
      RETURN                                                           REDUN  25
    3  IF IM NE MODSY, GO TO 4                                         REDUN  26
       LET DELTA = 3000.                                               REDUN  27
       RETURN                                                          REDUN  28
    4 LET IX = 0                                                       REDUN  29
    5 LOOP                                                             REDUN  30
      RETURN                                                           REDUN  31
      END                                                              REDUN  32
      ENDOGENOUS EVENT REFMO                                           REFMO   2
C                                                                      REFMO   3
CC    THIS ROUTINE TAKES CARE OF REFURB OF MODULES                     REFMO   4
C                                                                      REFMO   5
      LET IM = PMOD (REFMO)                                            REFMO   6
      LET MDCNT(IM) = MDCNT(IM) + 1                                    REFMO   7
      DESTROY REFMO                                                    REFMO   8
      IF TRIG EQ 0, WRITE ON 6,TIME,MNAME(IM)                          REFMO   9
      FORMAT(S5,M5.2.2,S43,A6,S4,*REFURBISHED*)                        REFMO  10
      RETURN                                                           REFMO  11
      END                                                              REFMO  12
      ENDOGENOUS EVENT REFSA                                           REFSA   2
C                                                                      REFSA   3
CC    THIS ROUTINE TAKES CARE OF REFURB OF SATELLITES                  REFSA   4
C                                                                      REFSA   5
      RETURN                                                           REFSA   6
      END                                                              REFSA   7
      ENDOGENOUS EVENT REFVE                                           REFVE   2
C                                                                      REFVE   3
```

```
C      THIS ROUTINE TAKES CARES OF REFURB OF VEHICLES                    REFVE   4
C                                                                        REFVE   5
       IF VNAME(REFVE) EQ SHUT, GO TO 1                                  REFVE   6
       IF EXVEH NE 0, GO TO 2                                            REFVE   7
     1 LET IX = 0                                                        REFVE   8
       IF TRIG EQ 0, WRITE ON 6,TIME,VNAME(REFVE),PMOD(REFVE)            REFVE   9
       FORMAT(SE,M5.2.2,S3,S60,A6,I3,S1,*REFURBISHED*)                   REFVE  10
     2 LET IC = 0                                                        REFVE  11
       IF VNAME(REFVE) EQ SEPS, GO TO 6                                  REFVE  12
       IF VNAME(REFVE) EQ SHUT, GO TO 5                                  REFVE  13
       LET IC = IC + VTUG(I), FOR I=(1)(NTUG)                            REFVE  14
       LET VTUG(PMOD(REFVE)) = 1                                         REFVE  15
       IF IC NE 0, GO TO 15                                              REFVE  16
       GO TO 10                                                          REFVE  17
     5 LET IC = IC + VSHUT(I), FOR I=(1)(NSHUT)                          REFVE  18
       LET VSHUT(PMOD(REFVE)) = 1                                        REFVE  19
       IF IC NE 0, GO TO 15                                              REFVE  20
       GO TO 10                                                          REFVE  21
     6 LET IC = IC + VSEPS(I), FOR I=(1)(NSEPS)                          REFVE  22
       LET VSEPS(PMOD(REFVE)) = 1                                        REFVE  23
       IF IC NE 0, GO TO 15                                              REFVE  24
    10 DO TO 11, FOR I=(1)(NORBS)                                        REFVE  25
       LET IORB = I                                                      REFVE  26
       IF ORB(IORB) IS EMPTY, GO TO 11                                   REFVE  27
       IF W(IORB) GT 0., GO TO 11                                        REFVE  28
       CALL GETV(IGO)                                                    REFVE  29
       IF IGO NE 0, GO TO 11                                             REFVE  30
       CALL SHIP(-1,0)                                                   REFVE  31
    11 LOOP                                                              REFVE  32
    15 DESTROY REFVE                                                     REFVE  33
       RETURN                                                            REFVE  34
       END                                                               REFVE  35
       ENDOGENOUS EVENT REMOV                                            REMOV   2
C                                                                        REMOV   3
C      SATELLITE FROM ORBIT                                              REMOV   4
C                                                                        REMOV   5
       LET IS = PSAT(REMOV)                                              REMOV   6
       LET NPOS(IS) = NPOS(IS) - 1                                       REMOV   7
       CALL STATUS(IS,0,9)                                               REMOV   8
       CALL QOMP(IS,0,IL)                                                REMOV   9
       DESTROY REMOV                                                     REMOV  10
       RETURN                                                            REMOV  11
       END                                                               REMOV  12
       ENDOGENOUS EVENT RETRI                                            RETRI   2
C                                                                        RETRI   3
C      THIS ROUTINE TAKES CARE OF RETRIEVAL OF SATELLITES                RETRI   4
C                                                                        RETRI   5
       LET RTFLG = 1                                                     RETRI   6
       CALL SHIP(PSAT(RETRI),0)                                          RETRI   7
       LET RTFLG = 0                                                     RETRI   8
```

```
      DESTROY RETRI                                                    RETRI   9
      RETURN                                                           RETRI  10
      END                                                              RETRI  11
      ENDOGENOUS EVENT SATDN                                           SATDN   2
C                                                                      SATDN   3
CCC   SATELLITE VOLUNTARILY GOES DOWN AT TERMINATION TIME              SATDN   4
C                                                                      SATDN   5
      LET IS = PSAT(SATDN)                                             SATDN   6
      DO TO 200, FOR ALL MODSY IN MOD(IS)                              SATDN   7
      CALL QDMP(IS,MODSY,ILL)                                          SATDN   8
  200 LOOP                                                             SATDN   9
      CALL STATUS(IS,0,3)                                              SATDN  10
      LET MARKS(IS) = 0                                                SATDN  11
      DESTROY SATDN                                                    SATDN  12
      RETURN                                                           SATDN  13
      END                                                              SATDN  14
      SUBROUTINE SAVER(T2,IS)                                          SAVER   2
      LET IPOL = POLDN(IISAT(IS))                                      SAVER   3
      LET JSY = IISYS(IS)                                              SAVER   4
      IF IPOL LT 2, RETURN                                             SAVER   5
      IF IPOL GT 4, RETURN                                             SAVER   6
      IF IPOL EQ 2, GO TO 10                                           SAVER   7
CCC                                                                    SAVER   8
CCC   SCHEDULE SATELLITE RETRIEVAL (RETRI) AT TERMINATION TIME +-      SAVER   9
CCC                                                                    SAVER  10
      IF MARKD(IS) EQ 0, GO TO 1                                       SAVER  11
      CANCEL RETRI CALLED MARKD(IS)                                    SAVER  12
      DESTROY RETRI CALLED MARKD(IS)                                   SAVER  13
      LET MARKD(IS) = 0                                                SAVER  14
    1 LET T = T2 + WAIT2                                               SAVER  15
      IF T LT TIME, GO TO 10                                           SAVER  16
      IF T GT TGOSY(JSY), GO TO 10                                     SAVER  17
      IF T GT TIMES- WSATN, GO TO 10                                   SAVER  18
      CREATE RETRI                                                     SAVER  19
      LET PSAT(RETRI) = IS                                             SAVER  20
      CAUSE RETRI AT T                                                 SAVER  21
   10 IF IPOL GT 3, GO TO 20                                           SAVER  22
C                                                                      SAVER  23
CCC   SCHEDULE NEW SATELLITE (NWSAT) AT TERMINATION TIME +-            SAVER  24
CCC                                                                    SAVER  25
      IF MARKU(IS) EQ 0, GO TO 2                                       SAVER  26
      CANCEL NWSAT CALLED MARKU(IS)                                    SAVER  27
      DESTROY NWSAT CALLED MARKU(IS)                                   SAVER  28
      LET MARKU(IS) = 0                                                SAVER  29
    2 LET T = T2                                                       SAVER  30
      IF T GT TGOSY(JSY), GO TO 20                                     SAVER  31
      IF T LT TIME, GO TO 20                                           SAVER  32
      IF T GT TIMES- WSATN, GO TO 20                                   SAVER  33
      CREATE NWSAT                                                     SAVER  34
      LET PSAT(NWSAT) = IS                                             SAVER  35
```

```
      CAUSE NWSAT AT T                                                  SAVER 36
   20 RETURN                                                            SAVER 37
      END                                                               SAVER 38
      SUBROUTINE SHIP (IS,IM)                                           SHIP   2
C                                                                       SHIP   3
CC    THIS IS THE LOADING ROUTINE                                       SHIP   4
C                                                                       SHIP   5
      IF IS GT 0, LET IORB = ORBIT(ITSAT(IS))                           SHIP   6
      LET EXVEH = EXORB(IORB)                                           SHIP   7
      IF IS LE 0, GO TO 30                                              SHIP   8
      IF IM EQ 0, GO TO 1                                               SHIP   9
      IF EXMOD EQ 100, RETURN                                           SHIP  10
    1 CALL QDMP(IS,IM,ILL)                                              SHIP  11
      IF ILL NE 0, RETURN                                               SHIP  12
      CALL PAYLQ(IS,IM)                                                 SHIP  13
      CALL GETV(IGO)                                                    SHIP  14
      IF W(IORB) GT 0., LET W(IORB) = 0.                                SHIP  15
    5 IF ORB(IORB) IS EMPTY, RETURN                                     SHIP  16
      IF NL(IORB) EQ IL, GO TO 10                                       SHIP  17
      IF W(IORB) LT 0., GO TO 10                                        SHIP  18
      IF W(IORB) GT 0., RETURN                                          SHIP  19
      CALL MARKQ                                                        SHIP  20
      IF NQ EQ 0, RETURN                                                SHIP  21
      CALL PROP(IGO)                                                    SHIP  22
      LET EXORB(IORB) = EXVEH                                           SHIP  23
      IF W(IORB) GE 0., RETURN                                          SHIP  24
      IF IGO EQ 3, RETURN                                               SHIP  25
C                                                                       SHIP  26
CC    LAUNCH PAYLOADS FROM QUEUE - SLOAD                                SHIP  27
C                                                                       SHIP  28
   10 IF NL(IORB) EQ 0, GO TO 30                                        SHIP  29
      IF IGO NE 0 , RETURN                                              SHIP  30
      CALL ISSUE                                                        SHIP  31
      LET W(IORB) = 0                                                   SHIP  32
      LET NL(IORB) = 0                                                  SHIP  33
      LET EXVEH = EXORB(IORB)                                           SHIP  34
      GO TO 30                                                          SHIP  35
   31 LET J = PITEM(FORB(IORB))                                         SHIP  36
      IF W(IORB) EQ 0., RETURN                                          SHIP  37
      LET NX = ISAT(J)                                                  SHIP  38
      LET NM = IMOD(J)                                                  SHIP  39
      IF EXVEH EQ 0, GO TO 60                                           SHIP  40
   32 CALL STATUS(NX,NM,7)                                              SHIP  41
      CALL DROPQ(J,IORB)                                                SHIP  42
      LET EXVEH = 0                                                     SHIP  43
      LET EXORB(IORB) = 0                                               SHIP  44
      LET TRIGS = 1                                                     SHIP  45
   30 IF ORB(IORB) IS EMPTY, RETURN                                     SHIP  46
      LET I = 1                                                         SHIP  47
      LET NL(IORB) = 0                                                  SHIP  48
```

```
      LET W(IORB) = 0                                                    SHIP  49
      DO TO 40, FOR ALL ITORB IN ORB(IORB)                               SHIP  50
      IF LQTIM(PITEM(ITORB)) LT 3000., GO TO 35                          SHIP  51
      IF IS GT 0, GO TO 40                                               SHIP  52
   35 LET NQ = I                                                         SHIP  53
      LET ILOAD(NQ) = PITEM(ITORB)                                       SHIP  54
      CALL PROP(IGO)                                                     SHIP  55
      LET EXORB(IORB) = EXVEH                                            SHIP  56
      IF W(IORB) LT 0., GO TO 50                                         SHIP  57
      LET I = I + 1                                                      SHIP  58
      IF I GT IL, GO TO 10                                               SHIP  59
   40 LOOP                                                               SHIP  60
      IF IS LE 0, GO TO 10                                               SHIP  61
      IF IS GT 0, RETURN                                                 SHIP  62
   41 LET NL(IORB) = 0                                                   SHIP  63
      LET W(IORB) = 0                                                    SHIP  64
      RETURN                                                             SHIP  65
   50 IF NL(IORB) NE 0, GO TO 10                                         SHIP  66
      IF IGO EQ 3, RETURN                                                SHIP  67
      GO TO 31                                                           SHIP  68
   60 IF RQUP(IORB) EQ 0, GO TO 32                                       SHIP  69
      IF RQSEP(IORB) NE 0, GO TO 32                                      SHIP  70
      LET EXVEH = 1                                                      SHIP  71
      LET EXORB(IORB) = 1                                                SHIP  72
      LET NL(IORB) = 0                                                   SHIP  73
      CALL STATUS(NX,NM,10)                                              SHIP  74
      GO TO 10                                                           SHIP  75
      END                                                                SHIP  76
      ENDOGENOUS EVENT START                                             START  2
C                                                                        START  3
C     THIS ROUTINE WILL INITIALIZE EACH MONTE CARLO CYCLE                START  4
CC                                                                       START  5
CC                          SET UP EVENTS FOR NEW SATELLITE LAUNCHES     START  6
C                                                                        START  7
      LET MSEP = 0                                                       START  8
      LET NEXIT = 0                                                      START  9
      DO TO 2, FOR I=(1)(SYORB)                                          START 10
      LET SATLF(I) = 0                                                   START 11
      LET LFSAT(I) = 0                                                   START 12
      LET BEGST(I) = 0                                                   START 13
      LET TLAST(I) = 0                                                   START 14
      LET SDISI(I) = 0                                                   START 15
      LET NPOS(I) = 0                                                    START 16
      IF MOD(I) IS EMPTY, GO TO 2                                        START 17
      DO TO 1, FOR ALL MODSY IN MOD(I)                                   START 18
      LET NUM  (MODSY) = 0                                               START 19
      LET LOADF(MODSY) = 0                                               START 20
      LET MSTAT(MODSY) = 0                                               START 21
    1 LOOP                                                               START 22
    2 LOOP                                                               START 23
```

