2(mix)

ASD-ASTN-1533

ASD-ASTN-1533
CR-123735

Summary Report

SHUTTLE OPERATIONS SIMULATION MODEL PROGRAMMERS'/USERS' MANUAL

(NASA-CR-123735) SHUTTLE OPERATIONS SIMULATION MODEL PROGRAMMERS'/USERS' MANUAL Summary Report D.G. Porter (Teledyne Brown Engineering) Jun. 1972 132 p CSCL 09B
N72-28200
Unclas
G3/08 34736

June 1972

TELEDYNE
BROWN ENGINEERING

Research Park • Huntsville, Alabama 35807

132

SUMMARY REPORT
ASD-ASTN-1533

SHUTTLE OPERATIONS SIMULATION MODEL PROGRAMMERS'/USERS' MANUAL

By

D. G. Porter

June 1972

Prepared For

SYSTEMS OPERATIONS BRANCH
MECHANICAL AND CREW SYSTEMS INTEGRATION DIVISION
ASTRONAUTICS LABORATORY
GEORGE C. MARSHALL SPACE FLIGHT CENTER

Contract No. NAS8-21804

Prepared By

SYSTEMS DESIGN AND SIMULATION BRANCH
SYSTEMS ENGINEERING DEPARTMENT
AEROSPACE SUPPORT DIVISION
TELEDYNE BROWN ENGINEERING COMPANY
HUNTSVILLE, ALABAMA

ABSTRACT

The prospective user of the Shuttle Operations Simulation (SOS) Model is given sufficient information by this document to enable him to perform simulation studies of the Space Shuttle launch-to-launch operations cycle. The procedures used for modifying the SOS Model to meet user requirements are described. The various control card sequences required to execute the SOS Model are given in the text of the report. The report is written for users with varying computer simulation experience.

A description of the components of the SOS Model is included that presents both an explanation of the logic involved in the simulation of the Shuttle operations cycle and a description of the routines used to support the actual simulation.

APPROVAL:

R. E. Talley, Chief
Operations Analysis Section

T. L. Ryan, Manager
Systems Design and Simulation Branch

B. D. Turney, Manager
Systems Engineering Department

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

DEFINITION OF SYMBOLS

CA	Cards added
CD	Cards deleted
CPU	Central Processing Unit
GPSS	General Purpose Systems Simulator
LUT	Launch Umbilical Tower
MSFC	Marshall Space Flight Center
SOS	Shuttle Operations Simulation

Control Card Sequence--A sequence of UNIVAC 1108 instructions that represents the information required to execute a computer program.

Insert Card--A specific card type used to indicate the addition to or the deletion of the statements contained in the computer program.

Location Number--The UNIVAC 1108 computer assigned number to each statement of the computer program.

Mnemonic Inserts--The GPSS II language commonly uses numeric block identification numbers. The language can accept alphanumerics as block numbers by using specific control card sequences. The alphanumeric block identification numbers are referred to as mnemonic inserts.

Model--A representation of a process or concept that permits the manipulation of its parameters as a means of determining how the concept or process behaves in various situations.

Operational Cycle--Those functions performed on each Shuttle element after its previous launch in preparation for its succeeding launch.

Simulation--Represents the behavior of a physical system by means of a computer and a program. The computer and program serves to demonstrate system behavior under the influence of stimuli and permits system performance analysis.

Subprogram--A routine designed to solve a particular problem that becomes applicable to the problem only when appropriate parametric values are supplied.

Supplemental Fortran Instructions--A collection of Fortran routines, both user written and GPSS II language supplied, that implement and support the simulation of the Shuttle turnaround cycle.

System Model Description--A sequence of GPSS II language instructions that, upon implementation, simulate the Space Shuttle launch-to-launch operations cycle.

Turnaround Cycle--Starting with a Shuttle vehicle at launch, represents those operations performed to prepare the vehicle to the same state of preparedness for its next launch.

SECTION I. INTRODUCTION

A. Purpose

A model has been developed that analyzes the major events of the Space Shuttle launch-to-launch operations cycle. The formulation of a model describing the Shuttle operations cycle was initiated by isolating the operational characteristics of the elements involved in the operations cycle. Next, the logical rules governing the interaction of those operational characteristics were identified and formulated into a model describing the entire Shuttle launch-to-launch operations cycle. To further analyze the system, a simulation technique was employed as an effective method of testing and evaluating the proposed real system. The simulation technique employed allows statistical estimates of the operational parameters in the real system. Using the results of the simulation technique only allows comparison of the alternatives to the real system, and does not necessarily generate an optimal solution. The resulting model has been named the Shuttle Operations Simulation (SOS) Model and, as developed, is activity oriented and designed to simulate the major events of the Space Shuttle launch-to-launch operations cycle. The SOS Model serves as a tool for conducting operations and logistical analysis relevant to the definition and design phase of the Space Shuttle program.

The Space Shuttle operational cycle is described by the General Purpose Systems Simulator (GPSS) II computer language which is supported by special purpose Fortran subprograms. As written, the SOS Model is capable of projecting real life activities and functions which will probably occur during the operational phase of the Space Shuttle program. From the results of parametric studies, real life projection and logistical and operational requirements can be determined for the Space Shuttle system.

This document is written to describe the use and operations of the SOS Model. Sufficient data and instructions are given to the potential user of the SOS Model to enable him to modify the model to meet his particular needs. A complete description of the components of the SOS Model is also given within the text of this document.

B. Scope

This report provides instructions for the use and operation of the SOS Model. Supplemental information is given to allow user modifications to the program enabling the user to meet his particular requirements. Section II of this document contains general information relating to the SOS Model operations and a description of the necessary input data required for implementing the SOS Model. Section III describes the two basic components of the SOS Model, the System Model Description, and the Supplemental Fortran Instructions. Included in Section IV are the control card sequences used to execute the SOS Model along with a description of each control card sequence. The Appendices contain data to support the sections of this document.

SECTION II. SOS MODEL OPERATING INSTRUCTIONS

A. General Information

The SOS Model is designed for execution by the Marshall Space Flight Center (MSFC) UNIVAC 1108 EXEC VIII Computer System and requires approximately 32,800 words of core storage. The SOS Model consists of two major components: the System Model Description and the Supplemental Fortran Instructions. The System Model Description component consists of a sequence of GPSS II language instructions that describe the launch-to-launch operation cycle of the Space Shuttle flight elements and support hardware. The Supplemental Fortran Instructions are a combination of user-written Fortran subprograms and the standard subprograms supplied as part of the GPSS II language. The GPSS II supplied subprograms are responsible for implementing the instructions contained in the System Model Description.

The SOS Model is maintained as a catalogued drum file by the UNIVAC 1108 computer located at MSFC. The catalogued file contains both the Supplemental Fortran Instructions and the System Model Description which defines a baseline configuration of the operational elements and their time distribution parameters. The System Model Description, as catalogued, defines a specific operational concept for the Space Shuttle launch-to-launch operations cycle. Using the basic configuration, a number of Shuttle flight elements and support facility arrangements are capable of being studied. Therefore, the user must decide the type of analysis to be performed and select the parameters to be varied.

B. Temporary System Model Modification Procedure

If the catalogued version of the System Model Description of the SOS Model is not entirely satisfactory to the user, temporary modifications to the System Model Description can be accomplished by utilizing the procedure described below. In making temporary model modifications, it will be useful to refer to Appendix A which contains a printout of the System Model Description. This printout is fully documented with each of the GPSS II language system variables used in the System Model Description defined prior to the first GENERATE block. Section IV shows the appropriate placement of the Model modification cards in the control card sequences required to execute the SOS Model.

Each statement of the System Model Description is computer numbered with the number positioned to the extreme left of each statement. Appendix A contains a computer printout of the System Model Description showing the GPSS II statements, block numbers, and computer assigned location numbers. To temporarily modify the System Model Description, such as to change the Booster and Orbiter storage capacities or to change the number of launch pads, the insert card method is used.