```
      IF TRIG EQ 0, WRITE ON 6                                                  START 24
      FORMAT(*1*,S27,*CHRONOLOGICAL TIME HISTORY OF BASE CYCLE*/S5,*TIMESTART 25
     *        SYSTEM     STATUS     SATELLITE STATUS     MODULE     STATUSSTART 26
     *    VEHICLE    STATUS*)                                                 START 27
      DO TO 10, FOR ALL NEW IN NEWS                                           START 28
      LET IS = SCHSY(NEW)                                                     START 29
      LET DTIME(IS) = 0                                                       START 30
      LET STAT(ITSYS(IS)) = DOWN                                              START 31
      LET SSTAT(L) = DOWN, FOR L=(FSAT(ITSYS(IS)))(LSAT(ITSYS(IS)))           START 32
      CREATE NWSAT                                                            START 33
      LET PSAT(NWSAT) = SCHSY(NEW)                                            START 34
      LET PMOD(NWSAT) = 0                                                     START 35
      CAUSE NWSAT AT SCHDT(NEW)                                               START 36
   10 LOOP                                                                    START 37
      IF TRIG EQ 0, WRITE ON 6, TIME                                          START 38
      FORMAT(* *,S4,M5.2.2,S3,*START SIMULATION*,//)                          START 39
      LET VSHUT(I) = 1, FOR I=(1)(NSHUT)                                      START 40
      LET VTUG(I) = 1, FOR I=(1)(NTUG)                                        START 41
      LET SUTFY(I) = 0, FOR I=(1)(NYEAR)                                      START 42
      LET SEPFY(I) = 0, FOR I=(1)(NYEAR)                                      START 43
      LET VSEPS(I) = 1, FOR I=(1)(NSEPS)                                      START 44
      LET TUGFY(I) = 0, FOR I=(1)(NYEAR)                                      START 45
      LET CVA(I) = 0., FOR I=(1)(3)                                           START 46
      LET TGO(I) = 0., FOR I=(1)(SYORB)                                       START 47
      LET TGOSY(I) = 0., FOR I=(1)(STSTB)                                     START 48
      LET BEGSY(I) = 0., FOR I=(1)(STSTB)                                     START 49
      LET TLASY(I) = 0., FOR I=(1)(STSTB)                                     START 50
      LET SDTSY(I) = 0., FOR I=(1)(STSTB)                                     START 51
      CREATE TERM                                                             START 52
      CAUSE TERM AT 3000.                                                     START 53
      DESTROY START                                                           START 54
      LET MDCNT(I) = 0, FOR I=(1)(MITAB)                                      START 55
      LET NOWAR(I) = 0, FOR I=(1)(MITAB)                                      START 56
      LET NOFAL(I) = 0, FOR I=(1)(MITAB)                                      START 57
      LET EXORB(I) = 0, FOR I=(1)(NORBS)                                      START 58
C     ****************************************                              START 59
C  REINITIALIZE NOMOD ON ALL SATELLITES                                       START 60
C  CREATE NEWME EVENTS                                                        START 61
C     ****************************************                              START 62
      RETURN                                                                  START 63
      END                                                                     START 64
      SUBROUTINE STATUS(IS,IM,IST)                                            STATUS  2
C                                                                             STATUS  3
C  IM = 0, SATELLITE                                                          STATUS  4
C  IM = +, REPLACEABLE MODULE                                                 STATUS  5
C                                                                             STATUS  6
C  IST = 1, AVAILABLE                                                         STATUS  7
C  IST = 2, UP                                                                STATUS  8
C  IST = 3, DOWN                                                              STATUS  9
C  IST = 4, LAUNCHED                                                          STATUS10
```

```
C IST = 5, ME UPGRADE                                                    STATUS11
C IST = 6, SATELLITE RETRIEVED                                           STATUS12
C  IST = 7, PAYLOAD IS TOO HEAVY, NOT FLOWN - DROPPED FROM QUEUE        STATUS13
C  IST = 8, WARNING ON MODULE                                           STATUS14
C  IST = 9, SATELLITE REMOVED FROM ORBIT                                 STATUS15
C                                                                        STATUS16
      LET DELTA = 0                                                      STATUS17
      LET JST = ITSAT(IS)                                                STATUS18
      LET JSY = ITSYS(IS)                                                STATUS19
      IF IM NE 0, LET JMD = NOMOD(IM)                                    STATUS20
      LET HALST(IS) = TIME                                               STATUS21
      LET HALSY(JSY) = TIME                                              STATUS22
      IF IST EQ 2, LET ISTAT = UP                                        STATUS23
      IF IST EQ 3, LET ISTAT = DOWN                                      STATUS24
      IF TRIG2 EQ 1, GO TO 10                                            STATUS25
      GO TO (10,8,8,10,10,10,10,2,4,10),IST                              STATUS26
    8 IF IM EQ 0, GO TO 5                                                STATUS27
      LET MSTAT(IM) = IST                                                STATUS28
      IF IST EQ 2, GO TO 2                                               STATUS29
C                                                                        STATUS30
C   NRU FAILURE-SCHEDULE NWSAT                                           STATUS31
C                                                                        STATUS32
      CALL REDUN(IS,IM)                                                  STATUS33
      LET IK = NRU(IM)                                                   STATUS34
      IF DELTA NE 0., GO TO 111                                          STATUS35
      IF EXMOD NE 0, LET IK = EXMOD                                      STATUS36
  111 IF IK NE 100, GO TO 1                                              STATUS37
      DO TO 200, FOR ALL MODSY IN MOD(IS)                                STATUS38
      CALL QDMP(IS,MODSY,ILL)                                            STATUS39
  200 LOOP                                                               STATUS40
      LET SSTAT(IS) = OUT                                                STATUS41
C BLOCK  -  TEST LAUNCH POLICY                                           STATUS42
      IF PDOWN EQ 0, GO TO 1                                             STATUS43
      LET I = TIME + HALI3                                               STATUS44
      CALL SAVER(I,IS)                                                   STATUS45
    1 IF DELTA NE 0., GO TO 7                                            STATUS46
      IF SSTAT(IS) NE OUT, LET SSTAT(IS) = ISTAT                         STATUS47
      GO TO 7                                                            STATUS48
    2 IF SSTAT(IS) EQ OUT, GO TO 10                                      STATUS49
      GO TO 6                                                            STATUS50
    4 LET ISTAT = SSTAT(IS)                                              STATUS51
      IF NPOS(IS) EQ 0, LET ISTAT = OUT                                  STATUS52
      LET SSTAT(IS) = ISTAT                                              STATUS53
      GO TO 7                                                            STATUS54
    5 LET SSTAT(IS) = ISTAT                                              STATUS55
      LET MSTAT(MODSY) = IST, FOR ALL MODSY IN MOD(IS)                   STATUS56
      IF SSTAT(IS) EQ DOWN, LET SSTAT(IS) = OUT                          STATUS57
      IF NPOS(IS) EQ 0, LET SSTAT(IS) = OUT                              STATUS58
      IF SSTAT(IS) EQ OUT, GO TO 7                                       STATUS59
    6 DO TO 38, FOR ALL MODSY IN MOD(IS)                                 STATUS60
```

```
      IF MSTAT(MODSY) EQ 2, GO TO 38                                    STATUS61
      CALL REDUN(IS,MODSY)                                              STATUS62
      IF DELTA EQ 0., GO TO 7                                           STATUS63
   38 LOOP                                                              STATUS64
      LET SSTAT(IS) = UP                                                STATUS65
    7 LET K = 0                                                         STATUS66
      LET KK = 0                                                        STATUS67
      IF TRIG + TRIG2 EQ 1, GO TO 10                                    STATUS68
      DO TO 39, FOR I=(FSAT(JSY))(LSAT(JSY))                            STATUS69
      IF SSTAT(I) EQ UP, LET K = K + 1                                  STATUS70
      IF SSTAT(I) EQ DOWN, LET KK = KK +1                               STATUS71
   39 LOOP                                                              STATUS72
      LET IPOL = POLON(JST)                                             STATUS73
      LET IT = DOWN                                                     STATUS74
      IF IPOL EQ 0, LET IT = OUT                                        STATUS75
      IF IPOL EQ 1, LET IT = OUT                                        STATUS76
      IF IPOL EQ 4, LET IT = OUT                                        STATUS77
      IF TIME GE TGOSY(JSY), LET IT = OUT                               STATUS78
      IF K NE 0, LET IT = DOWN                                          STATUS79
      IF KK NE 0, LET IT = DOWN                                         STATUS80
      LET STAT(JSY)= IT                                                 STATUS81
      IF K GE NEUP(JSY), LET STAT(JSY) = UP                             STATUS82
   10 LET IP = IS - FSAT(JSY) + 1                                       STATUS83
      IF TRIG + TRIG2 EQ 1, GO TO 54                                    STATUS84
      IF TLAST(IS) EQ 0., GO TO 54                                      STATUS85
      IF SSTAT(IS) EQ UP, GO TO 51                                      STATUS86
      IF TLAST(IS) LT 0., GO TO 52                                      STATUS87
      LET SDIST(IS) = SDIST(IS) + TIME - TLAST(IS)                      STATUS88
      LET TLAST(IS) = -TIME                                             STATUS89
      GO TO 52                                                          STATUS90
   51 IF TLAST(IS) GT 0., GO TO 52                                      STATUS91
      LET A = TIME + TLAST(IS)                                          STATUS92
      LET TLAST(IS) = TIME                                              STATUS93
      IF A EQ 0., GO TO 52                                              STATUS94
      LET ONTST(IS) = ONTST(IS) + A                                     STATUS95
      LET C223(IS) = C223(IS) + 1.                                      STATUS96
      IF  N223(IS) GT A, LET N223(IS) = A                               STATUS97
      IF  X223(IS) LT A, LET X223(IS) = A                               STATUS98
   52 LET IY = JSY                                                      STATUS99
      IF STAT(IY) EQ UP, GO TO 53                                       STATU100
      IF TLASY(IY) LT 0., GO TO 54                                      STATU101
      LET SDISY(IY) = SDISY(IY) + TIME - TLASY(IY)                      STATU102
      LET TLASY(IY) = -TIME                                             STATU103
      GO TO 54                                                          STATU104
   53 IF TLASY(IY) GT 0., GO TO 54                                      STATU105
      LET A = TIME + TLASY(IY)                                          STATU106
      LET TLASY(IY) = TIME                                              STATU107
      IF A EQ 0., GO TO 54                                              STATU108
      LET ONTSY(IY) = ONTSY(IY) + A                                     STATU109
      LET C208(IY) = C208(IY) + 1.                                      STATU110
```

```
      IF  N208(IY) GT A, LET N208(IY) = A                                    STATU111
      IF  X208(IY) LT A, LET X208(IY) = A                                    STATU112
   54 IF TRIG NE 0, RETURN                                                   STATU113
      IF TIME LT TIMEB,  RETURN                                              STATU114
      IF TRIG2 EQ 0, CALL FILES(IS,IM,IST)                                   STATU115
      LET NSY = SYNAM(JSY)                                                   STATU116
      LET NSS = STAT(JSY)                                                    STATU117
      LET KST = SNAME(JST)                                                   STATU118
      LET KSS = SSTAT(IS)                                                    STATU119
      IF IM EQ 0, GO TO (11,12,12,14,14,16,17,17,18,19), IST                 STATU120
      LET MST = MNAME(JMD)                                                   STATU121
      GO TO (21,22,22,24,25,25,27,26,22,29),IST                              STATU122
   11 WRITE ON 6,TIME,NSY,IP,NSS,KST                                         STATU123
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*AVAILABLE*)                  STATU124
      RETURN                                                                 STATU125
   12 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS                                     STATU126
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6)                           STATU127
      RETURN                                                                 STATU128
   14 WRITE ON 6,TIME,NSY,IP,NSS,KST                                         STATU129
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*LAUNCHED*)                   STATU130
      RETURN                                                                 STATU131
   16 WRITE ON 6,TIME,NSY,IP,NSS,KST                                         STATU132
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*RETRIEVED*)                  STATU133
      RETURN                                                                 STATU134
   17 WRITE ON 6,TIME,NSY,IP,BLANK,KST                                       STATU135
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*PAYLOAD TOO HEAVY  +++++STATU136
     *++++++++++++++++++++++++++++++++++++++++++++++++++*)                   STATU137
      RETURN                                                                 STATU138
   21 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST                                 STATU139
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,*AVAILABLE*)      STATU140
      RETURN                                                                 STATU141
   22 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST,ISTAT                           STATU142
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,A6)               STATU143
      RETURN                                                                 STATU144
   24 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST                                 STATU145
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,*LAUNCHED*)       STATU146
      RETURN                                                                 STATU147
   25 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST                                 STATU148
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,*ME UPGRADE*)STATU149
      RETURN                                                                 STATU150
   26 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST                                 STATU151
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,*WARNING*)        STATU152
      RETURN                                                                 STATU153
   18 WRITE ON 6,TIME,NSY,IP,NSS,KST                                         STATU154
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*REMOVED*)                    STATU155
      RETURN                                                                 STATU156
   19 WRITE ON 6,TIME,NSY,IP,NSS,KST                                         STATU157
      FORMAT(S5,M5.2.2,S3,A6,I3,S1,A6,S4,A6,S4,*SATELLITE REQUIRES EXPENSTATU158
     *DED VEHICLE-------------------------------*)                          STATU159
      RETURN                                                                 STATU160
```

```
   27 WRITE ON 6,TIME,NSY,IP,BLANK,KST,BLANK,MST                          STATU161
      FORMAT(S5,M5.2,2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,S4,*PAYLOAD TOO   STATU162
     *HEAVY ++++++++++++++++++++++++++++++++++++++++++++++++*)            STATU163
      RETURN                                                             STATU164
   29 WRITE ON 6,TIME,NSY,IP,NSS,KST,KSS,MST                             STATU165
      FORMAT(S5,M5.2,2,S3,A6,I3,S1,A6,S4,A6,S4,A6,S4,A6,*MODULE REQUIRESSTATU166
     * EXPENDED VEHICLE--------------------*)                           STATU167
      RETURN                                                             STATU168
      END                                                                STATU169
      ENDOGENOUS EVENT TERM                                              TERM   2
C                                                                        TERM   3
CCC   THIS ROUTINE WILL BE ACTIVATED AT THE END OF A MONTE CARLO CYCLE   TERM   4
CCC                                                                      TERM   5
CCC   IT MAY RESTART THE PROGRAM FOR THE NEXT CYCLE OR CAUSE THE         TERM   6
CCC                                                                      TERM   7
CCC   TERMINATION OF THE RUN WITH STATISTICS.                            TERM   8
C                                                                        TERM   9
      IF TRIG EQ 0, WRITE ON 6,TIME                                      TERM  10
      FORMAT(/,S5,M5.2,2,S3,*TERMINATE SIMULATION*)                      TERM  11
      IF TRIG2 EQ 0, CALL FILEO                                          TERM  12
C                                                                        TERM  13
C  GATHER MONTE CARLO END OF CYCLE STATISTICS FOR VEHICLES/SATELLITES    TERM  14
C                                                                        TERM  15
      LET TRIG = TRIG + 1                                                TERM  16
      IF LOAD IS NOT EMPTY, GO TO 1                                      TERM  17
   10 CALL MCVEH                                                         TERM  18
      CALL MCMOD                                                         TERM  19
      CALL MCSAT                                                         TERM  20
      CALL MCSYS                                                         TERM  21
      IF TRIG GE TRIGS, GO TO 5                                          TERM  22
C                                                                        TERM  23
CCC   INITIALIZE  ANOTHER CYCLE                                          TERM  24
C                                                                        TERM  25
      CREATE START                                                       TERM  26
      LET TIME = 0.                                                      TERM  27
      CAUSE START AT 1.                                                  TERM  28
      DESTROY TERM                                                       TERM  29
      RETURN                                                             TERM  30
C                                                                        TERM  31
CCC   LOADING QUEUE CONTAINS TRASH -- STOP RUN                           TERM  32
C                                                                        TERM  33
    1 DO TO 20, FOR ALL PAYLD IN LOAD                                    TERM  34
      IF LQTIM(PAYLD) GT 3000., CALL QDMP(ISAT(PAYLD),IMOD(PAYLD),ILL)   TERM  35
   20 LOOP                                                               TERM  36
      IF LOAD IS EMPTY, GO TO 10                                         TERM  37
      WRITE ON 6                                                         TERM  38
      FORMAT(S5,*---RUN STOPPED DUE TO DATA IN LOADING QUEUE AT END OF CTERM  39
     *YCLE*)                                                             TERM  40
      DO TO 2, FOR ALL PAYLD IN LOAD                                     TERM  41
      LET I = SNAME(IISAT(ISAT(PAYLD)))                                  TERM  42
```

```
      LET A = LQTIM(PAYLD)                                                    TERM  43
      IF IMOD(PAYLD) EQ 0, WRITE ON 6,I,A                                     TERM  44
      FORMAT(S5,*SATELLITE - *,A6,* SINCE *,M5.2.2)                           TERM  45
      IF IMOD(PAYLD) NE 0, WRITE ON 6,MNAME(NOMOD(IMOD(PAYLD))),I,A           TERM  46
      FORMAT(S5,*MODULE - *,A6,* ON SATELLITE - *,A6,* SINCE *,M5.2.2)        TERM  47
    2 LOOP                                                                    TERM  48
      LET TRIGS = 1                                                           TERM  49
      GO TO 10                                                                TERM  50
C                                                                             TERM  51
CCC   FINAL OUTPUT                                                            TERM  52
C                                                                             TERM  53
    5 CALL TERMO                                                              TERM  54
      STOP                                                                    TERM  55
      END                                                                     TERM  56
      SUBROUTINE TERMO                                                        TERMO  2
C                                                                             TERMO  3
CCC   OUTPUT STATISTICS AT END OF RUN                                         TERMO  4
C                                                                             TERMO  5
      WRITE ON 6,TRIGS                                                        TERMO  6
      FORMAT(*1      STATISTICAL SUMMARY FOR*,I4,* MONTE CARLO CYCLES*/TERMO  7
     */)                                                                      TERMO  8
      LET A = TRIGS                                                           TERMO  9
      WRITE ON 6                                                              TERMO 10
      FORMAT(S25,*FLIGHT SUMMARY*/S18,*SHUTTLE*,S15,*TUG*,S17,*SEPS*/         TERMO 11
     **    YEAR     MIN   AVG   MAX      MIN   AVG   MAX      MIN   AVG     TERMO 12
     *MAX*)                                                                   TERMO 13
      DO TO 10, FOR I=(1)(NYEAR)                                              TERMO 14
      LET II = I - 1                                                          TERMO 15
      LET TUGFY(I) = TIMEB + II                                               TERMO 16
      IF MAX90(I) EQ 0, GO TO 10                                              TERMO 17
      LET B = SUM39(I)                                                        TERMO 18
      LET B = B/A                                                             TERMO 19
      LET C = SUM90(I)                                                        TERMO 20
      LET C = C/A                                                             TERMO 21
      LET D = SUM86(I)                                                        TERMO 22
      LET D = D/A                                                             TERMO 23
      WRITE ON 6,TUGFY(I),MIN90(I),C,MAX90(I),MIN39(I),B,MAX39(I),            TERMO 24
     * MIN86(I),D,MAX86(I)                                                    TERMO 25
      FORMAT(I8,I8,D4.1,I6,I8,D4.1,I6,I8,D4.1,I6)                             TERMO 26
   10 LOOP                                                                    TERMO 27
      LET B = ITFLT                                                           TERMO 28
      LET B = B/A                                                             TERMO 29
      LET C = IFSUT                                                           TERMO 30
      LET C = C/A                                                             TERMO 31
      LET D = IFSEP                                                           TERMO 32
      LET D = D/A                                                             TERMO 33
      WRITE ON 6,NFSUT,C,MFSUT,NTFLI,B,MTFLT,NFSEP,D,MFSEP                    TERMO 34
      FORMAT(* PROGRAM*,I8,D4.1,I6,I8,D4.1,I6,I8,D4.1,I6)                     TERMO 35
      DO TO 5, FOR I=(1)(3)                                                   TERMO 36
      IF MID(I) EQ 1000., LET MID(I) = 0.                                     TERMO 37
```

```
      IF MCVA(I) EQ 1000., LET MCVA(I) = 0.                                    TERMO 38
      IF TCVA(I) EQ 0., GO TO 5                                                TERMO 39
      LET VTO(I) = VTO(I)*360./TCVA(I)                                         TERMO 40
      LET TCVA(I) = TCVA(I)/A                                                  TERMO 41
      LET MTO(I) = MTO(I)*360.                                                 TERMO 42
      LET XTO(I) = XTO(I)*360.                                                 TERMO 43
      IF I EQ 1, LET E = C                                                     TERMO 44
      IF I EQ 2, LET E = B                                                     TERMO 45
      IF I EQ 3, LET E = D                                                     TERMO 46
      LET TCVA(I) = TCVA(I)*100./E                                             TERMO 47
      LET MCVA(I) = MCVA(I)*100./E                                             TERMO 48
      LET XCVA(I) = XCVA(I)*100./E                                             TERMO 49
    5 LOOP                                                                     TERMO 50
      WRITE ON 6,MCVA(1),TCVA(1),XCVA(1),MCVA(2),TCVA(2),XCVA(2),              TERMO 51
     * MCVA(3),TCVA(3),XCVA(3)                                                 TERMO 52
      FORMAT(*0PERCENT*,D6.1,2D4.1,D6.1,2D4.1,D6.1,2D4.1)                      TERMO 53
      WRITE ON 6,MTO(1),VTO(1),XTO(1),MTO(2),VTO(2),XTO(2),                    TERMO 54
     * MTO(3),VTO(3),XTO(3)                                                    TERMO 55
      FORMAT(*0 DELAY *,D6.1,2D4.1,D6.1,2D4.1,D6.1,2D4.1)                      TERMO 56
      LET EXTUG = EXTUG/A                                                      TERMO 57
      IF EXTUG NE 0., WRITE ON 6,EXTUG                                         TERMO 58
      FORMAT(*0     AVERAGE NO. OF EXPENDED TUGS = *,D5.1)                     TERMO 59
      WRITE ON 6                                                               TERMO 60
      FORMAT(*1*,S30,*ORBIT TRAFFIC SUMMARY*/*0*,S13,*AVERAGE FLIGHTS*,STERMO 61
     *15,*AVERAGE UP WEIGHT*,S9,*SHUTTLE ONLY*/S3,*ORBIT   SHUTTLE    TUTERMO 62
     *G    SEPS      SHUTTLE      TUG      SEPS    LOAD FACTOR*/S1)          TERMO 63
      DO TO 30, FOR I=(1)(NORBS)                                               TERMO 64
      IF ORBID(I) EQ 0, GO TO 30                                               TERMO 65
      IF WSHUT(I) NE 0., LET WSHUT(I) = WSHUT(I)/CSHUT(I)                      TERMO 66
      IF WSEPS(I) NE 0., LET WSEPS(I) = WSEPS(I)/CSEPS(I)                      TERMO 67
      IF WTUG (I) NE 0., LET WTUG (I) = WTUG (I)/CTUG (I)                      TERMO 68
      LET CSHUT(I) = CSHUT(I)/A                                                TERMO 69
      LET CSEPS(I) = CSEPS(I)/A                                                TERMO 70
      LET CTUG (I) = CTUG (I)/A                                                TERMO 71
      LET J = RQSUT(IORB)                                                      TERMO 72
      IF J EQ 0, LET J = 1                                                     TERMO 73
      LET B = WSHUT(I)/WCONV(J)                                                TERMO 74
      WRITE ON 6,ORBID(I),CSHUT(I),CTUG(I),CSEPS(I),WSHUT(I),WTUG(I),          TERMO 75
     * WSEPS(I),B                                                             TERMO 76
      FORMAT(S3,A6,3D6.1,D12,2D9,D9.2)                                         TERMO 77
   30 LOOP                                                                     TERMO 78
      WRITE ON 6                                                               TERMO 79
      FORMAT(*1*)                                                              TERMO 80
      LET TSATS = 0.                                                           TERMO 81
      LET EQSAT = 0.                                                           TERMO 82
      DO TO 13, FOR I=(1)(SISTB)                                               TERMO 83
      IF SYNAM(I) EQ 0, GO TO 13                                               TERMO 84
      IF FSAT(I) EQ 0, GO TO 13                                                TERMO 85
      WRITE ON 6,SYNAM(I)                                                      TERMO 86
      FORMAT(*0   STATISTICS FOR SYSTEM - *,A6)                                TERMO 87
```

```
      DO TO 12, FOR J=(FSAT(I))(LSAT(I))                                    TERMO 88
      LET TRES = 0.                                                         TERMO 89
      LET ICEQ = 0                                                          TERMO 90
      IF SORTE(ITSAT(J)) NE 0., GO TO 110                                   TERMO 91
      IF MOD(J) IS EMPTY, GO TO 12                                          TERMO 92
      WRITE ON 6                                                            TERMO 93
      FORMAT(*0 MODULE    MIN   AVG   MAX  MIN FLT  AVG FLT  MAX FLT*)     TERMO 94
      DO TO 11, FOR ALL MODSY IN MOD(J)                                     TERMO 95
      LET B = SUMNU(MODSY)                                                  TERMO 96
      LET B = B/A                                                           TERMO 97
      IF NRU(MODSY) NE 100, LET ICEQ = ICEQ +1                              TERMO 98
      LET TRES = TRES + B                                                   TERMO 99
      LET D = MINLF(MODSY)                                                  TERMO100
      LET E = SUMLF(MODSY)                                                  TERMO101
      LET E = E/A                                                           TERMO102
      LET F = MAXLF(MODSY)                                                  TERMO103
      LET D = D/100.                                                        TERMO104
      LET E = E/100.                                                        TERMO105
      LET F = F/100.                                                        TERMO106
      IF MAXNU(MODSY) EQ 0, WRITE ON 6, MNAME(NOMOD(MODSY))                 TERMO107
     *,NRU(MODSY)                                                           TERMO108
      FORMAT(S3,A6,I3)                                                      TERMO109
      IF MAXNU(MODSY) NE 0, WRITE ON 6, MNAME(NOMOD(MODSY)),                TERMO110
     *NRU(MODSY),                                                           TERMO111
     * MINNU(MODSY),B,                                                     TERMO112
     * MAXNU(MODSY),D,E,F                                                  TERMO113
      FORMAT(S3,A6,2I3,D4.1,I6,3D5.2)                                       TERMO114
   11 LOOP                                                                  TERMO115
  110 LET S227(J) = S227(J)/A                                               TERMO116
      LET B = NDEP(J)                                                       TERMO117
      LET B = B/A                                                           TERMO118
      WRITE ON 6,SNAME(ITSAT(J)),B,N227(J),S227(J),X227(J)                  TERMO119
      FORMAT(*  SATELLITE*/S3,A6,S6,D4.1,S6,3D5.2)                          TERMO120
      IF SORTE(ITSAT(J)) NE 0., GO TO 12                                    TERMO121
      LET TSATS = TSATS + B                                                 TERMO122
      LET E = ICEQ                                                          TERMO123
      IF E NE 0., LET TRES = TRES/E                                         TERMO124
      LET TRES =TRES+B                                                      TERMO125
      WRITE ON 6, TRES                                                      TERMO126
      FORMAT(*   EQ SAT*,S6,D4.2)                                          TERMO127
      LET EQSAT = EQSAT + TRES                                              TERMO128
      LET SUMSL(J) = SUMSL(J)/A                                             TERMO129
      WRITE ON 6,MINSL(J),SUMSL(J),MAXSL(J)                                 TERMO130
      FORMAT(*0 SATELLITE TOTAL FLIGHTS  *,3D5.2)                           TERMO131
      IF C223(J) EQ 0., LET C223(J) = 1.                                    TERMO132
      LET E = PERST(J) /A                                                   TERMO133
      LET E = DNTST(J) *360./C223(J)                                        TERMO134
      IF N223(J) EQ 1000., LET N223(J) = 0.                                 TERMO135
      LET N223(J) = N223(J)*360.                                            TERMO136
      LET X223(J) = X223(J)*360.                                            TERMO137
```