The insert card method requires an insert card which contains any of the following: the location number of the statement to be replaced (Example 1), the sequential location numbers of the statements to be modified (Example 2), or the location number of the statement which modifications are to follow (Example 3). All insert cards are punched beginning with a minus sign in column one. The inserted statements must conform with GPSS II block type formats (Reference 2). The following examples illustrate the use of the insert card method for implementing temporary model modifications. The format for these examples is: original GPSS II statements from the System Model Description, solid line, insert cards and new GPSS II statements. A brief discussion follows each example.

Example 1:

```
000051        2     CAPACITY     2
000052        3     CAPACITY     2
000053        4     CAPACITY     9
----------------------------------
-51,51
 2     CAPACITY     1
-53,53
 4     CAPACITY     7
```

In Example 1 the storage capacities of both Booster maintenance and Booster storage, identified as location Nos. 51 and 53, respectively, were modified. The storage capacity of Booster maintenance was changed from two to one, and the Booster storage capacity was changed from nine to seven. Notice on each insert card that the location number is listed twice. This example illustrates a one for one replacement of the statement whose location was 53. It should be noted that all statements between the insert card and the succeeding insert card, if any, will replace the statement at location 53. Thus, it is possible to delete the statement at location 53 or to insert a finite number of statements in place of the statement at location 53.

Example 2:

```
000000          11    CAPACITY    2
000001          12    CAPACITY    2
000002          13    CAPACITY    9
---------------------------------------
-00,02
 11    CAPACITY    3
 12    CAPACITY    1
 13    CAPACITY    8
```

Example 2 illustrates the modification procedure required when replacing all statements between and including two statement location numbers. The storage capacities of Orbiter maintenance, Orbiter test, and Orbiter storage were changed from two, two, and nine, respectively, to three, one and eight, respectively. The insert card contains the two numbers, inclusive, that identify the program segment to be modified. Using the above example and the discussion at the conclusion of Example 1, it is possible to delete all statements between and including two location numbers or to insert a finite number of statements in place of the statements between and including the indicated location numbers.

Example 3:

```
000080          31    CAPACITY    1
---------------------------------------
-80
 32    CAPACITY    3
 33    CAPACITY    4
 34    CAPACITY    3
```

Example 3 illustrates how a number of statements can be inserted to follow a particular statement. It should be noted that only one location number is required on the insert card with this number preceeded by a minus sign.

Example 4:

```
000469          70    QUEUE          21                    ALL    71      72
```

```
-469,469
 70    QUEUE         21                  ALL     71      73
```

Example 4 changes the System Model Description from the use of two launch pads to the use of three launch pads. By using the following QUEUE card, instead of the QUEUE card shown previously in this example, the System Model Description can be modified to limit the number of launch pads to one.

```
 70    QUEUE         21                          71
```

It should be noted that the maximum number of available launch pads is three.

C. Temporary Supplemental Fortran Instructions Modification Procedure

Temporary modifications to the subprograms contained in the Supplemental Fortran Instructions can be accomplished by utilizing the insert card method outlined in the preceeding discussion. Appendix B contains a printout of the Supplemental Fortran Instructions. Like the System Model Description, each statement of the Supplemental Fortran Instructions has a computer assigned location number. These location numbers are used in conjunction with the insert card method previously outlined to perform the temporary modifications required by the user of the SOS Model.

Mention of the modification procedure to the Supplemental Fortran Instructions has been made only for completeness. Thus, no effort has been made to include examples of modifications. Modifications to the Supplemental Fortran Instructions should be attempted by only the most experienced user. Section IV shows the control card sequence required when modifications are made.

D. Input Data Description

The data input requirements for the execution of the SOS Model consist of the two card types described below.

Card type one, which is commonly referred to as the comment card, is used by the SOS Model to place a desired comment or heading on the output summary tables resulting from the execution of the SOS Model. All 80 columns of the card may be used to obtain the heading for the output summary tables.

Card type two, which is commonly referred to as the data card, defines the traffic density to be scheduled, the number of Booster and Orbiter elements to be used, the number of active Launch Umbilical Towers (LUT's), the level of significance for the statistics collected by the Confidence Interval Option (Reference 3), and the random number generator seeds for both the Traffic Model Simulator and the Shuttle Operations Simulator. The user has the option of choosing any of the 5 available traffic densities, numbered consecutively from 1 to 5 which represent 20, 35, 45, 55, and 75 launches per year, respectively. The required format for the type two data card is:

Column 1	Traffic Density Required
Columns 2 through 5 (right justified)	Number of Boosters
Columns 6 through 10 (right justified)	Number of Orbiters
Columns 11 through 15 (right justified)	Number of LUT's
Columns 16 through 20 (right justified)	Level of Significance for Confidence Interval Option
Columns 61 through 70	Random Number Seed for the Traffic Model Simulator
Columns 71 through 80	Random Number Seed for the Shuttle Operations Simulator

Sample data cards can be seen in Section IV which contains the control card sequences used to execute the SOS Model.

The Fortran format associated with data card type two in SUBROUTINE HELP is:

FORMAT (I1, I4, 3I5, 40X, 2I10) .

Following the second data card type, it is necessary to place a card which terminates the computer's execution of the current set of data. This is accomplished by placing (after the data card) a card that has the word END punched in columns seven through nine.

SECTION III. SOS MODEL DESCRIPTION

A. SOS Model Components

The SOS Model consists of two major components: the System Model Description and the Supplemental Fortran Instructions (Figure 1). The model, formed by analyzing the operational elements of the Space Shuttle turnaround cycle, was translated into the GPSS II language. The resulting set of instructions that describe the model are known as the System Model Description. When the System Model Description is implemented, the launch-to-launch operations cycle of the Space Shuttle vehicle and support hardware is simulated. The Supplemental Fortran Instructions of the SOS Model contain the routines which implement the instructions contained in the System Model Description. The routines contained in the Supplemental Fortran Instructions are mostly user written Fortran routines. However, there are several routines supplied by the GPSS II language that are included in the Supplemental Fortran Instructions.

Reference 1 describes the SOS Model. Sections III B and III C give more detailed data relating to the two major components of the SOS Model.

B. SOS Model System Description

The System Model Description component of the SOS Model is a sequence of GPSS II language instructions that allows the simulation of the operational facts concerning individual Shuttle processing facilities, LUT's, Orbiters, and Boosters. The necessary instructions are included to simulate the operations involved in the complete launch-to-launch cycle of the Space Shuttle. The operations in the launch-to-launch cycle are expected to have time distributions associated with their performance. In order to reflect this, frequency distribution functions have been estimated that approximate the expected real life time distributions for the performance of each operation. The estimated frequency distribution functions are used in conjunction with the Monte Carlo technique to provide a stochastic simulation of the launch-to-launch operations for the Space Shuttle.