```
      WRITE ON 6,N216(J),F,X216(J)                                      TERM0138
      FORMAT(*0 PERCENT SATELLITE AVAIL. *,3D5.2)                       TERM0139
      WRITE ON 6,N223(J),E,X223(J)                                      TERM0140
      FORMAT(*0 DELAY INTERVAL TO RESTORE*,3D5.2)                       TERM0141
   12 LOOP                                                              TERM0142
      LET SYLF(I) = SYLF(I)/A                                           TERM0143
      WRITE ON 6,NSYLF(I),SYLF(I),XSYLF(I)                              TERM0144
      FORMAT(*0 SYSTEM TOTAL FLIGHTS     *,3D5.2)                       TERM0145
      IF FSAT(I) EQ 0, GO TO 13                                         TERM0146
      LET E = PERSY(I)/A                                                TERM0147
      IF C208(I) EQ 0., LET C208(I) = 1.                                TERM0148
      LET E = DNTSY(I)*360./C208(I)                                     TERM0149
      IF N208(I) EQ 1000., LET N208(I) = 0.                             TERM0150
      LET N208(I) = N208(I)*360.                                        TERM0151
      LET X208(I) = X208(I)*360.                                        TERM0152
      IF X200(I) EQ 0., GO TO 16                                        TERM0153
      WRITE ON 6,N200(I),F,X200(I)                                      TERM0154
      FORMAT(*0 PERCENT SYSTEM AVAILABLE *,3D5.2)                       TERM0155
      WRITE ON 6,N208(I),E,X208(I)                                      TERM0156
      FORMAT(*0 DELAY INTERVAL TO RESTORE*,3D5.2)                       TERM0157
   16 WRITE ON 6                                                        TERM0158
      FORMAT(*-------------------------------------------------*)       TERM0159
   13 LOOP                                                              TERM0160
      WRITE ON 6, TSATS,EQSAT                                           TERM0161
      FORMAT(*0    TOTAL SATELLITES *,D4.1/*0    EQU.  SATELLITES *,D4.2TERM0162
     *)                                                                 TERM0163
      WRITE ON 6                                                        TERM0164
      FORMAT(*1    MODULE SUMMARY*//S20,*HARN*,S24,*FAIL*,S22,*REPLACE*/TERM0165
     **  NAME      MIN       AVR       MAX       MIN       AVR       MAXTERM0166
     *IN       AVR       MAX*/)                                         TERM0167
      DO TO 15, FOR I=(1)(MITAB)                                        TERM0168
      IF MNAME(I) EQ 0, GO TO 15                                        TERM0169
      IF S121(I) + S125(I) + S129(I) EQ 0, GO TO 14                     TERM0170
      LET B = S121(I)                                                   TERM0171
      LET B = B/A                                                       TERM0172
      LET C = S125(I)                                                   TERM0173
      LET C = C/A                                                       TERM0174
      LET D = S129(I)                                                   TERM0175
      LET D = D/A                                                       TERM0176
      IF N125(I) EQ 1000, LET N125(I) = 0                               TERM0177
      IF N129(I) EQ 1000, LET N129(I) = 0                               TERM0178
      IF N121(I) EQ 1000, LET N121(I) = 0                               TERM0179
      WRITE ON 6,MNAME(I),N125(I),C,X125(I),N129(I),D,X129(I),N121(I),B,TERM0180
     *X121(I)                                                           TERM0181
      FORMAT(S2,A6,I6,D7.1,2I9,D7.1,2I9,D7.1,I9)                        TERM0182
      GO TO 15                                                          TERM0183
   14 WRITE ON 6,MNAME(I)                                               TERM0184
      FORMAT(S2,A6)                                                     TERM0185
   15 LOOP                                                              TERM0186
      RETURN                                                            TERM0187
```

```
      END                                                                TERMO188
      ENDOGENOUS EVENT WARN                                              WARN    2
C                                                                        WARN    3
C     THIS ROUTINE WILL ATTEMPT TO SCHEDULE THE LAUNCHING OF A REPLACEMENT WARN    4
C                                                                        WARN    5
C     MODULE.  IF SUCCESSFUL, THE CORRESPONDING FAILURE MUST BE BLOCKED  WARN    6
C                                                                        WARN    7
C     IF IT EXISTS                                                       WARN    8
C                                                                        WARN    9
      LET IS = PSAT(WARN)                                                WARN   10
      LET IM = PMOD(WARN)                                                WARN   11
      IF SSTAT(IS) EQ OUT, RETURN                                        WARN   12
      LET NOWAR(NOMOD(IM)) = NOWAR(NOMOD(IM)) + 1                        WARN   13
      CALL STATUS(IS,IM,B)                                               WARN   14
      LET DELAY = WSATU                                                  WARN   15
      IF TIME + DELAY GT TGO(IS), RETURN                                 WARN   16
      CALL SHIP(IS,IM)                                                   WARN   17
      CALL REDUN(IS,IM)                                                  WARN   18
      IF DELTA GT 0., RETURN                                             WARN   19
      CREATE LAUNC                                                       WARN   20
      LET PSAT(LAUNC) = IS                                               WARN   21
      LET PMOD(LAUNC) = IM                                               WARN   22
      CAUSE LAUNC AT TIME + DELAY                                        WARN   23
      RETURN                                                             WARN   24
      END                                                                WARN   25
      SUBROUTINE WEIBUL (AW,BW,TW,AF,BF,TF)                              WEIBUL  2
C                                                                        WEIBUL  3
C     WEIBUL FUNCTION FOR FAILURE AND WARNING TIMES                      WEIBUL  4
C                                                                        WEIBUL  5
      LET TW = 0.                                                        WEIBUL  6
      IF AW EQ 0., GO TO 5                                               WEIBUL  7
      IF TIMEC EQ 0., GO TO 1                                            WEIBUL  8
      LET AX = TIMEC                                                     WEIBUL  9
      GO TO 2                                                            WEIBUL10
    1 LET AX = RANF(N)                                                   WEIBUL11
    2 LET AX = -ALOG(AX)                                                 WEIBUL12
      IF BW NE 1., LET AX = AX**(1./BW)                                  WEIBUL13
      LET TW = AW*AX                                                     WEIBUL14
      LET TF = 0.                                                        WEIBUL15
      IF AF EQ 0., RETURN                                                WEIBUL16
      LET AX = TW/AF                                                     WEIBUL17
      IF BF NE 1., LET AX = AX**BF                                       WEIBUL18
      LET AN3 = EXP(-AX)                                                 WEIBUL19
      IF TIMEC EQ 0., GO TO 3                                            WEIBUL20
      LET AX = TIMEC                                                     WEIBUL21
      GO TO 4                                                            WEIBUL22
    3 LET AX = RANF(N)                                                   WEIBUL23
    4 LET AX = -ALOG(AX*AN3)                                             WEIBUL24
      IF BF NE 1., LET AX = AX**(1./BF)                                  WEIBUL25
      LET TF = AF*AX                                                     WEIBUL26
```

```
      RETURN                                                            WEIBUL27
    5 LET TF = 0.                                                       WEIBUL28
      IF AF EQ 0., RETURN                                               WEIBUL29
      IF TIMEC EQ 0., GO TO 6                                           WEIBUL30
      LET AX = TIMEC                                                    WEIBUL31
      GO TO 7                                                           WEIBUL32
    6 LET AX = RANF(N)                                                  WEIBUL33
    7 LET AX = -ALOG(AX)                                                WEIBUL34
      IF BF NE 1., LET AX = AX**(1./BF)                                 WEIBUL35
      LET TF = AF*AX                                                    WEIBUL36
      RETURN                                                            WEIBUL37
      END                                                               WEIBUL38
      SUBROUTINE PRFORM(DVLEG,PLEG,NLEG,WPER,NEXIT,ERFLG,NT)            PRFORM  3
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),MS(3),WPA(3),EISP(3)              PRFORM  4
     X             ,REUSE(3),WGA,TR                                   PRFORM  5
     X,FEAS(2)                                                          PRFORM  6
      COMMON/MISC/G                                                     PRFORM  7
      COMMON/QUIP/  ID,TU,HCO,ICOS,MPT,TLEFT,MDT,NTUGS                  PRFORM  8
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       PRFORM  9
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV               PRFORM10
      COMMON/C1/MISSN                                                   PRFORM11
      DIMENSION DVLEG(10),PLEG(10)                                      PRFORM12
      REAL MPT,MDT                                                      PRFORM13
      INTEGER ERFLG                                                     PRFORM14
      INTEGER SEPS,SPAR                                                 PRFORM15
      REAL MS,MPLA,MPLB,MPLS                                            PRFORM16
      NAMELIST/HELL/DVLEG,PLEG,NLEG,WPER,NEXIT,ERFLG,NTUGS,TYPE,NSTG,   PRFORM17
     * SPAR,MS,WPA,EISP,REUSE,WGA,TR,G,ID,TU,HCO,ICOS,MPT,TLEFT,MDT,     PRFORM18
     * SEPS,MS,E,P,SISP,SEPK,SR,TSEP,NSERV,DTHETA,MPLS,PSERV,VSERV       PRFORM19
     *,FEAS                                                             PRFORM20
     *,MPLA,MPLB                                                        PRFORM21
      DATA TYPE/10HEXP XSTAGE/                                          PRFORM22
      DATA NSERV/0/                                                     PRFORM23
      DATA PSERV,VSERV/86165.,3074.66/                                  PRFORM24
      DATA G/32.1725/                                                   PRFORM25
      IF (NSTG .LT. 0)  STOP                                            PRFORM26
C                                                                       PRFORM27
C             PERF - SETS UP AND CHOOSES THE SPECIFIC                   PRFORM28
C                    PERFORMANCE SUBROUTINE TO BE EXECUTED              PRFORM29
C           SSHOT - SLINGSHOT - LIQUID UPPERS                           PRFORM30
C           SSLQD - SINGLE STAGE LIQUID                                 PRFORM31
C           TRNKC - TRANS KICK - SOLID UPPERS                           PRFORM32
C           SEPSIM- SEPS SIMULATOR                                      PRFORM33
C                                                                       PRFORM34
      IF ( SEPS .NE. 0 )  GO TO 40                                      PRFORM35
      IF ( NSTG .GT. 1 )  GO TO 10                                      PRFORM36
      CALL  SSLQD (DVLEG,PLEG,NLEG)                                     PRFORM37
      GO TO 50                                                          PRFORM38
   10 DO 20  I = 2,NSTG                                                 PRFORM39
      IF (SPAR(I) .NE. 0 ) GO TO 30                                     PRFORM40
```

```
   20 CONTINUE                                                          PRFORM41
      CALL  SSHOT (DVLEG,PLEG,NLEG)                                     PRFORM42
      GO TO 50                                                          PRFORM43
   30 CALL TRNKC(DVLEG,PLEG)                                            PRFORM44
      GO TO 50                                                          PRFORM45
   40 MISSN = MISSN + 1                                                 PRFORM46
      MPLA = PLEG(1)                                                    PRFORM47
      MPLB = PLEG(NLEG)                                                 PRFORM48
      CALL SEPX  (MPLA, MPLB,ERFLG,NEXIT )                              PRFORM49
      IK = 6                                                            PRFORM50
      NT = NTUGS                                                        PRFORM51
      IF (NEXIT.EQ.1) RETURN                                            PRFORM52
      IF(NEXIT.EQ.2) RETURN                                             PRFORM53
      IF(NEXIT.EQ.5) RETURN                                             PRFORM54
      IF(NEXIT.EQ.6) RETURN                                             PRFORM55
      RETURN                                                            PRFORM56
   50 WPER = 100.*(1.-AMAX1(FEAS(1),FEAS(2)))                           PRFORM57
      RETURN                                                            PRFORM58
      END                                                               PRFORM59
      SUBROUTINE CONEC(NS,NVEH,ISESP)                                   CONEC   2
C                                                                       CONEC   3
C     THIS ROUTINE WILL GET THE NECESSARY VEHICLE DATA                  CONEC   4
C                                                                       CONEC   5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       CONEC   6
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              CONEC   7
     X             ,REUSE(3),WGA,TR                                   CONEC   8
      INTEGER SEPS                                                      CONEC   9
      IF(NSTG.EQ.0) NSTG = 1                                            CONEC  10
      NSTG = NS                                                         CONEC  11
      SEPS = ISESP                                                      CONEC  12
      RETURN                                                            CONEC  13
      END                                                               CONEC  14
      SUBROUTINE LINKT(I,A,B,C,D,E,JF,G)                                LINKT   2
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              LINKT   3
     X             ,REUSE(3),WGA,TR                                   LINKT   4
      INTEGER SPAR                                                      LINKT   5
      EISP(I) = A                                                       LINKT   6
      WS(I) = B + C                                                     LINKT   7
      WPA(I) = D                                                        LINKT   8
      REUSE(I) = 1. - E                                                 LINKT   9
    5 SPAR(I) = JF                                                      LINKT  10
      WGA = G                                                           LINKT  11
      RETURN                                                            LINKT  12
      END                                                               LINKT  13
      SUBROUTINE  SSLQD (DVLEG,PLEG,NLEG)                               SSLQD   2
C                                                                       SSLQD   3
C           SSLQD - PERFORMANCE ROUTINE FOR SINGLE STAGE LIQUID         SSLQD   4
C                                                                       SSLQD   5
C     GENERAL INPUT                                                     SSLQD   6
C          WS    THE STRUCTURE WEIGHT FOR THE STAGES                  SSLQD   7
```