The following is a brief description of the overall logic in the System Model Description. Figure 2 is a detailed flowchart depicting

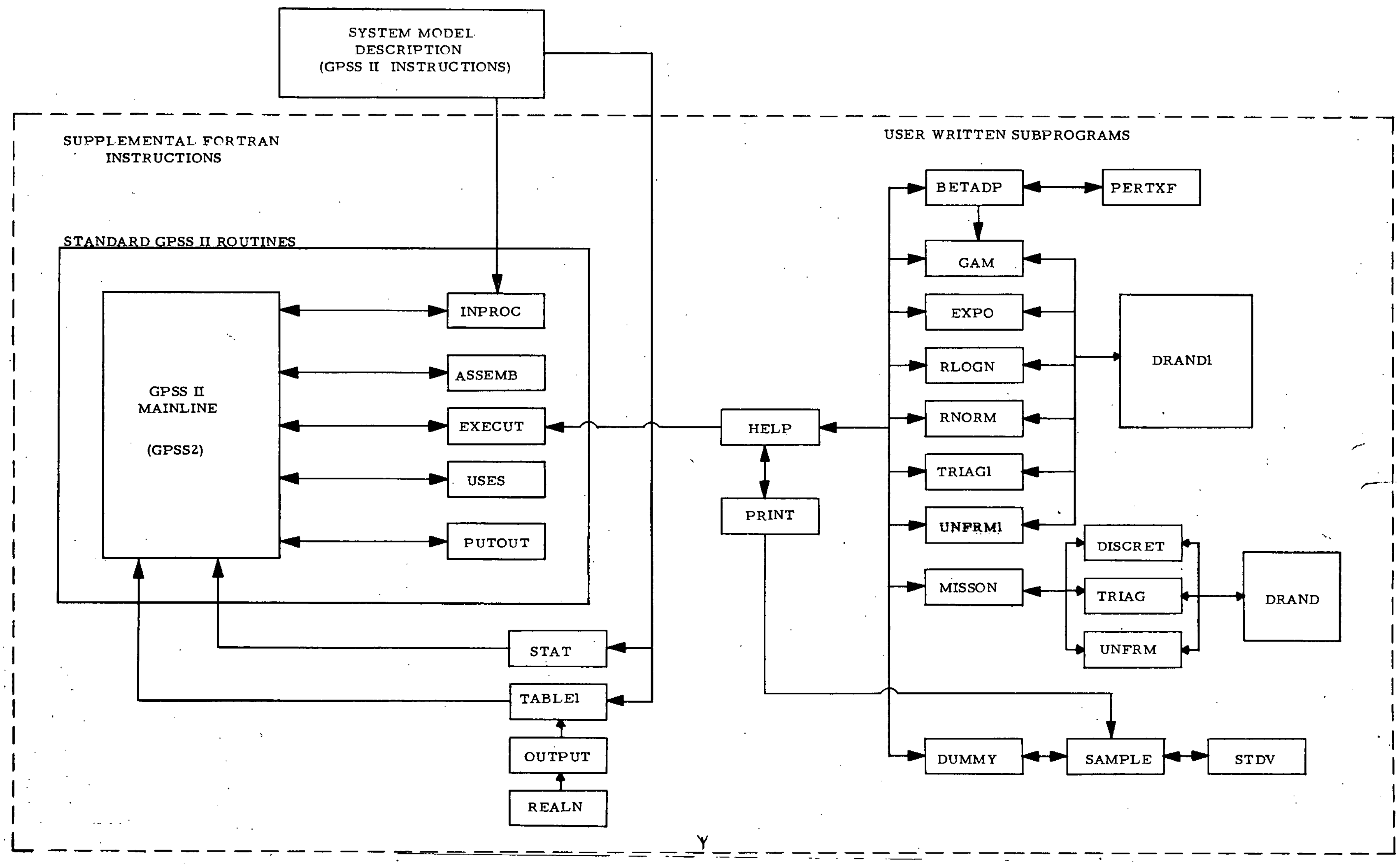

FIGURE 1. SOS MODEL CONFIGURATION

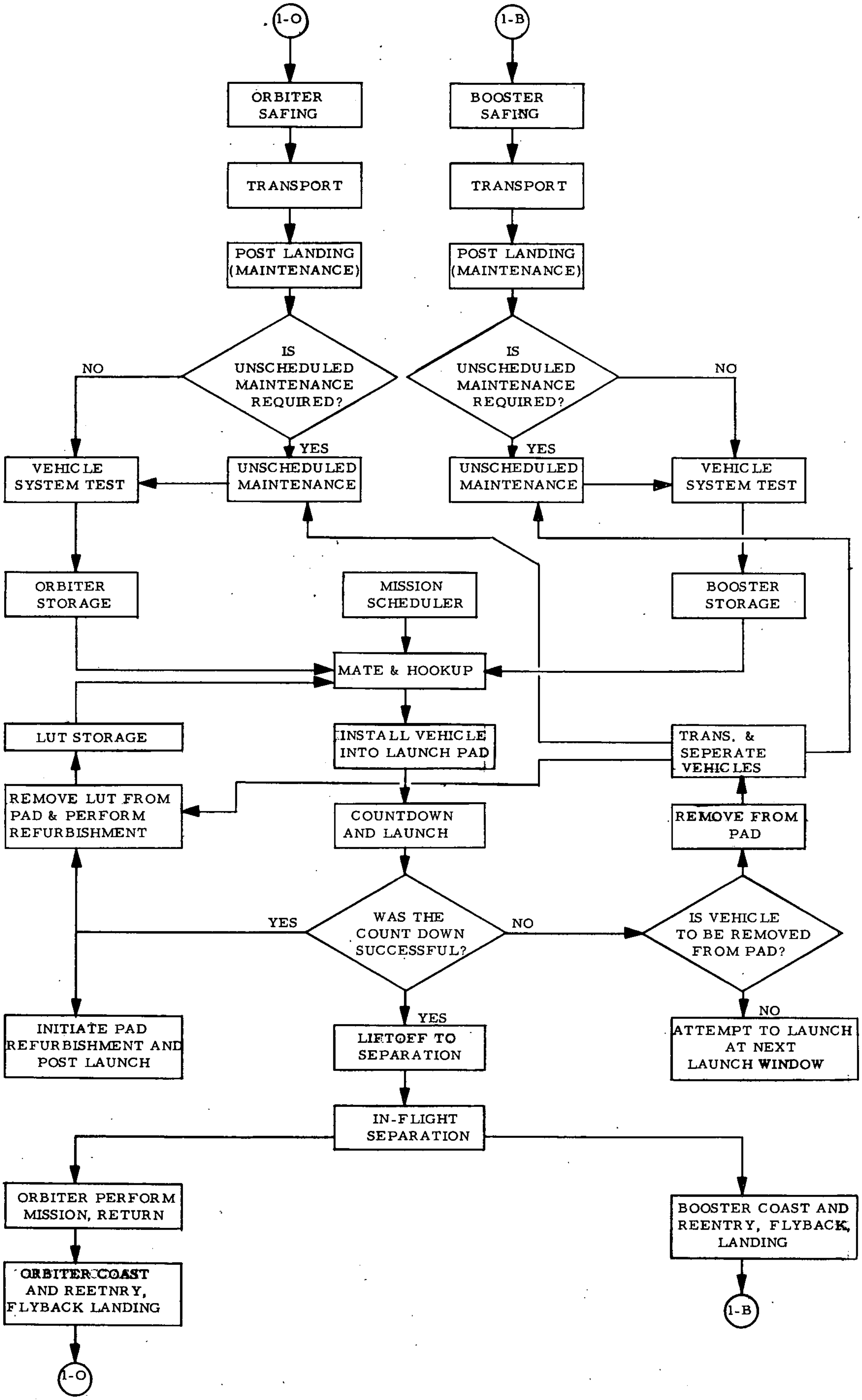

FIGURE 2. SYSTEM MODEL DESCRIPTION FLOWCHART

the functional logic of the Shuttle turnaround cycle. In the ensuing discussion, references are made to GPSS II system variables through the use of parentheses. A brief definition of the commonly used system variables is contained in Table I.

Using a GPSS II HELP block and Fortran SUBROUTINE HELP, a data card is read which contains the number of Boosters, Orbiters, and LUT's. Each of these values is placed in a separate storage location (SAVEX) for use during the simulation.

The simulation proceeds by using a HELP block to determine both a realistic first launch date and the parameters associated with the mission scheduled for that date. The HELP block, through Fortran SUBROUTINE MISSON, also establishes the time between launches. Throughout the simulation, the launch intervals and the parameters for both the current and the succeeding missions are always known.

The earliest feasible storage exit time prior to launch is the maximum processing time required to prepare the vehicle for launch. The request for vehicle elements for the current mission is not made until this time. The ready storage time prior to storage exit is determined by subtracting both the current simulated clock time (C1) and the maximum time required to process the vehicle from mate and hookup initiation to launch (V5) from the launch date of the current mission (X21). At the maximum processing time prior to the scheduled mission, a request is made to determine if there is a Booster, Orbiter, and LUT available in their respective storages. If all three elements are available, their storage departure time is recorded and they are placed in a queue for the mate and hookup operation (Q20).

The time a request may be delayed (V11) because of vehicle element unavailability is the difference between the maximum and minimum time to process the vehicle from mate and hookup initiation to launch. If this time is exceeded, the mission is cancelled and a record of the unavailable elements is kept. When all vehicle elements are available and the maximum time delay has not occurred, the pad availability is checked. The vehicle elements are removed from storage, sent to the mate and hookup waiting line (Q2), and this information is recorded. If a pad will not be available, the mission is cancelled and the pad unavailability is recorded.

When the vehicle elements complete mate and hookup (S20), the vehicle goes to the launch pad waiting line (Q21) where it is

TABLE I. DEFINITION OF COMMONLY USED SYSTEM VARIABLES*

Scope	Name	Meaning
Equipment	Fn Sn	Facility status, facility n Storage occupancy, storage n
Queue	Qn	Queue length, queue n
System	Vn Xn Kn	Defined variable, numbered n Stored value, cell n Constant n

*A more complete description of the available GPSS II system variables can be found in Reference 2.

assigned to the first available pad after the pads have been checked in sequential order. (After a vehicle has been assigned a pad, the pad is unavailable until either the vehicle has been launched and the pad refurbished or until the vehicle is removed from the pad.)

Once a pad has been assigned to a vehicle and a LUT, they are transferred to the pad from the launch pad waiting line (Q21). The vehicle is installed on the pad and precountdown preparations are performed (S22). The countdown time (V6) is compared to the time left to launch (V2) to determine if sufficient time remains to complete countdown. If excess time exists to launch, a built-in variable length hold (V3) is utilized. When insufficient time remains to meet the first launch window, the second launch window is tried.

Should a failure occur while the vehicle is in countdown, a check is made to determine if the failure can be corrected on the pad. If the failure cannot be corrected on the pad, the vehicle and LUT are removed from the pad (S25) and the pad is released for the use of another mission. The Orbiter is separated from the Booster and LUT (S30) and sent to maintenance (S11). The Booster is separated from the LUT (S19) and they are sent to their respective maintenances (S2, S18). If no failure occurs, the vehicle is launched. The LUT is removed from the pad (S17) and sent to maintenance (S18) and the launch pad is refurbished (S27) and released for future launches. The lift-off time, time between lift-off and separation, Booster flight time, and Orbiter flight time are recorded.

After the Booster flys back and lands, it enters the waiting line for Booster safing (Q1) and enters the safing facility (S1) when it becomes available. Upon completion of safing, the Booster enters the maintenance waiting line (Q2) and maintenance is performed. When maintenance is completed, the Booster enters storage and becomes available for future launch requests.

After the Orbiter has performed its mission, flown back to landing site and landed, it enters the waiting line for Orbiter safing (Q10). When the safing facility (S10) becomes available, the Orbiter is safed and transported to the Orbiter maintenance waiting line (Q11). After maintenance is completed on the Orbiter, it is transported to storage for future use.

When the above Shuttle operations cycle logic is simulated, three computer outputs that describe the simulation results are available to the user of the SOS Model. Sample computer outputs resulting from executing the SOS Model can be found in Appendices C, D, and E.

C. SOS Model Supplemental Fortran Instructions

The Supplemental Fortran Instructions component of the SOS Model is composed of user written Fortran routines and the GPSS II supplied routines. The instructions necessary to simulate the System Model Description of the SOS Model are supplied by the Supplemental Fortran Instructions. Figure 1 denotes the manner in which the System Model Description interfaces with the Supplemental Fortran Instructions.

The GPSS2 routine, as supplied by the GPSS II language, is the nucleus of the SOS Model and, as such, actually performs the operations required to execute the System Model Description which in turn simulates launch-to-launch operations cycle of the Space Shuttle. All of the GPSS II supplied routines are in binary format and cannot be changed, with the exception of the GPSS2 routine. Consequently, the GPSS2 routine is the only GPSS II supplied routine discussed within this document. At the discretion of the user, this routine may be altered through the use of Fortran parameter statements to increase or decrease the size limitations of the System Model Description. However, this should be attempted by only the most experienced of users.

Upon request, the GPSS2 routine calls either of two user written Fortran subroutines: TABLE1 and STAT. Both subroutines are used to read the System Model Description of the SOS Model as data. Both subroutines are described below.

TABLE1 is used to output a table of time distribution parameters for each of the operations simulated by the System Model Description. The output of the time distribution parameters is accomplished by reading each statement of the System Model Description until a comment card is located that has the key symbol IDST in columns 68 through 71, with columns 67 and 72 containing a blank. The program continues reading cards until a HELP card is found which is used to supply the desired information for the time distribution parameters. The program prints the title of the activity found on the comment card and the time distribution parameters found on the HELP card. Appendix C contains an example listing of the TABLE1 output. It should be noted that TABLE1 and its support routines of OUTPUT and REALN are not active during the dynamic simulation of the Space Shuttle operations cycle. Section IV gives the control card sequences required to obtain the TABLE1 output.

STAT is used to read and store headings to be in the output of confidence interval statistics. A detailed description of the use of the Confidence Interval Option is contained in Reference 3. Each statement of the System Model Description is read until a statement with the key symbol CONF in columns 67 through 70 is found. At such time, the heading contained on that statement is stored to be used during the output of the confidence interval statistics. Appendix E contains an example output from the Confidence Interval Options. STAT, like TABLE1, is not active during the dynamic simulation of the operations cycle of the Space Shuttle. A complete description of the use of SUBROUTINE STAT and of the Confidence Interval Option can be found in Reference 3.

The primary user written routine of the Supplemental Fortran Instructions is the HELP routine. The HELP routine enables the GPSS II user to perform operations that are not easily obtainable from using the GPSS II block types. Upon completion of the operations in the HELP routine, the calculated values are transmitted back to the System Model through a specified SAVEX (system variable Xn), which is a core storage location. The user employing the GPSS II HELP block must supply a Fortran subroutine called HELP which has five or less fixed point variables as arguments. These arguments are specified on the GPSS II HELP card in the X, Y, Z, Mean, and Mod field locations.

The launch requirements for the Space Shuttle and the Orbiter on-orbit mission requirements are supplied by SUBROUTINE MISSON which is supported by subprograms UNFRM, DICRET, TRIAG, and DRAND. To gain access to SUBROUTINE MISSON, one must use the HELP block in the GPSS II model as illustrated in the following example:

```
* CALL HELP TO GENERATE ATTRIBUTES OF NEXT LAUNCH
 24    HELP          K1     X11    X12    BOTH  10     25     X13    X10
```

Upon entering the HELP block, the program calls for SUBROUTINE HELP with its five fixed point variables. The proper form of SUBROUTINE HELP is shown below:

```
SUBROUTINE HELP(K1,K2,K3,K4,K5)
```

The first variable in the call statement, K1, is used by a Fortran computed GO TO statement within SUBROUTINE HELP to determine which Fortran user written routine is to be used to assist the operations of the GPSS II block types. In the above example, the value of K1 is used to call SUBROUTINE MISSON from SUBROUTINE HELP.

SUBROUTINE MISSON calculates the length of the next Orbiter mission, determines the mission type, computes the time until the next launch, and the time between launch opportunities. These values are calculated as different variables in SUBROUTINE MISSON and reset equal to K2, K3, K4, and K5 by SUBROUTINE HELP. The calculated values are then stored in the SAVEX locations of X11, X12, X13, and X15 for use by the GPSS II program.