```
C                 WPA   THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES   SSLQD   8
C                 EISP  EFFECTIVE ISP  (SEC)                            SSLQD   9
C                 G     GRAVITY  (CONSTANT)                             SSLQD  10
C                 WGA   ALLOWABLE GROSS WEIGHT                          SSLQD  11
C                 NSTG  NUMBER OF STAGES                                SSLQD  12
C                 REUSE REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE    SSLQD  13
C                                                                       SSLQD  14
C           SPECIFIC INPUT                                              SSLQD  15
C                 DVLEG DELTA V FOR EACH LEG                            SSLQD  16
C                 PLEG PAYLOAD ON EACH LEG                              SSLQD  17
C                 NLEG NUMBER OF LEGS                                   SSLQD  18
C                                                                       SSLQD  19
C           OUTPUT                                                      SSLQD  20
C                 FEAS(1)    PROPELLENT WEIGHT RATIO                  SSLQD  21
C                 FEAS(2)    GROSS WEIGHT RATIO                       SSLQD  22
C                                                                       SSLQD  23
C                 IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED  SSLQD  24
C                 IF  GREATER THEN 1 CONSTRAINTS EXCEEDED               SSLQD  25
C                                                                       SSLQD  26
      DIMENSION DVLEG(10),PLEG(10)                                      SSLQD  27
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)              SSLQD  28
     X             ,REUSE(3),WGA,IR                                     SSLQD  29
     X,FEAS(2)                                                          SSLQD  30
      COMMON/MISC/G                                                     SSLQD  31
      REAL        MR                                                    SSLQD  32
      WP = 0.0                                                          SSLQD  33
      N = NLEG                                                          SSLQD  34
      DN1 = G* EISP(1)                                                  SSLQD  35
      DO  10  I = 1,NLEG                                                SSLQD  36
      EXP1 = DVLEG(N) / DN1                                             SSLQD  37
      MR =  EXP( EXP1)                                                  SSLQD  38
      WPI = (WS(1)+ WP + PLEG(N) ) * (MR - 1.0)                         SSLQD  39
      WP =  WP +  WPI                                                   SSLQD  40
   10 N = N - 1                                                         SSLQD  41
C                                                                       SSLQD  42
C                 OK  - NOW HAVE WEIGHT FOR LEG                         SSLQD  43
C                                                                       SSLQD  44
      WG = WP + WS(1) + PLEG(1)                                         SSLQD  45
      FEAS(1) =  WP / WPA(1)                                            SSLQD  46
      FEAS(2) =  WG / WGA                                               SSLQD  47
      RETURN                                                            SSLQD  48
      END                                                               SSLQD  49
      SUBROUTINE  SSHOT (DVLEG,PLEG,NLEG)                               SSHOT  2
C                                                                       SSHOT  3
C           SSHOT - PERFORMANCE ROUTINE FOR THE SLINGSHOT TYPE          SSHOT  4
C                   DEPLOYMENT - UP TO 3 LEGS AND EITHER  2 OR          SSHOT  5
C                   3 STAGES.                                           SSHOT  6
C                                                                       SSHOT  7
C           GENERAL INPUT                                               SSHOT  8
C                 WS    THE STRUCTURE WEIGHT FOR THE STAGES             SSHOT  9
```

```
C                   WPA   THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES   SSHOT 10
C                   EISP  EFFECTIVE ISP  (SEC)                            SSHOT 11
C                   G     GRAVITY  (CONSTANT)                             SSHOT 12
C                   WGA   ALLOWABLE GROSS WEIGHT                          SSHOT 13
C                   NSTG  NUMBER OF STAGES                                SSHOT 14
C                   REUSE REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE    SSHOT 15
C                                                                         SSHOT 16
C           SPECIFIC INPUT                                                SSHOT 17
C                   DVLEG DELTA V FOR EACH LEG                            SSHOT 18
C                   PLEG PAYLOAD ON EACH LEG                              SSHOT 19
C                   NLEG NUMBER OF LEGS                                   SSHOT 20
C                                                                         SSHOT 21
C           OUTPUT                                                        SSHOT 22
C                   FEAS(1)     PROPELLENT WEIGHT RATIO                   SSHOT 23
C                   FEAS(2)     GROSS WEIGHT RATIO                        SSHOT 24
C                                                                         SSHOT 25
C                   IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED  SSHOT 26
C                   IF  GREATER THEN 1 CONSTRAINTS EXCEEDED               SSHOT 27
C                                                                         SSHOT 28
      DIMENSION DVLEG(10),PLEG(10),WP(3)                                  SSHOT 29
      COMMON/FUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)                SSHOT 30
     X             ,REUSE(3),WGA,TR                                       SSHOT 31
     X,FEAS(2)                                                           SSHOT 32
      COMMON/MISC/G                                                       SSHOT 33
      REAL        MR                                                      SSHOT 34
C                                                                         SSHOT 35
C                   INITILIZE AND COMPUTE STAGE WT                        SSHOT 36
C                                                                         SSHOT 37
      WP(NSTG) = 0.0                                                      SSHOT 38
      DN1 =EISP(NSTG) * G                                                 SSHOT 39
      IF ( NLEG .EQ.  1  )   GO TO 20                                     SSHOT 40
C                                                                         SSHOT 41
C                   IF MORE THAN ONE LEG COMPUTE N WTS                    SSHOT 42
C                                                                         SSHOT 43
      N = NLEG                                                            SSHOT 44
      DO  10     I = 2,NLEG                                               SSHOT 45
            EX1 = DVLEG(N) / DN1                                          SSHOT 46
            MR  = EXP(EX1)                                                SSHOT 47
            WPI = (WS(NSTG) + WP(NSTG) + PLEG(N) ) * (MR -1.0)            SSHOT 48
            WP(NSTG) =  WP(NSTG) + WPI                                    SSHOT 49
   10 N = N - 1                                                           SSHOT 50
C                                                                         SSHOT 51
C                   COMPUTE RATIO AND TEST IF OK                          SSHOT 52
C                                                                         SSHOT 53
      FEAS(1) =  WP(NSTG) / WPA(NSTG)                                     SSHOT 54
      FEAS(2) = 0.8                                                       SSHOT 55
      IF ( FEAS(1) .GT. 1.0 )  RETURN                                     SSHOT 56
C                                                                         SSHOT 57
C                   MISSION FEASABLE - CONTINUE                           SSHOT 58
C                   PICK UP SINGLE LEG COMPUTATION                        SSHOT 59
```

```
C                                                                       SSHOT  60
   20 MR = (WS(NSTG) + WPA(NSTG) + PLEG(1) ) /                          SSHOT  61
     X      (WS(NSTG) + WP(NSTG ) + PLEG(1) )                          SSHOT  62
      T1 = ALOG( MR )                                                   SSHOT  63
C                                                                       SSHOT  64
CCC                  NOW FORM DELTA V FOR UPPER STAGE AND               SSHOT  65
CCC                  SEE IF ITS SUFFICIENT                              SSHOT  66
C                                                                       SSHOT  67
      DLTVU = DN1 * T1                                                  SSHOT  68
      IF  ( DLTVU .LT. DVLEG(1)  )   GO TO 30                           SSHOT  69
C                                                                       SSHOT  70
CCC                  ITS SUFFICIENT - SET FLAG AND RETURN               SSHOT  71
C                                                                       SSHOT  72
      FEAS(1) = .5                                                      SSHOT  73
      RETURN                                                            SSHOT  74
C                                                                       SSHOT  75
CCC                  NO IT NEEDS MORE                                   SSHOT  76
C                                                                       SSHOT  77
   30 WP(NSTG) = WPA(NSTG)                                              SSHOT  78
      DLTVL =  DVLEG(1) - DLTVU                                         SSHOT  79
      DLTVLU =  0.0                                                     SSHOT  80
      WG2   = PLEG(1)                                                   SSHOT  81
C                                                                       SSHOT  82
CCC                  TEST THE NUMBER OF STAGES -                        SSHOT  83
C                                                                       SSHOT  84
      IF ( NSTG .EQ. 2 )   GO TO 60                                     SSHOT  85
C                                                                       SSHOT  86
CCC                  ITS A THREE STAGE VEHICLE - SEE IF THE             SSHOT  87
CCC                       SECOND STAGE IS EXPENDABLE                    SSHOT  88
      WP(2) = 0.0                                                       SSHOT  89
      IF  ( REUSE(2) .EQ. 0. )   GO TO 40                               SSHOT  90
      EXP2 =  DLTVL*REUSE(2) /(G*EISP(2))                               SSHOT  91
      MR =  EXP (EXP2)                                                  SSHOT  92
      WP(2) = WS(2) * ( MR - 1.0 )                                      SSHOT  93
C                                                                       SSHOT  94
CCC                  TEST IF THERE IS ENOJGH PROPELLENT                 SSHOT  95
C                                                                       SSHOT  96
      IF  ( WP(2) .LT. WPA(2) )  GO TO 40                               SSHOT  97
C                                                                       SSHOT  98
CCC                  NO - SECOND STAGE CANNOT EVEN RETURN - ABORT       SSHOT  99
C                                                                       SSHOT100
      FEAS(1) = 1.5                                                     SSHOT101
      RETURN                                                            SSHOT102
C                                                                       SSHOT103
CCC                  ITS OK - CONTINUE                                  SSHOT104
C                                                                       SSHOT105
   40 WG2 = PLEG(1) + WP(3) + WS(3)                                     SSHOT106
      MR =  (WS(2) + WPA(2) + WG2 ) /  (WS(2) + WP(2) + WG2 )           SSHOT107
      DLTVLU = G* EISP(2) * ALOG(MR)                                    SSHOT108
C                                                                       SSHOT109
```

```
C                    TEST IF SECOND STAGE CAN DO THE MISSION                    SSHOT110
C                                                                               SSHOT111
      IF ( DLTVLU .LT. DLTVL )   GO TO  50                                      SSHOT112
      FEAS(1) = .7                                                              SSHOT113
      RETURN                                                                    SSHOT114
C                                                                               SSHOT115
C                    NO CONTINUE                                                SSHOT116
C                                                                               SSHOT117
   50 WP(2) = WPA(2)                                                            SSHOT118
C                                                                               SSHOT119
C                              ONLY TWO STAGE RETURN                            SSHOT120
C                                                                               SSHOT121
   60 DLTVLL = DLTVL - DLTVLU                                                   SSHOT122
      T2 = G * EISP(1)                                                          SSHOT123
C                                                                               SSHOT124
C                    SET UP AND TEST IF THE STAGE IS REUSABLE                   SSHOT125
C                                                                               SSHOT126
      WP(1) = 0.0                                                               SSHOT127
      IF ( REUSE(1) .EQ. 0.)  GO TO 70                                          SSHOT128
C                                                                               SSHOT129
C                    NO COMPUTE THE WP                                          SSHOT130
C                                                                               SSHOT131
      EXP3 =  DLTVLL*REUSE(1)  /T2                                              SSHOT132
      MR = EXP (EXP3)                                                           SSHOT133
      WP(1) = WS(1) * (MR -1.0)                                                 SSHOT134
C                                                                               SSHOT135
C                    TEST IF FIRST STAGE CAN RETURN                             SSHOT136
C                                                                               SSHOT137
      IF ( WP(1) .LT.  WPA(1) )  GO TO 70                                       SSHOT138
      FEAS(1) = 1.3                                                             SSHOT139
      RETURN                                                                    SSHOT140
C                                                                               SSHOT141
C                    ITS OK - CONTINUE                                          SSHOT142
C                                                                               SSHOT143
   70 EXP4 = DLTVLL/T2                                                          SSHOT144
      MR = EXP (EXP4)                                                           SSHOT145
      WG2 = WG2 + WP(2) + WS(2)                                                 SSHOT146
      WPI = (WS(1) + WP(1) + WG2) *(MR-1.0)                                     SSHOT147
      WP1    = WP(1) + WPI                                                      SSHOT148
      FEAS(1) = WP1   / WPA(1)                                                  SSHOT149
      WG = WG2 + WS(1) + WP1                                                    SSHOT150
      FEAS(2) = WG / WGA                                                        SSHOT151
      RETURN                                                                    SSHOT152
      END                                                                       SSHOT153
      SUBROUTINE TRNKC(DVLEGX,PLEG)                                             TRNKC   2
C                                                                               TRNKC   3
C          GENERAL INPUT                                                        TRNKC   4
C               WS     THE STRUCTURE WEIGHT FOR THE STAGES                     TRNKC   5
C               WPA    THE ALLOWABLE PROPELLENT WEIGHT FOR THE STAGES          TRNKC   6
C               EISP   EFFECTIVE ISP  (SEC)                                    TRNKC   7
```

```
C                 G      GRAVITY  (CONSTANT)                                  TRNKC   8
C                 WGA    ALLOWABLE GROSS WEIGHT                              TRNKC   9
C                 NSTG   NUMBER OF STAGES                                    TRNKC  10
C                 REUSE  REUSABLE FLAG  0 = EXPENDABLE , 1 = REUSABLE        TRNKC  11
C                                                                            TRNKC  12
C           SPECIFIC INPUT                                                   TRNKC  13
C                 DVLEG(1)  DELTA V FOR LOW ALTITUDE BURN                    TRNKC  14
C                 DVLEG(2)  DELTA V FOR HIGH ALTITUDE BURN                   TRNKC  15
C                 NLEG      SET EQUAL TO 2                                   TRNKC  16
C                                                                            TRNKC  17
C           OUTPUT                                                           TRNKC  18
C                 FEAS(1)     PROPELLENT WEIGHT RATIO                        TRNKC  19
C                 FEAS(2)     GROSS WEIGHT RATIO                             TRNKC  20
C                                                                            TRNKC  21
C                 IF  LESS THAN OR EQUAL TO 1 CONSTRAINTS NOT EXCEEDED       TRNKC  22
C                 IF  GREATER THEN 1 CONSTRAINTS EXCEEDED                    TRNKC  23
C                                                                            TRNKC  24
      DIMENSION PLEG(1),WP(3)                                                TRNKC  25
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),WS(3),WPA(3),EISP(3)                   TRNKC  26
     X             ,REUSE(3),WGA,TR                                          TRNKC  27
     X,FEAS(2)                                                              TRNKC  28
      COMMON/MISC/G                                                          TRNKC  29
      COMMON/DELTAV/DVLEG(2)                                                 TRNKC  30
C                                                                            TRNKC  31
C                 INITILIZE AND COMPUTE STAGE WT                             TRNKC  32
C                                                                            TRNKC  33
      REAL        MRK2,MRKMX,MRCK,MRAB,MR1                                   TRNKC  34
      FEAS(1) = 0.5                                                          TRNKC  35
      FEAS(2) = 0.5                                                          TRNKC  36
      IF ( NSTG .EQ. 2 )  GO TO 10                                           TRNKC  37
      WPL2 = PLEG(1)                                                         TRNKC  38
      DVK2 = DVLEG(2)                                                        TRNKC  39
      EXP1 =  DVK2 / (G * EISP(3) )                                          TRNKC  40
      MRK2 =  EXP (EXP1)                                                     TRNKC  41
C                                                                            TRNKC  42
C                 SECOND KICK MUST DO ALL OF SECOND BURN                     TRNKC  43
C                 NOW GET FUEL REQUIRED FOR SECOND KICK                      TRNKC  44
C                                                                            TRNKC  45
      WP(3) = (WS(3) + WPL2) * (MRK2 -1.0)                                   TRNKC  46
      FEAS(1) =  WP(3) / WPA(3)                                              TRNKC  47
C                                                                            TRNKC  48
C                 IF THE SECOND KICK CANNOT DO THE SECOND BURN, EXIT.        TRNKC  49
C                                                                            TRNKC  50
      IF  ( FEAS(1) .GT.  1.0)  RETURN                                       TRNKC  51
C                                                                            TRNKC  52
C                 SECOND STAGE ASSUMED FULL - EXCESS FUEL IS                 TRNKC  53
C                 USED UP BY YAW STEERING                                    TRNKC  54
C                                                                            TRNKC  55
      WGK2 = WS(3) + WPL2 + WPA(3)                                           TRNKC  56
C                                                                            TRNKC  57
```