SECTION IV. UNIVAC 1108 CONTROL CARD SEQUENCES USED IN THE EXECUTION OF THE SOS MODEL

A. General Discussion

Execution of the SOS Model can be accomplished by choosing the appropriate control card sequence of those described in this section. Multiple executions of the SOS Model can be accomplished by using the Systems Model Description as catalogued on the UNIVAC 1108 drum files. Two executions, both with and without the Confidence Interval Option and each simulating 500 Shuttle launch requests, can be obtained from one computer run with a resulting Central Processing Unit (CPU) time of approximately 3 minutes.

In selecting the appropriate control card sequence, it is assumed that the user has acquainted himself with the material and techniques presented in this document, and that he has defined his objectives sufficiently to enable an intelligent decision in the choice of control card sequences. The control card deck setups which follow are used in conjunction with the execution of the SOS Model. No effort has been made to enumerate all the possible control card sequences. Instead, only the most commonly used sequences are presented. A description of each individual control card is provided in Appendix F.

B. Multiple Executions of the SOS Model

The basic control card sequence used for executing the SOS Model, when no modification or additions are made to either the System Model Description or the Supplemental Fortran Instructions, is shown in Figure 3. The sequence (Figure 3) includes two of the options available to the user of the SOS Model. Both options are for the convenience of the user and the inclusion or deletion of either option has no effect on the results of the simulation.

The @SETC 4 control card placed before the @XQT control card generates the TABLE1 output as described previously in this document. In order to produce the output table, it was necessary to place the System Model Description in the execution run stream. This was accomplished by the addition of the one extra @ADD MODEL control card placed immediately after the first @XQT control card.

```
@RUN,//T SAMPLE,999999,PORTERBIN225,3,150
@ASG,T GPSSE8,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
@ASG,T MODEL,F2
@DATA FILE2,MODEL
@END
@SETC 4
@XQT GPSSE8.MAPGPS
@ADD MODEL.
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
       END
@SETC 3
@XQT GPSSE8.MAPGPS
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
       END
@FIN
@FIN
```

FIGURE 3. MULTIPLE EXECUTION OF THE SOS MODEL WITHOUT MODIFICATIONS

The @SETC 3 control card placed before the second @XQT control card eliminates the printout of the model listing but allows the output of the summary results and the standard GPSS II output.

C. Modifications to the System Model Description

If the catalogued version of the System Model Description is not completely satisfactory to the user of the SOS Model, temporary model changes can be made through the use of the control card sequence shown in Figure 4.

The control card sequence shown in Figure 4 is used for multiple executions of the SOS Model. The sequence also encompasses the use of the TABLE1 output and the use of the @SETC 3 command. If only one execution is desired, the two @FIN cards should follow the first END card.

D. Modifications to the Supplemental Fortran Instructions

The user written routines of the Supplemental Fortran Instructions can be added to or modified in order to meet the user's requirements by utilizing the control card sequence shown in Figure 5. The use of this control card sequence enables the user to test both a modified user written routine and a newly developed user written routine by utilizing the catalogued drum files. The control card sequence (Figure 5) is an example of only one execution of the SOS Model; but, multiple executions are possible by utilizing the information previously supplied in this section. The procedure for modifying the user written routines entails the use of the insert card method described previously in conjunction with modifications to the System Model Description. More detailed information concerning the user written routines can be found in Appendix B.

E. Creation of an Update Tape

The control card sequence (Figure 6) is used to create a tape that includes all updates to both the Supplemental Fortran Instructions and the System Model Description. The created tape, which includes all updates, can then be used to update the catalogued drum file of the SOS Model. Great care should be taken when creating a tape to be used for updating the SOS Model. To ensure the proper operation of the updated tape, Figure 7 gives the control card sequence used to test the newly created tape.

```
@RUN,Z/T SAMPLE,999999,PORTERBIN225,3,150
@ASG,T GPSSE8,F2
@ASG,T MODEL,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
@DATA FILE2,MODEL
****MODEL CHANGE CARDS*****
*****(IF REQUIRED)********
@END
@SETC 4
@XQT GPSSE8.MAPGPS
@ADD MODEL.
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
       END
@SETC 3
@XQT GPSSE8.MAPGPS
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
       END
@FIN
@FIN
```

FIGURE 4. MULTIPLE EXECUTION OF THE SOS MODEL WITH SYSTEM MODEL DESCRIPTION MODIFICATIONS

```
@RUN,//T SAMPLE,999999,PORTERBIN225,3,150
@ASG,T GPSSE8,F2
@ASG,T MODEL,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
*** ANY SUPPORTING FORTRAN SUBPROGRAMS WHICH ARE
*** BEING MODIFIED SHOULD BE INSERTED AS FOLLOWS.
*** (REPLACE 'NAME' WITH THE APPROPRIATE SUBPROGRAM NAME)
@FOR,US GPSSE8.NAME,GPSSE8.NAME
*** INSERT THE CHANGES TO SUBPROGRAM 'NAME'.
***
*** ANY SUPPORTING FORTRAN SUBPROGRAMS WHICH ARE
*** BEING ADDED SHOULD BE INSERTED AS FOLLOWS.
*** (REPLACE 'NAME' WITH THE APPROPRIATE SUBPRGRAM NAME)
@FOR,IS GPSSE8.NAME,GPSSE8.NAME
*** INSERT THE ADDED SUBPROGRAM 'NAME'.
@HDG,P (IN THESE SPACES,ANY DESIRED COMMENT)
@PREP GPSSE8.
@MAP,NX GPSSE8.MAP,GPSSE8.MAPGPS
@DATA FILE2,MODEL
****MODEL CHANGE CARDS*****
*****(IF REQUIRED)********
@END
@XQT GPSSE8.MAPGPS
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
      END
@FIN
@FIN
```

FIGURE 5. EXECUTION OF THE SOS MODEL WITH SUPPLEMENTAL FORTRAN INSTRUCTIONS MODIFICATIONS

```
@RUN,//T SAMPLE,999999,PORTERBIN225,3,200/5000
@ASG,T TAPENAME,T,SAVE02
@ASG,T GPSSE8,F2
@ASG,T DUMMYP,F2
@ASG,T MODEL,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
@HDG,P <<<PROGRAM FILE UPDATE NUMBER AAA>>>
***INSERT ALL SUBPROGRAM UPDATES AND INSERTIONS HERE***
@PREP GPSSE8.
@MAP,XS GPSSE8.MAP,GPSSE8.MAPGPS
@HDG,P <<<SOS MODEL UPDATE MSFC VERSION III-BBB>>>
@DATA,L FILE2,MODEL
-1,1
       ASSEMBLER
       JOB
***INSERT ALL MODEL MODIFICATIONS HERE***
-XXX,XXX
       END
@END
@SETC 2
@BRKPT PUNCH$/DUMMYP
@XQT GPSSE8.MAPGPS
@ADD MODEL.
@BRKPT PUNCH$
@DATA,L DUMMYP,MODEL
-YYY,YYY
       START        501
@END
@REWIND TAPENAME
@COPOUT GPSSE8,TAPENAME
@COPY,GM MODEL,TAPENAME
@ERS GPSSE8.
@REWIND TAPENAME
@COPIN TAPENAME,TPF$.
@FREE TAPENAME.
@HDG,P <REVISION NUMBER CCC TO THE SYSTEM FILE>
@PRT,T
@XQT,LA SYS$*MSFC$.LISTIT
@FIN
```

FIGURE 6. CREATION OF A TAPE FOR UPDATING THE SOS MODEL

A few words of explanation are needed before the user attempts to create a tape containing the inserts and additions to either of the major components of the SOS Model. In the control card sequence (Figure 6), it should be noted that there are two insert cards, one with a location number of XXX and one with a location number of YYY. Before attempting to create the tape, location number XXX needs to be replaced by the location number of the GPSS II START card. At this time in the control card sequence, the System Model Description is assembled with all the statements reassigned new location numbers. The START card has been temporarily replaced by an END card which prohibits the execution of the model and prevents an error occurring during the assembly process. Location number YYY needs to be the new location number of the inserted END card and is determined by knowing the total number of new cards added (CA) and the number of cards deleted (CD) from the current version of the model. The new insert number of the END statement is found by:

$$YYY = XXX + CA - CD$$

At this time in the control card sequence, the START card permanently replaces the END card prior to storing the System Model Description on tape.