```
C                    NOW SET UP PARAMETERS FOR FIRST KICK                    TRNKC  58
C                                                                            TRNKC  59
      DVC =  DVLEG(1)*.75                                                    TRNKC  60
      DVAB = DVLEG(1) -DVC                                                   TRNKC  61
      WPL = WGK2                                                             TRNKC  62
      GO TO 20                                                               TRNKC  63
C                                                                            TRNKC  64
C                    MUST SET UP PARAMETERS IF THERE WAS NO SECOND KICK      TRNKC  65
C                                                                            TRNKC  66
   10 DVC = DVLEG (2)                                                        TRNKC  67
      DVAB = DVLEG(1)                                                        TRNKC  68
      WPL = PLEG(1)                                                          TRNKC  69
C                                                                            TRNKC  70
C                    NOW CONTINUE THE PROCESS                                TRNKC  71
C                                                                            TRNKC  72
   20 MRKMX =  1.0 + ( WPA(2) / (WPL + WS(2) ) )                             TRNKC  73
      EXP2 =  DVC / (G* EISP(2))                                             TRNKC  74
      MRCK =  EXP (EXP2)                                                     TRNKC  75
      EXP3 =  DVAB/ (G* EISP(1))                                             TRNKC  76
      MRAB = EXP (EXP3)                                                      TRNKC  77
C                                                                            TRNKC  78
C                    CHECK IF KICK HAS MORE FUEL THEN REQUIRED               TRNKC  79
C                                                                            TRNKC  80
      REXP  =REUSE(1) +1.0                                                   TRNKC  81
      IF ( MRKMX .GT. MRCK )   MRKMX = MRCK                                  TRNKC  82
      ALFINV =  EISP(2) / EISP(1)                                            TRNKC  83
      MR1 = MRAB * MRCK ** ALFINV / MRKMX**ALFINV                            TRNKC  84
      WP(2)=(WS(2) + WPL ) * (MRKMX -1.0 )                                   TRNKC  85
      WP(1) = ((MR1-1.)*(WPA(2)+WS(2)+WPL)) + ((MR1**REXP-1.)                TRNKC  86
     *      * WS(1) )                                                        TRNKC  87
      WG =  WPL + WPA(2) +WS(2) + WP(1) + WS(1)                              TRNKC  88
      FEAS(1) =  WP(1) / WPA(1)                                              TRNKC  89
      FEAS(2) =  WG / WGA                                                    TRNKC  90
      RETURN                                                                 TRNKC  91
      END                                                                    TRNKC  92
      SUBROUTINE LOSEP(A,B,C,D,H,I,F,G)                                      LOSEP   2
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                            LOSEP   3
      REAL I                                                                 LOSEP   4
      MS = A                                                                 LOSEP   5
      E  = B                                                                 LOSEP   6
      P = C                                                                  LOSEP   7
      SISP = D                                                               LOSEP   8
      SR = H                                                                 LOSEP   9
      SEPK = 1. - I                                                          LOSEP  10
      TSEP = F                                                               LOSEP  11
      MPT = G                                                                LOSEP  12
      RETURN                                                                 LOSEP  13
      END                                                                    LOSEP  14
      SUBROUTINE SEPSV(N,PER,VS,DT,PAY)                                      SEPSV   2
      DIMENSION DT(10),PAY(10)                                               SEPSV   3
```

```
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV                SEPSV   4
      REAL MPLS                                                          SEPSV   5
      NSERV = N                                                          SEPSV   6
      DO 5 I = 1,NSERV                                                   SEPSV   7
      DTHETA(I) = DT(I)                                                  SEPSV   8
    5 MPLS(I) = PAY(I)                                                   SEPSV   9
      RETURN                                                             SEPSV  10
      END                                                                SEPSV  11
      SUBROUTINE TWOBR(DV,DV1)                                           TWOBR   2
C                                                                        TWOBR   3
C   TRANSFER ON TWO DV S RATHER THAN ONE                                 TWOBR   4
C                                                                        TWOBR   5
      COMMON/DELTAV/DVLEG(2)                                             TWOBR   6
      DVLEG(1) = DV1*1.01                                                TWOBR   7
      DVLEG(2) = (DV - DV1)*1.01                                         TWOBR   8
      RETURN                                                             TWOBR   9
      END                                                                TWOBR  10
      SUBROUTINE  SEPX  (MPLA,MPLB,ERELG,NEXIT )                         SEPX    2
C                                                                        SEPX    3
C                 SEPX  THE SEP EXECTIVE ROUTINE IT PERFORMS THE         SEPX    4
C                       LOGIC OF UTILIZING OF THE SEPS VEHICLE           SEPX    5
C                                                                        SEPX    6
C                 SPECFIC INPUT                                          SEPX    7
C                       MPLA  PAYLOAD TO BE DEPLOYED                     SEPX    8
C                       MPLB  PAYLOAD TO BE RETRIEVED                    SEPX    9
C                       ERELG  0 = DO NOT ERASE PREVIOUS MANEUVER        SEPX   10
C                              1 = ERASE THE PREVIOUS MANEUVER           SEPX   11
C                       NEXIT SET TO 0 ON DATA CARD OF DRIVER            SEPX   12
C                                                                        SEPX   13
C                 COMMON INPUT (SEPVEH)                                  SEPX   14
C                       MS                                               SEPX   15
C                       MPT    AMOUNT OF FUEL REMAINING                  SEPX   16
C                       TLEFT AMOUNT OF TIME REMAINING                   SEPX   17
C                       E                                                SEPX   18
C                       P                                                SEPX   19
C                       SISP  SPECIFIC IMPULSE SEPS                      SEPX   20
C                       MDT                                              SEPX   21
C                       RIGAP                                            SEPX   22
C                       TSEP                                             SEPX   23
C                       RSEP                                             SEPX   24
C                       SG     GRAVITY CONSTANT                          SEPX   25
C                                                                        SEPX   26
C                 OUTPUT                                                 SEPX   27
C                       NEXIT TYPE OF EXIT FROM SEPIM SUBROUTINE         SEPX   28
C                       NTUGS NUMBER OF TUG FLIGHTS REQUIRED TO DO THE   SEPX   29
C                             MISSION AND RETURN THE EXPENDED SEPS,IF    SEPX   30
C                             ANY.  NTUGS WILL BE BETWEEN 1 AND 3.       SEPX   31
C                       TLEFT TIME AND FUEL REMAINING ON SEPS VEHEHICLE   SEPX   32
C                       MPT           IN ORBIT                           SEPX   33
C                                                                        SEPX   34
```

```
      COMMON/TUGVEH/TYPE,NSTG,SPAR(3),XS(3),WPA(3),EISP(3)                SEPX   35
     X                ,REUSE(3),WGA,TR                                   SEPX   36
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                        SEPX   37
      COMMON/OUTP/  TD,TU,HCO,ICOS,MPT,TLEFT,MDT,NTUGS                   SEPX   38
      COMMON/C2/TS                                                       SEPX   39
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV                SEPX   40
      REAL MPLS                                                          SEPX   41
      REAL MDT,MPT,MPTSV,MPLB,MPLA,MPTSV1                                SEPX   42
      INTEGER      ERFLG                                                 SEPX   43
      DATA REUSE(1)/1./                                                  SEPX   44
      DATA NFL/0/                                                        SEPX   45
C                                                                        SEPX   46
CCC                 SEARCH INPUT PARAMETERS FOR MISTAKES                 SEPX   47
C                                                                        SEPX   48
      IERR=0                                                             SEPX   49
      IF ( REUSE(1) .LT. 0.5 .AND. SEPK .GT. 0.5 ) GO TO 160             SEPX   50
      IF ( REUSE(1) .LT. 0.5 .AND. MPLB .GT. 0.0 ) IERR=15               SEPX   51
      IF ( SEPK .LT. 0.5 .AND. MPLB .GT. 0.0 ) IERR=16                   SEPX   52
      IF ( IERR .EQ. 0 ) GO TO 5                                         SEPX   53
      MPLB=0.0                                                           SEPX   54
    5 CONTINUE                                                           SEPX   55
      RTINC=500.0                                                        SEPX   56
      HCO=160.                                                           SEPX   57
      TU=0.0                                                             SEPX   58
      TS=0.0                                                             SEPX   59
      TD=0.0                                                             SEPX   60
      ICOS=28.5                                                          SEPX   61
C                                                                        SEPX   62
      IF (NEXIT .GE. 1 )  GO TO 10                                       SEPX   63
C                                                                        SEPX   64
CCC                 INITIALIZATION CALCULATATIONS                        SEPX   65
C                                                                        SEPX   66
      IF(NFL.NE.0) GO TO 10                                              SEPX   67
      NFL = 0                                                            SEPX   68
      C = SISP * 9.80621                                                 SEPX   69
      MDT = ( E* P * 4.409246 ) / ( C * C )                              SEPX   70
      TLEFT = TSEP                                                       SEPX   71
      MPT =  86400.0 * MDT * TLEFT                                       SEPX   72
      TSAVE = TLEFT                                                      SEPX   73
      MPTSV = MPT                                                        SEPX   74
      RTCAP = 0.0                                                        SEPX   75
      RTCPSV = 0.0                                                       SEPX   76
CCC                 INITIALIZATION COMPLETE                              SEPX   77
C                                                                        SEPX   78
C                                                                        SEPX   79
   10 NTUGS = 1                                                          SEPX   80
      SAVET = TLEFT                                                      SEPX   81
      SAVEM = MPT                                                        SEPX   82
      SAVER = RTCAP                                                      SEPX   83
      IF ( ERFLG .GE. 1 )  GO TO 20                                      SEPX   84
```

```
                    NO - SAVE PRESENT CONDITIONS                        SEPX   85
C                                                                       SEPX   86
      MPTSV = MPT                                                       SEPX   87
      TSAVE = TLEFT                                                     SEPX   88
      RTCPSV = RTCAP                                                    SEPX   89
      GO TO 30                                                          SEPX   90
C                                                                       SEPX   91
C                   ERASE -                                             SEPX   92
C                                                                       SEPX   93
   20 TLEFT = TSAVE                                                     SEPX   94
      MPT = MPTSV                                                       SEPX   95
      RTCAP = RTCPSV                                                    SEPX   96
C                                                                       SEPX   97
C                   OK - NOW SEE IF PAYLOAD TO BE RETRIEVED, IF         SEPX   98
C                        ANY, CAN BE BROUGHT DOWN BY A NEARLY           SEPX   99
C                        EXPENDED SEPS.                                 SEPX  100
C                                                                       SEPX  101
C                                                                       SEPX  102
   30 IF(MPLB.LT.5.) GO TO 40                                           SEPX  103
      IF ( SEPK .LT. 0.5 ) GO TO 40                                     SEPX  104
      IF  ( MPLB .GT. RTCAP ) GO TO 40                                  SEPX  105
C                                                                       SEPX  106
C                   INITIATE A TUG FLIGHT TO RETRIEVE PAYLOAD           SEPX  107
C                   AND THE NEARLY EXPENDED SEPS.                       SEPX  108
C                                                                       SEPX  109
      MPLB = 0.0                                                        SEPX  110
      RTCAP = 0.0                                                       SEPX  111
      NTUGS = 2                                                         SEPX  112
C                                                                       SEPX  113
C                   NOW TRY TO PERFORM THE REMAINING MISSION            SEPX  114
C                   WITH THE PRESENT SEPS                               SEPX  115
C                                                                       SEPX  116
   40 CALL  SEPIM (MPLA,MPLB,0,NEXIT )                                  SEPX  117
C                                                                       SEPX  118
C                  SEE IF IT CAN BE DONE - 1,2,5,6 OK -  3,4,7  NO      SEPX  119
C                                                                       SEPX  120
      IF  ( NEXIT .LT. 8 )  GO TO 90                                    SEPX  121
C                                                                       SEPX  122
C                  ITS NOT CONCLUSIVE - CONTINUE                        SEPX  123
C                       DETERMINE THE MAX WEIGHT THAT THE               SEPX  124
C                       PRESENT SEPS CAN RETRIEVE AND RETURN ITSELF     SEPX  125
C                       TO EARTH.                                       SEPX  126
C                                                                       SEPX  127
C                                                                       SEPX  128
      MPTSV1=MPTSV                                                      SEPX  129
      TSAV1=TSAVE                                                       SEPX  130
      IF(RTCAP.LE.5.) GO TO 50                                          SEPX  131
      NTUGS = 2                                                         SEPX  132
   50 RTCAP=0.0                                                         SEPX  133
      IF ( SEPK .LT. 0.5 ) GO TO 75                                     SEPX  134
C
```

```
C                  NOW RETAIN THE PRESENT SEPS AS A NEARLY EXPENDED SEPS  SEPX 135
CC                 AND DETERMINE ITS RETRIEVE CAPABILITIES                SEPX 136
C                                                                         SEPX 137
      RTCAP=RTINC+10.0                                                    SEPX 138
   60 CALL SEPIM  (0.0,RTCAP,1,NEXIT )                                    SEPX 139
      IF ( NEXIT .LE. 5  .OR. NEXIT .EQ. 9 ) GO TO 100                    SEPX 140
      IF ( NEXIT .EQ. 7 .OR. NEXIT .EQ.8 ) GO TO 70                       SEPX 141
      MPT = MPTSV1                                                        SEPX 142
      TLEFT = TSAV1                                                       SEPX 143
      RTCAP=RTCAP+RTINC                                                   SEPX 144
      GO TO 60                                                            SEPX 145
C                                                                         SEPX 146
CC                 IT CAN NO LONGER DO IT - INITIATE A NEW SEPS           SEPX 147
C                                                                         SEPX 148
   70 RTCAP=RTCAP-RTINC                                                   SEPX 149
   75 TLEFT=TSEP                                                          SEPX 150
      MPT = TSEP *86400.0 * MDT                                           SEPX 151
C                                                                         SEPX 152
CC                 CHECK ON THE NEARLY EXPENDED ONE                       SEPX 153
C                                                                         SEPX 154
      IF(MPLB.LT.5.) GO TO 80                                             SEPX 155
      IF ( MPLB .GT. RTCAP ) GO TO 80                                     SEPX 156
C                                                                         SEPX 157
CCC                  THE PAYLOAD TO BE RETRIEVED IS WITHIN THE          SEPX 158
CCC                  CAPABILITY OF THE NEARLY EXPENDED SEPS.            SEPX 159
CCC                  INITIATE A TUG FLIGHT TO RETRIEVE PAYLOAD          SEPX 160
CCC                  AND SEPS.                                          SEPX 161
C                                                                         SEPX 162
      MPLB = 0.0                                                          SEPX 163
      RTCAP = 0.0                                                         SEPX 164
      NTUGS = NTUGS + 1                                                   SEPX 165
C                                                                         SEPX 166
CC                 CONTINUE - SEE IF THE PAYLOAD TO BE DEPLOYED CAN       SEPX 167
CC                            BE TAKEN UP ON INITIAL FLIGHT OF NEW SEPS   SEPX 168
C                                                                         SEPX 169
   80 CALL  SEPIM (MPLA,0,0,NEXIT )                                       SEPX 170
      IF ( NEXIT .GT. 4 )  GO TO 110                                      SEPX 171
      IF ( NEXIT .GE. 3 )  GO TO 90                                       SEPX 172
C                                                                         SEPX 173
CC                 CONTINUE - SEE IF THERE IS A PAYLOAD TO BE RETRIEVED.  SEPX 174
C                                                                         SEPX 175
      IF(MPLB.LE.5.) GO TO 90                                             SEPX 176
C                                                                         SEPX 177
CCC                CAN PAYLOAD TO BE RETRIEVED BE TAKEN ON SUBSEQUENT     SEPX 178
CCC                SEPS FLIGHT                                            SEPX 179
C                                                                         SEPX 180
      CALL  SEPIM (0,MPLB,0,NEXIT )                                       SEPX 181
      IF  (NEXIT .LE. 4 .OR.  NEXIT .EQ. 10 )  GO TO 120                  SEPX 182
      IF ( NEXIT .GE. 7 ) GO TO 90                                        SEPX 183
      NTUGS = NTUGS + 1                                                   SEPX 184
```

```
   90 CONTINUE                                                          SEPX  185
      IF(NEXIT.EQ.1) RETURN                                             SEPX  186
      IF(NEXIT.EQ.2) RETURN                                             SEPX  187
      IF(NEXIT.EQ.5) RETURN                                             SEPX  188
      IF(NEXIT.EQ.6) RETURN                                             SEPX  189
      TLEFT = SAVET                                                     SEPX  190
      MPT = SAVEM                                                       SEPX  191
      RTCAP = SAVER                                                     SEPX  192
      RETURN                                                            SEPX  193
C                                                                       SEPX  194
C                 ***   ERROR CONDITIONS   ***                        SEPX  195
C                                                                       SEPX  196
C                                                                       SEPX  197
  100 IERR = 1                                                          SEPX  198
      GO TO 150                                                         SEPX  199
  110 IERR = 2                                                          SEPX  200
      GO TO 150                                                         SEPX  201
  120 IERR = 3                                                          SEPX  202
  150 CONTINUE                                                          SEPX  203
      NTUGS = -IERR                                                     SEPX  204
      RETURN                                                            SEPX  205
  160 IERR=10                                                           SEPX  206
      GO TO 150                                                         SEPX  207
      END                                                               SEPX  208
      SUBROUTINE  FAZS                                                  FAZS    2
C                                                                       FAZS    3
C  PERFORMS SEPS PHASING, ASSUMING CONSTANT SEP THRUSTING.              FAZS    4
C  INPUTS:  NSERV=NUMBER OF SERVICE LEGS.                               FAZS    5
C           OTHETA= ANGULAR TRAVEL (DEG) OF EACH SERVICE LEG.           FAZS    6
C           MPLS= PAYLOAD (LBS) ON EACH SERVICE LEG.                    FAZS    7
C           PSERV,VSERV= PERIOD (SEC) AND VELOCITY (MPS) OF SERVICE ORBIFAZS    8
C  OUTPUTS:  MPT= FEUL REMAINIG AFTER PHASING (LBS).                    FAZS    9
C            TLEFT= TIME REMAINING ON SEPS AFTER PHASING (DAYS).        FAZS   10
C                                                                       FAZS   11
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       FAZS   12
      COMMON/OUTP/  TO,TU,HCO,ICOS,MPT,TLEFT,MDT,NTUGS                 FAZS   13
      COMMON/SERVIS/NSERV,OTHETA(10),MPLS(10),PSERV,VSERV               FAZS   14
      COMMON/C2/TS                                                      FAZS   15
      COMMON/TSA/TPLS(30),TUP,TDOWN                                     FAZS   16
      REAL MS,MPT,MDT,MPLS,MKG                                          FAZS   17
      F=(MDT*9.80621*SISP)/2.204623                                     FAZS   18
      CONSI1=(3.0*F*PSERV)/(4.0*VSERV)                                  FAZS   19
      TS=TLEFT                                                          FAZS   20
      DO 100 N=1,NSERV                                                  FAZS   21
      MKG=(MS+MPT+MPLS(N))/2.204623                                     FAZS   22
      REV=SQRT((MKG*OTHETA(N))/(360.*CONSI1))                           FAZS   23
      TLEFT=TLEFT-((REV*PSERV)/86400.)                                  FAZS   24
      TPLS(N) = TS-TLEFT                                                FAZS   25
      MPT=MPT-(MDT*REV*PSERV)                                           FAZS   26
  100 CONTINUE                                                          FAZS   27
```

```
      TS=TS-TLEFT                                                       FAZS  28
      TDOWN = TS                                                        FAZS  29
      RETURN                                                            FAZS  30
      END                                                               FAZS  31
      SUBROUTINE TPHAS(A,N)                                             TPHAS  2
      COMMON/ISA/TPLS(30),TUP,TDOWN                                     TPHAS  3
      DIMENSION A(1)                                                    TPHAS  4
      A(1) = TUP/360.                                                   TPHAS  5
      A(N) = TDOWN/360.                                                 TPHAS  6
      IF(N.EQ.2) RETURN                                                 TPHAS  7
      NX = N-2                                                          TPHAS  8
      DO 5 I=1,NX                                                       TPHAS  9
    5 A(I+1) = TPLS(I)/360.                                             TPHAS 10
      RETURN                                                            TPHAS 11
      END                                                               TPHAS 12
      SUBROUTINE  SEPIM (MPLA,MPLB,KSEP,NEXIT)                          SEPIM  2
C                                                                       SEPIM  3
C                   SEPIM THIS SUBROUTINE COMPUTES THE PERFORMANCE      SEPIM  4
C                         OF THE SEPS ON A DEPLOY MISSION.              SEPIM  5
C                                                                       SEPIM  6
C                   SPECIFIC INPUT                                      SEPIM  7
C                         MPLA  PAYLOAD TO BE DEPLOYED                  SEPIM  8
C                         MPLB  PAYLOAD TO BE RETRIEVED                 SEPIM  9
C                         KSEP  ERASE FLAG                              SEPIM 10
C                                     0 = DONT ERASE PRIEVIOUS MANEUVER SEPIM 11
C                                     1 = ERASE PRIEVIOUS MANEUVER      SEPIM 12
C                         NEXIT SET TO 0 PRIOR TO ENTRY                 SEPIM 13
C                                                                       SEPIM 14
C                   OUTPUT                                              SEPIM 15
C                         NEXIT TYPE OF EXIT FROM SEPS IF MISSION POSSIBLESEPIM 16
C                         NTUGS NUMBER OF TUG FLIGHTS REQUIRED TO       SEPIM 17
C                               DO THE MISSION AND RETURN EXPENDED SEPS,SEPIM 18
C                               IF ANY.  NTUGS WILL BE BETWEEN 1 AND 3  SEPIM 19
C                         TLEFT  TIME AND FUEL REMAINING ON SEPS IN ORBITSEPIM 20
C                         MPT                                           SEPIM 21
C                                                                       SEPIM 22
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                       SEPIM 23
      COMMON/OUTP/  TD,TU,HCO,ICOS,MPT,TLEFT,MDT,NTUGS                  SEPIM 24
      COMMON/TABLE/TUGDV(20)                                            SEPIM 25
      COMMON/SERVIS/NSERV,DTHETA(10),MPLS(10),PSERV,VSERV               SEPIM 26
      REAL          MPLA,MPLB,MS,MPT,MRTJG,HCO,ICOS                     SEPIM 27
      REAL           MPLS                                               SEPIM 28
      REAL MDT                                                          SEPIM 29
      TU=0.0                                                            SEPIM 30
      TD=0.0                                                            SEPIM 31
      HCO=160.                                                          SEPIM 32
      ICOS=28.5                                                         SEPIM 33
C                                                                       SEPIM 34
C                   FIRST TEST IF THERE IS A SEPS AVAILABLE             SEPIM 35
C                                                                       SEPIM 36
```

```
      IF ( TLEFT .LT. TSEP -.001 )  GO TO 20                            SEPIM 37
C                                                                       SEPIM 38
C                   NO - ITS A NEW SEPS                                 SEPIM 39
      TLEFT = TSEP                                                      SEPIM 40
      WPLA = MPLA + MS + MPT                                            SEPIM 41
      WPLB =  0.0                                                       SEPIM 42
C                                                                       SEPIM 43
CCC                 CALL TUGCP TO DETERMINE TUG CAPABILITY              SEPIM 44
C                                                                       SEPIM 45
      CALL TUGCP  (WPLA,WPLB,MRTUG,DVTUG)                               SEPIM 46
      IF (DVTUG .LT. TUGDV(13) )  GO TO 10                              SEPIM 47
C                                                                       SEPIM 48
CCC                       TUG DELIVERS SEPS AND MPLA TO SYNC EQ*        SEPIM 49
C                                                                       SEPIM 50
      TLEFT = TLEFT -.005                                               SEPIM 51
      TU = 0.0                                                          SEPIM 52
      TD = 0.0                                                          SEPIM 53
      HCO=19323.                                                        SEPIM 54
      ICOS=0.0                                                          SEPIM 55
      IF (NSERV.GT.0) CALL FAZS                                         SEPIM 56
      NEXIT = 2                                                         SEPIM 57
      RETURN                                                            SEPIM 58
C                                                                       SEPIM 59
CCC                       NEXT CHECK IF ITS CAPABLE                     SEPIM 60
C                                                                       SEPIM 61
   10 NEXIT = 3                                                         SEPIM 62
      IF ( DVTUG .LT. TUGDV(1) )  RETURN                                SEPIM 63
C                                                                       SEPIM 64
CCC                 ITS OK - CONTINUE DETERMINE CHANGEOVER ORBIT        SEPIM 65
C                                                                       SEPIM 66
      CALL  INTORB (DVTUG,HCO,ICOS)                                     SEPIM 67
C                                                                       SEPIM 68
CCC                 DETERMINE THE SEPS DELTA V                          SEPIM 69
C                                                                       SEPIM 70
      CALL  SEPDV (HCO,ICOS,DVSEP,MRSEP)                                SEPIM 71
C                                                                       SEPIM 72
CCC                 PERFORM UP LEG AND PHASING                          SEPIM 73
C                                                                       SEPIM 74
      CALL  PLUPD (MPLA,MRSEP,TU )                                      SEPIM 75
      IF (NSERV.GT.0) CALL FAZS                                         SEPIM 76
C                                                                       SEPIM 77
CCC                 SET NEXIT AND TEST IF THERE IS FUEL REMAINING       SEPIM 78
C                                                                       SEPIM 79
      NEXIT = 1                                                         SEPIM 80
      IF ( MPT .GE. 0.0 ) RETURN                                        SEPIM 81
C                                                                       SEPIM 82
CCC                 SEPS CANNOT DELIVER THE PAYLOAD - SET FLAG AND ABORT SEPIM 83
C                                                                       SEPIM 84
      NEXIT = 4                                                         SEPIM 85
      RETURN                                                            SEPIM 86
```