The sample control card sequence (Figure 7) is an example of one execution of the SOS Model. Multiple executions are possible by following the procedures outlined previously in this section. The numbers XXXXX on the @ASG, T TAPENAME card should be replaced by the number of the tape to be tested.

F. Mnemonic Inserts to the System Model Description

The control card sequence (Figure 8) is used when the modifications to the System Model Description contain mnemonic location references in the GPSS block types. Mnemonic references are helpful during the development of major modifications to the SOS Model when block location numbers are in a constant state of change. The presence of an insert card containing location number XXX should be noted in Figure 8. The number XXX should be changed to the location number of the START card before an attempt is made to execute this control card sequence. This will place an end card immediately after the START card as is required by mnemonic decks.

```
@RUN,//T SAMPLE,999999,PORTERBIN225,3,150
@ASG,I TAPENAME,T,XXXXX
@ASG,T GPSSE8,F2
@ASG,I FILE2,F2
@ASG,I MODEL,F2
@REWIND TAPENAME
@COPIN TAPENAME,GPSSE8.
@COPY,G TAPENAME,FILE2
@FREE TAPENAME
@DATA FILE2,MODEL
@END
@XQT GPSSE8.MAPGPS
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
      END
@FIN
@FIN
```

FIGURE 7. TEST OF AN UPDATE TAPE

```
@RUN,//T SAMPLE,999999,PORTERBIN225,3,150
@ASG,T GPSSE8,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
@ASG,T MODEL,F2
*** INSERT ALL SUBPROGRAM UPDATES OR INSERTIONS HERE ***
@DATA FILE2,MODEL
-1,1
      ASSEMBLER
      JOB
***INSERT ALL MODEL MODIFICATIONS HERE***
-XXX
      END
@END
@XQT GPSSE8.MAPGPS
@ADD MODEL.
*******COMMENT CARD*******
********DATA CARD*********
@FIN
@FIN
```

FIGURE 8. EXECUTION OF THE SOS MODEL WITH MNEMONIC INSERTS TO THE SYSTEM MODEL DESCRIPTION

G. Example Production Deck and Explanation

Figure 9 is an example of an SOS Model production deck with temporary System Model Description modifications. The System Model portion was modified to change the statements at locations 51, 489, and 513. The modification at location 51 altered the capacity of the Booster maintenance facility and the modification at location 489 changed the number of available launch pads to 3. The modification at location 513 resulted in a mean time for countdown preparations of 2.0 days being associated with block number 80.

Two executions of the SOS Model are required to complete the case study. The @SETC 4 command is used to obtain a listing of the time distribution parameters contained in the Systems Model Description. The @SETC 3 command is used to prevent a listing of the System Model Description for the last execution of the program. An example of the TABLE1 output from the above run is found in Appendix C.

The first requested execution of the SOS Model uses traffic density four with the number of LUT's and launch pads fixed at three and three, respectively. The number of active Boosters and Orbiters for the first study are four and five, respectively. The random number seeds for the entire case study are constant. The random number seed for the Traffic Model Simulator is 3154267131 and the random number seed for the Shuttle Operations Simulator is 4339968911. (Note: Any ten digit odd number with the last digit not ending in five may serve as a random number seed.) The second execution uses the same study conditions as the first execution with the exception of the number of Orbiters being changed to four. Sample output of the summary table and the GPSS II output resulting from the above case study can be found in Appendix D.

```
@RUN,//P SAMPLE,999999,PORTERBIN225,3,150
@ASG,T GPSSE8,F2
@ASG,T MODEL,F2
@COPY,RSA MASTER*GPSSE8.,GPSSE8.
@DATA FILE2,MODEL
-51,51
 2     CAPACITY     1
-489,489
 70    QUEUE        21                       ALL    71     73
-513,513
 79    HELP         K3     K1500 K2000              80             K2500 X10
@END
@SETC 4
@XQT GPSSE8.MAPGPS
@ADD MODEL.
@ADD MODEL.
      DETERMINATION OF FLEET SIZE REQUIREMENTS
4    4     5    3
      END
@SETC 3
@XQT GPSSE8.MAPGPS
@ADD MODEL.
      DETERMINATION OF FLEET SIZE REQUIREMENTS
4    4     4    3                                                        315426713143399668911
      END
@FIN
@FIN
```

FIGURE 9. EXAMPLE PRODUCTION DECK FOR THE EXECUTION OF THE SOS MODEL

REFERENCES

1. Shuttle Operations Simulation Model Description. S&E-012-001-2H, Rev. A. Marshall Space Flight Center, Huntsville, Alabama, February 15, 1971.

2. Univac General Purpose Systems Simulator II. UP-4129 Reference Manual.

3. Ghiglieri, F.J. and Porter, D.G.: Requirements for the Integration of Confidence Interval Logic into the Shuttle Operations Simulation Model. Interim Report ASD-ASTN-1485, Teledyne Brown Engineering Company, Huntsville, Alabama, March 1972.

APPENDIX A. SYSTEM MODEL DESCRIPTION LISTING

PRECEDING PAGE BLANK NOT FILMED


```
          JOB
*                                                                         IIIA  20
*------------------------------------------------------------------------IIIA  30
*SHUTTLE OPERATIONS SIMULATION MODEL MSFC VERSION III-D MAR. 31,1972     IIID  10
*------------------------------------------------------------------------IIIA  50
*                                                                         IIIA  60
*                                                                         IIIA  70
*                                                                         IIIA  80
*                            ALL TIMES EXPRESSED                          IIIA  90
*                            IN THIS PROGRAM ARE IN                       IIIA 100
*                   TERMS OF MILLI WORKING DAYS (1/1000 OF A WORKING DAY) IIIA 110
*                            (WORKING DAY = 16 HOURS)                     IIIB  10
*                                                                         IIIA 120
*                                                                         IIIA 130
*                                                                         IIIA 140
*                 PERFORMANCE TIME DISTRIBUTION CODES                     IIIA 150
*                                                                         IIIA 160
*                            K3      TRIANGULAR                           IIIA 170
*                            K4      NORMAL                               IIIA 180
*                            K8      EXPONENTIAL                          IIIA 190
*                            K9      LOGNORMAL                            IIIA 200
*                            K11     CONSTANT                             IIIA 210
*                            K12     UNIFORM                              IIIC  11
*                            K13     BETA                                 IIIC  12
*                                                                         IIIA 220
*                                                                         IIIA 230
*                                                                         IIIA 240
*------------------------------------------------------------------------IIIA 250
*                                                                         IIIB  20
*                       CURRENT MAXIMUM PROGRAM LIMITS                    IIIB  30
*                                                                         IIIB  40
*      BLOCKS               - 400                                         IIIB  50
*      FACILITIES           - 15                                          IIIB  60
*      STORAGES             - 50                                          IIIB  70
*      QUEUES               - 50                                          IIIB  80
*      LOGIC SWITCHES       - 25                                          IIIB  90
*      SAVEX LOCATIONS      - 50                                          IIIB 100
*      FUNCTIONS            - 10                                          IIIB 110
*      TABLES AND QTABLES - 30    (COMBINED TOTAL)                        IIIB 120
*      VARIABLE STATEMENTS- 30                                            IIID  20
*      PARAMETERS           - 20                                          IIIB 140
*      TRANSACTION NUMBERS- 100                                           IIIB 150
*------------------------------------------------------------------------IIIB 160
*                                                                         IIIB 170
*                                                                         IIIB 180
*------------------------------------------------------------------------IIIB 190
*                                                                         IIIA 260
*                            STORAGE CAPACITIES                           IIIA 270
*                                                                         IIIA 280
1      CAPACITY    1                                                      IIIA 290
2      CAPACITY    2                                                      IIIA 300
3      CAPACITY    2                                                      IIIA 310
4      CAPACITY    9                                                      IIIA 320
*5                                                                        IIIA 330
```

```
*6                                                                   IIIA 340
*7                                                                   IIIA 350
*8                                                                   IIIA 360
*9                                                                   IIIA 370
 10    CAPACITY   1                                                  IIIA 380
 11    CAPACITY   2                                                  IIIA 390
 12    CAPACITY   2                                                  IIIA 400
 13    CAPACITY   9                                                  IIIA 410
 14    CAPACITY   9                                                  IIIA 420
*15                                                                  IIIA 430
 16    CAPACITY   10                                                 IIIA 440
 17    CAPACITY   2                                                  IIIA 450
 18    CAPACITY   2                                                  IIIA 460
 19    CAPACITY   1                                                  IIIA 470
 20    CAPACITY   1                                                  IIIA 480
*21                                                                  IIIA 490
 22    CAPACITY   2                                                  IIIA 500
 23    CAPACITY   2                                                  IIIA 510
*24                                                                  IIIA 520
 25    CAPACITY   2                                                  IIIA 530
*26                                                                  IIIA 540
 27    CAPACITY   2                                                  IIIA 550
 28    CAPACITY   2                                                  IIIA 560
 29    CAPACITY   1                                                  IIIA 570
 30    CAPACITY   1                                                  IIIA 580
 31    CAPACITY   1                                                  IIIA 590
*------------------------------------------------------------------- IIIA 600
*                                                                    IIIA 610
*                                                                    IIIA 620
*------------------------------------------------------------------- IIIA 630
*                                                                    IIIA 640
*                 STORAGE AND QUEUE DEFINITIONS                      IIIB 200
*                                                                    IIIA 660
*        NOTE - THE DEFINITIONS WHICH ARE FOLLOWED BY THE NOTATION (Q)  IIIB 210
*               ALSO REFER TO A QUEUE WHICH IMMEDIATELY PRECEEDS THE    IIIB 220
*               STORAGE. THE DEFINITIONS WHICH ARE NOT FOLLOWED BY THE  IIIB 230
*               NOTATION (Q) HAVE NO QUEUE ASSOCIATED WITH THEM.        IIIB 240
*  1 - BOOSTER SAFING (Q)                                            IIIB 250
*  2 - BOOSTER MAINTENANCE (Q)                                       IIIB 260
*  3 - TRANSPORT BOOSTER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE (Q) IIIB 270
*  4 - BOOSTER STORAGE                                               IIIA 700
*  5                                                                 IIIA 710
*  6                                                                 IIIA 720
*  7                                                                 IIIA 730
*  8                                                                 IIIA 740
*  9                                                                 IIIA 750
* 10 - ORBITER SAFING (Q)                                            IIIB 280
* 11 - ORBITER MAINTENANCE (Q)                                       IIIB 290
* 12 - TRANSPORT ORBITER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE (Q) IIIB 300
* 13 - ORBITER STORAGE                                               IIIA 790
* 14 - ORBITER BOOST TO ORBIT, PERFORM MISSION, RETURN               IIIA 800
* 15                                                                 IIIA 810
* 16 - LAUNCH UMBILICAL TOWER (LUTS) STORAGE                         IIIB 310
* 17 - REMOVE POST LAUNCH LUT FROM PAD                               IIIA 830
* 18 - TRANSPORT LUT TO MAINT.,REFURBISH/TEST,TRANSPORT TO STORAGE (Q) IIIB 320
* 19 - SEPARATE BOOSTER FROM LUT,TRANSPORT BOOSTER TO MAINTENANCE    IIIA 850
```

```
* 20 - MATE AND HOOKUP BOOSTER,ORBITER AND LUT (Q)                        IIIB 330
* 21 - QUEUE IMMEDIATELY PRECEEDING THE PAD FACILITIES (NOTE- THERE IS   IIIB 340
*      NO ASSOCIATED STORAGE)                                             IIIB 350
* 22 - COUNTDOWN PREPARATION                                              IIIA 880
* 23 - COUNTDOWN                                                          IIIA 890
* 24                                                                      IIIA 900
* 25 - REMOVE BOOSTER,ORBITER AND LUT FROM PAD                            IIIA 910
* 26                                                                      IIIA 920
* 27 - REFURBISH LAUNCH PAD                                               IIIA 930
* 28 - LIFTOFF THROUGH SEPARATION                                         IIIA 940
* 29 - READY STORAGE FOR NEXT BOOSTER, ORBITER AND LUT (Q)                IIIB 360
*       (IMMEDIATELY PRIOR TO THE BEGINNING OF THE MATE OPERATION)        IIIA 960
* 30 - TRANSPORT VEHICLE,SEPARATE ORBITER,TRANSPORT ORBITER TO MAINT.    IIIA 970