```
C                                                                       SEPIM  87
C                    THIS ENTRY POINT FOR SEPS AVAILABLE                SEPIM  88
C                    IN SYNC EQ. ORBIT                                  SEPIM  89
C                                                                       SEPIM  90
   20 A = KSEP                                                          SEPIM  91
      WPLB = MPL3+A*SEPK*MS                                             SEPIM  92
      WPLA = MPLA                                                       SEPIM  93
C                                                                       SEPIM  94
C                         DETERMINE THE TUG CAPABILITY                  SEPIM  95
C                                                                       SEPIM  96
      CALL  TUGCP (WPLA,WPLB,MRTUG,DVTUG )                              SEPIM  97
      IF ( DVTUG .LT. TUGDV(13) )  GO TO 30                             SEPIM  98
C                                                                       SEPIM  99
C                         NO - TUG ALONE CAN DELIVER AND RETRIEVE       SEPIM100
C                              PAYLOADS TO SYNC EQ ORBIT                SEPIM101
C                                                                       SEPIM102
      TU = 0.0                                                          SEPIM103
      TD = 0.0                                                          SEPIM104
      HCO=19323.                                                        SEPIM105
      ICOS=0.0                                                          SEPIM106
      IF (NSERV.GT.0) CALL FAZS                                         SEPIM107
      NEXIT = 6                                                         SEPIM108
      RETURN                                                            SEPIM109
                                                                        SEPIM110
C                         TUG ALONE CAN NOT DO IT- CHECK IF ALL OK      SEPIM111
C                                                                       SEPIM112
   30 NEXIT = 7                                                         SEPIM113
      IF ( DVTUG .LT. TUGDV(1) ) RETURN                                 SEPIM114
C                                                                       SEPIM115
C                    ITS OK - CONTINUE                                  SEPIM116
C                         DETERMINE CHANGEOVER ORBIT                    SEPIM117
C                                                                       SEPIM118
      CALL  INTORB (DVTUG,HCO,ICOS )                                    SEPIM119
      CALL  SEPDV (HCO,ICOS,DVSEP,MRSEP)                                SEPIM120
      CALL  PLUPD (MPLB,MRSEP,TD )                                      SEPIM121
C                                                                       SEPIM122
C                         SET UP AND CHECK CONSTRAINTS                  SEPIM123
C                                                                       SEPIM124
      NEXIT = 8                                                         SEPIM125
      IF ( MPT .LT. 0.0 )   RETURN                                      SEPIM126
      IF ( KSEP .EQ. 0 )    GO TO 40                                    SEPIM127
C                                                                       SEPIM128
C                         SEPS RETRIEVED ALONG WITH PAYLOAD             SEPIM129
C                                                                       SEPIM130
      TU = 0.0                                                          SEPIM131
      NEXIT = 10                                                        SEPIM132
      RETURN                                                            SEPIM133
C                                                                       SEPIM134
C                         CONTINUE PROCESS                              SEPIM135
C                                                                       SEPIM136
```

```
   40 CALL  PLUPD (MPLA,MRSEP,TU)                                        SEPIM137
      IF (NSERV.GT.0) CALL FAZS                                          SEPIM138
      NEXIT = 9                                                          SEPIM139
      IF ( MPT .LT. 0.0 )  RETURN                                        SEPIM140
C                                                                        SEPIM141
CC                    MISSION COMPLETE                                   SEPIM142
C                                                                        SEPIM143
      NEXIT = 5                                                          SEPIM144
      RETURN                                                             SEPIM145
      END                                                                SEPIM146
      SUBROUTINE  TUGCP (WPLA,WPLB,MRTUG,DVTUG )                         TUGCP  2
C                                                                        TUGCP  3
CC                  TUGCP - CALLS THE APPROPRIATE TUG EQUATIONS.       TUGCP  4
CC                        ( AT PRESENT - ONLY OPTION IS SINGLE         TUGCP  5
CC                          STAGE CRYOGENIC TUG. )                     TUGCP  6
C                                                                        TUGCP  7
      CALL  CRYO1 (WPLA,WPLB,MRTUG,DVTUG)                                TUGCP  8
      RETURN                                                             TUGCP  9
      END                                                                TUGCP 10
      SUBROUTINE  CRYO1 (WPLA,WPLB,MRTUG,DVTUG)                          CRYO1  2
C                                                                        CRYO1  3
CC                  CRYO1- FINDS THE DELTA V CAPABILITY OF A           CRYO1  4
CC                         SINGLE STAGE TUG WITH PAYLOADS WPLA AND WPLB. CRYO1  5
CC                                                                       CRYO1  6
C                                                                        CRYO1  7
      COMMON/TUGVEH/TYPE,NSIG,SPAR(3),WS(3),WPA(3),EISP(3)               CRYO1  8
     X             ,REUSE(3),WGA,TR                                      CRYO1  9
     X,FEAS(2)                                                          CRYO1 10
      COMMON/MISC/G                                                      CRYO1 11
      REAL        MRTUG                                                  CRYO1 12
      WP=WPA(1)                                                          CRYO1 13
      IF ((WS(1)+WPA(1)+WPLA) .GT.WGA) WP=WGA-(WS(1)+WPLA)               CRYO1 14
      MRTUG=(WP+WS(1)+WPLA)/(WS(1)+WPLA)                                 CRYO1 15
      IF (REUSE(1).LT.0.5) GO TO 100                                     CRYO1 16
      BZ=WS(1)+WS(1)+WPLA+WPLB                                           CRYO1 17
      CZ=-WP*(WPLB+WS(1))                                                CRYO1 18
      WP1 = (-BZ+SQRT(BZ*BZ-4.*CZ))/2.                                   CRYO1 19
      MRTUG=(WP1+WPLB+WS(1))/(WPLB+WS(1))                                CRYO1 20
  100 ALMR=ALOG(MRTUG)                                                   CRYO1 21
      DVTUG = G*EISP(1)*ALMR/(TR+1.)                                     CRYO1 22
      RETURN                                                             CRYO1 23
      END                                                                CRYO1 24
      SUBROUTINE  PLUPD (MPLU,MRSEP,T )                                  PLUPD  2
C                                                                        PLUPD  3
CC                  PLUP - CARRIES SEPS PAYLOAD UP                     PLUPD  4
C                                                                        PLUPD  5
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                        PLUPD  6
      COMMON/OUTP/  TD,TU,HCO,ICOS,MPT,TLEFT,MDT,NTUGS                   PLUPD  7
      REAL        MS,MPT,MDT,MPLU,MRSEP,MPT1                             PLUPD  8
      MPT1 = (( MPT + MS + MPLU ) / MRSEP) - (MS+MPLU)                   PLUPD  9
```

```
      T  =  (MPT - MPT1 ) / (86400.0 * MDT)                              PLUPD 10
      TLEFT  = TLEFT - T                                                 PLUPD 11
      MPT = MPT1                                                         PLUPD 12
      RETURN                                                             PLUPD 13
      END                                                                PLUPD 14
      SUBROUTINE  SEPDV (HCO,ICOS,DVSEP,MRSEP )                         SEPDV  2
C                                                                        SEPDV  3
C           SEPDV - CALCULATES THE REQUIRED SEP DELTA VELOCITY           SEPDV  4
C                   NEEDED FOR SYNC EQ. AND THE CORRESPONDING            SEPDV  5
C                   MASS RATIO.                                          SEPDV  6
C                                                                        SEPDV  7
C           INPUT                                                        SEPDV  8
C                 HCO   ORBIT ALTITUDE                                   SEPDV  9
C                 ICOS  INCLINATION                                      SEPDV 10
C                                                                        SEPDV 11
C           OUTPUT                                                       SEPDV 12
C                 DVSEP THE SEP DELTA V                                  SEPDV 13
C                 MRSEP THE MASS RATIO                                   SEPDV 14
C                                                                        SEPDV 15
      COMMON/SEPVEH/SEPS,MS,E,P,SISP,SEPK,SR,TSEP                        SEPDV 16
      COMMON/MISC/G                                                      SEPDV 17
      REAL ICOS,MU                                                       SEPDV 18
      REAL MRSEP                                                         SEPDV 19
      DATA HS,MU,RE,DTR/19323.,1.4076538E16,3443.9308,57.295779513/      SEPDV 20
      DATA FTPNM/6076.1155/,PI02/1.5707963267 94/                        SEPDV 21
      RCO = (HCO+RE)*FTPNM                                               SEPDV 22
      RS  = (HS +RE)*FTPNM                                               SEPDV 23
      VCO = SQRT(MU/RCO)                                                 SEPDV 24
      VS  = SQRT(MU/RS)                                                  SEPDV 25
      CICO = COS(PI02*ICOS/DTR)                                          SEPDV 26
      DVSEP = SQRT(VCO**2+VS**2-(VS+VS)*VCO*CICO)                        SEPDV 27
      MRSEP = EXP(DVSEP/(G*SISP))                                        SEPDV 28
      RETURN                                                             SEPDV 29
      END                                                                SEPDV 30
      SUBROUTINE  INTORB (DVTUG,HCO,ICOS )                              INTORB 2
C                                                                        INTORB 3
C           INTORB - AN INTERPOLATION SCHEME TO DETERMINE                INTORB 4
C                    THE OPTIMUM CHANGEOVER ORBIT ALTITUDE               INTORB 5
C                    AND INCLINATION.                                    INTORB 6
C                                                                        INTORB 7
C           INPUT                                                        INTORB 8
C                 DVTUG - TUG DELTA V                                    INTORB 9
C                                                                        INTORB10
C           OUTPUT                                                       INTORB11
C                 HCO   ALTITUDE OF CHANGEOVER ORBIT                     INTORB12
C                 ICOS  INCLINATION OF CHANGEOVER ORBIT                  INTORB13
C                                                                        INTORB14
C                                                                        INTORB15
      COMMON/TABLE/TUGDV(20)                                             INTORB16
      REAL ICOS,INC(20),ALT(20)                                          INTORB17
```

```
      DATA        TUGDV/ 10295.74,10600.0,10900.0,11200.0,11500.0,      INTORB18
     X                   11800.0,12100.0,12400.0,12700.0,13000.0,       INTORB19
     X                   13300.0,13600.0,13835.17, 7* 0.0/              INTORB20
      DATA        ALT  / 8000.0,8000.0,8000.0,8000.0,8000.0,8500.0,     INTORB21
     X                   9500.0,10500.0,11500.0,13000.0,14500.0,        INTORB22
     X                   17000.0,18000.0, 7*0.0/                        INTORB23
      DATA        INC  / 28.5,19.6,15.8,12.8,10.14,8.86,8.52,7.67,      INTORB24
     X                   6.4,5.5, 3.87, 2.45, 8* 0.0 /                  INTORB25
C                                                                       INTORB26
C                 FIND THE RANGE OF DELTA V                             INTORB27
C                                                                       INTORB28
      DO 20 NP1 = 2,12                                                  INTORB29
      IF (DVTUG .LE. TUGDV(NP1))GO TO 30                                INTORB30
   20 CONTINUE                                                          INTORB31
C                                                                       INTORB32
C                 FOUND THE RANGE COMPUTE THE ALT AND INC.              INTORB33
C                                                                       INTORB34
   30 NPO = NP1 - 1                                                     INTORB35
      FRAC = ( DVTUG - TUGDV(NPO ) ) / (TUGDV(NP1) - TUGDV(NPO ))       INTORB36
      HCO  =  ALT(NPO )  + FRAC* (ALT(NP1) - ALT(NPO ))                 INTORB37
      ICOS =  INC(NPO )  + FRAC* (INC(NP1) - INC(NPO ))                 INTORB38
      RETURN                                                            INTORB39
      END                                                               INTORB40
      SUBROUTINE CON(I,K)                                               CON     2
      K=0                                                               CON     3
      IF(I.EQ.1H ) RETURN                                               CON     4
      K=100                                                             CON     5
      IF(I.EQ.4H    0) K=0                                              CON     6
      IF(I.EQ.4H    1) K=1                                              CON     7
      IF(I.EQ.4H    2) K=2                                              CON     8
      IF(I.EQ.4H    3) K=3                                              CON     9
      IF(I.EQ.4H    4) K=4                                              CON    10
      IF(I.EQ.4H    5) K=5                                              CON    11
      IF(I.EQ.4H    6) K=6                                              CON    12
      IF(I.EQ.4H    7) K=7                                              CON    13
      IF(I.EQ.4H    8) K=8                                              CON    14
      IF(I.EQ.4H    9) K=9                                              CON    15
      IF(I.EQ.4H   10) K=10                                             CON    16
      IF(I.EQ.4H   11) K=11                                             CON    17
      IF(I.EQ.4H   12) K=12                                             CON    18
      IF(I.EQ.4H   13) K=13                                             CON    19
      IF(I.EQ.4H   14) K=14                                             CON    20
      IF(I.EQ.4H   15) K=15                                             CON    21
      RETURN                                                            CON    22
      END                                                               CON    23
```

AEROSPACE CORPORATION

INTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE

OPERATIONS ANALYSIS (STUDY 2.1)
Program Listing for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-74(7341)-7	30 Sept. 1974	Unclassified

(NOTE: FOR OFF-SITE PERSONNEL, SHOW LOCATION SYMBOL, e.g. JOHN Q. PUBLIC/VAFB)

T. J. Lang
Wolfe, R.R.
Wray, S.T. (3)
STSCF

APPROVED BY Robert R Wolfe DATE 9/30/74

SHEET 1 OF 1

AEROSPACE FORM 2394 REV 7-66

THE AEROSPACE CORPORATION

EXTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE

OPERATIONS ANALYSIS (STUDY 2.1)
Program Listing for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-74(7341)-7	30 Sept. 1974	Unclassified

MILITARY AND GOVERNMENT OFFICES	ASSOCIATE CONTRACTORS AND OTHERS

(NOTE: SHOW FULL MAILING ADDRESS; INCLUDE ZIP CODE, MILITARY OFFICE SYMBOL, AND "ATTENTION" LINE.)

NASA Scientific and Technical
Information Facility
P.O. Box 33
College Park, Maryland 20740 (3)*

NASA Headquarters
Washington, D.C. 20546
Attn: New Technology Representative
Code KT

NASA Headquarters
Washington, D.C. 20546
Attn: Dr. Jack Wild, Code MTE

NASA Headquarters
Washington, D.C. 20546
Attn: Mr. R. Carley, Code MTE (2)

NASA Headquarters
Washington, D.C. 20546
Attn: Mr. V.N. Huff, Code MTE (50)

*Includes 1 reproducible copy

AFR 80-45 DISTRIBUTION STATEMENT X'D BELOW APPLIES

☐ NO DISTRIBUTION STATEMENT
(Classified documents only)

☐ A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

☐ B. DISTRIBUTION LIMITED TO U.S. GOV'T AGENCIES ONLY;

_____ (Reason)

_____ (Date statement applied). OTHER REQUESTS FOR THIS DOCUMENT MUST BE REFERRED TO _____ (Controlling DOD office)

APPROVED BY Robert R Wolfe (FOR THE AEROSPACE CORPORATION) DATE 30 Sept 74

APPROVED BY _____ (FOR COGNIZANT AF OFFICE) _____ (SYMBOL) DATE _____

IF LIST COMPRISES TWO OR MORE SHEETS, COMPLETE THIS SIGNATURE BLOCK ON LAST SHEET ONLY

SHEET 1 OF 1

AEROSPACE FORM 2380 REV 7-71