* 31 - RESTRICTS THE REQUEST FOR VEHICLE ELEMENTS TO ONE MISSION          IIIB 370
*      AT A TIME. SUCCEEDING MISSIONS MUST WAIT UNTIL THE MISSION         IIIB 380
*      IMMEDIATELY IN FRONT OF THEM HAS EITHER SUCCESSFULLY MET ITS       IIIB 390
*      REQUEST OR HAS BEEN CANCELLED DUE TO AN EXCESSIVE WAIT FOR         IIIB 400
*      VEHICLE ELEMENTS. (Q)                                              IIIB 410
*-------------------------------------------------------------------------IIIA 990
*                                                                         IIIA1000
*                                                                         IIIA1010
*-------------------------------------------------------------------------IIIA1020
*                                                                         IIIA1030
*                      FUNCTION DEFINITIONS                               IIIA1040
*                                                                         IIIA1050
*  1 - IS A CONTINUOUS LINEAR FUNCTION WITH A SLOPE OF 1.'ITS END         IIIC  20
*      POINTS ARE LOCATED AT (0,0) AND (999999,999999). IT IS USED        IIIB 420
*      TO CONVERT THE INPUT SYSTEM VARIABLE SAVEX 10 (X10) INTO           IIIB 430
*      FUNCTION 1 (FN1), WHICH CAN BE USED AS A MODIFIER IN AN ADVANCE    IIIB 440
*      BLOCK. THE FUNCTION IS DESCRIBED SUCH THAT THE CONVERSION          IIIB 450
*      IS ONE TO ONE (FN1=X10).                                           IIIB 460
*---------------------------                                              IIIB 470
   1 FUNCTION    X10   C2                                                 IIIA1080
0    0      999999999999                                                  IIIA1090
*-------------------------------------------------------------------------IIIA1100
*                                                                         IIIA1110
*                                                                         IIIA1120
*-------------------------------------------------------------------------IIIA1130
*                                                                         IIIA1140
*                           SAVEX DEFINITIONS                             IIIA1150
*                                                                         IIIA1160
*      NOTE A - UNLESS OTHERWISE STATED THE FOLLOWING DEFINITIONS         IIIB 480
*               REPRESENT THE TOTAL COUNT OF THE TRANSACTIONS             IIIB 490
*               ENTERING THE STATEMENT.                                   IIIB 500
*      NOTE B - SAVEXES 10,11, AND 12 ARE FIRST USED TO INITIALIZE        IIIB 510
*               THE NUMBER OF BOOSTERS, ORBITERS, AND LUTS RESPECTIVELY.IIIB 520
*               THEY ARE THEN USED AS DEFINED BELOW.                      IIIB 530
*                                                                         IIIB 540
*                                                                         IIIB 550
*  1 - BOOSTER AND ORBITER ARE AVAILABLE                                  IIIB 560
*  2 - BOOSTER AND ORBITER ARE NOT AVAILABLE                              IIIB 570
*  3 - BOOSTER IS NOT AVAILABLE                                           IIIB 580
*  4 - ORBITER IS NOT AVAILABLE                                           IIIB 590
*  5 - BOOSTER AND/OR ORBITER AND/OR LUT NOT AVAILABLE                    IIIB 600
*  6 - TOTAL GENERATED LAUNCH REQUESTS                                    IIIB 610
*  7 - ORBITER AND/OR LUT NOT AVAILABLE                                   IIIB 620
```

```
*  8 - LUT NOT AVAILABLE WHEN BOOSTER AND ORBITER ARE AVAILABLE        IIIB 630
*  9 -                                                                 IIIB 640
* 10 - THE INPUT VARIABLE TO FUNCTION 1 (SEE NOTE B ABOVE)             IIIB 650
* 11 - THE LENGTH OF THE NEXT ORBITER MISSION (SEE NOTE B ABOVE)       IIIB 660
* 12 - MISSION TYPE FOR NEXT MISSION          (SEE NOTE B ABOVE)       IIIB 670
* 13 - TIME UNTIL NEXT LAUNCH                                          IIIB 680
* 14 -                                                                 IIIB 690
* 15 - TIME BETWEEN LAUNCH OPPORTUNITIES FOR NEXT MISSION              IIIB 700
* 16 - LENGTH OF ORBITER MISSION FOR NEXT MISSION                      IIIC  30
* 17 - MISSION TYPE OF CURRENT MISSION                                 IIIB 710
* 18 - TIME UNTIL CURRENT LAUNCH                                       IIIB 720
* 19 -                                                                 IIIB 730
* 20 - TIME BETWEEN LAUNCH OPPORTUNITIES FOR CURRENT MISSION           IIIB 740
* 21 - TIME OF FIRST LAUNCH OPPORTUNITY FOR CURRENT MISSION            IIIB 750
* 22 - VEHICLES REMOVED FROM PAD (FAILURE ON PAD)                      IIIB 760
* 23 - VEHICLES REMOVED FROM PAD (CONSTRAINS NEXT LAUNCH)              IIIB 770
* 24 - VEHICLES REMOVED FROM PAD (NO SECOND LAUNCH WINDOW)             IIIB 780
* 25 - LAUNCHES MISSED DUE TO PAD UNAVAILABILITY                       IIIB 790
* 26 - THE NUMBER OF VEHICLES LAUNCHED AT AN ALTERNATE LAUNCH WINDOW   IIIB 800
* 27 - LAUNCH WINDOW INDICATOR                                         IIIC  40
* 28 - NUMBER OF LAUNCHES AT PRIMARY WINDOW                            IIIC  50
* 29 - NUMBER OF TIMES A VEHICLE WAS REPAIRED ON THE PAD               IIIC  60
* 30 - TOTAL SUCCESSFULLY ACCOMPLISHED LAUNCHES                        IIIB 840
* 31 -                                                                 IIIB 850
* 32 -                                                                 IIIC  61
* 33 -                                                                 IIIC  62
* 34 -                                                                 IIIC  63
* 35 - AMOUNT OF TIME AFTER PRIMARY WINDOW AT WHICH LAUNCH OCCURS      IIIC  64
*------------------------------------------------------------------IIIA1500
*                                                                  IIIA1510
*                                                                  IIIA1520
*------------------------------------------------------------------IIIA1530
*                                                                  IIIA1540
*                     VARIABLE DEFINITIONS                         IIIA1550
*                                                                  IIIA1560
*  1 - GROUND PROCESSING TIME FOR EACH BOOSTER                         IIIB 860
         (TIME OUT OF STORAGE)-(FLIGHT TIME)+(TIME IN STORAGE)         IIIB 870
*                                                                      IIIB 880
*  2 - TIME LEFT BEFORE LAUNCH (USED TO DETERMINE IF SUFFICIENT TIME   IIIB 890
*      REMAINS BEFORE LAUNCH TO INITIATE AND COMPLETE COUNTDOWN)       IIIB 900
*        (TIME TO LAUNCH FROM EARLIEST POSSIBLE STORAGE EXIT)-(ELAPSED IIIB 910
*         TIME FROM EARLIEST POSSIBLE STORAGE EXIT)                    IIIB 920
*         NOTE- IF FIRST LAUNCH WINDOW IS MISSED V2 WILL BE NEGATIVE.  IIIC  70
*               THIS IS TAKEN INTO ACCOUNT IN LATER VARIABLES.         IIIC  80
*                                                                      IIIB 930
*  3 - FLOATING HOLD TIME BETWEEN COMPLETION OF ALL PRECOUNTDOWN       IIIB 940
*      OPERATIONS AND THE INITIATION OF FINAL COUNTDOWN FOR THE        IIIB 950
*      FIRST LAUNCH OPPORTUNITY                                        IIIB 960
*        (TIME LEFT BEFORE LAUNCH)-(COUNTDOWN TIME)                    IIIB 970
*                                                                      IIIB 980
*  4 - TIME TO SECOND LAUNCH WINDOW                                    IIIC  90
*      (TIME FIRST WINDOW WAS MISSED BY)+(TIME BETWEEN FIRST AND       IIIC 100
*        SECOND WINDOW)                                                IIIC 110
*                                                                      IIIB1040
*  5 - MAXIMUM TIME TO PROCESS VEHICLE FROM MATE AND HOOKUP INITIATION IIIB1050
*      TO LAUNCH. THIS ALSO REPRESENTS THE EARLIEST POSSIBLE STORAGE   IIIB1060
```

```
*          EXIT FOR THE VEHICLE ELEMENTS.                                IIIB1070
*           (ASSIGN CONSTANT = SUM OF MAXIMUM TIMES IN EACH              IIIB1080
*            PRELAUNCH OPERATION)                                        IIIB1090
*                                                                        IIIB1100
*  6 - LENGTH OF COUNTDOWN                                               IIIB1110
*           (PREASSIGNED CONSTANT)                                       IIIB1120
*                                                                        IIIB1130
*  7 - GROUND PROCESSING TIME FOR EACH ORBITER                           IIIB1140
*           (TIME OUT OF STORAGE)-(FLIGHT TIME)+(TIME IN STORAGE)        IIIB1150
*                                                                        IIIB1160
*  8 - MINIMUM TIME TO PROCESS VEHICLE FROM MATE AND HOOKUP INITIATION   IIIB1170
*          TO LAUNCH. THIS ALSO REPRESENTS THE LATEST POSSIBLE STORAGE   IIIB1180
*          EXIT FOR THE VEHICLE ELEMENTS IF THE FIRST LAUNCH WINDOW IS   IIIB1190
*          TO BE MET.                                                    IIIB1200
*           (ASSIGN CONSTANT = SUM OF MINIMUM TIMES IN EACH              IIIB1210
*            PRELAUNCH OPERATION)                                        IIIB1220
*                                                                        IIIB1230
*  9 - TIME VEHICLES ARE HELD IN READY STORAGE PRIOR TO EARLIEST         IIIB1240
*          POSSIBLE INITIATION OF MATE AND HOOKUP RELATIVE TO THEIR MISSION IIIB1250
*           (LAUNCH TIME)-(CURRENT TIME)-(VARIABLE 5)                    IIIB1260
*                                                                        IIIB1290
* 10 - THE NUMBER OF SIMULATIONS FOR WHICH STATISTICS ARE COLLECTED      IIIB1300
*           (ASSIGN CONSTANT = ONE LESS THAN NUMBER OF SIMULATIONS       IIIB1310
*            SPECIFIED ON THE START CARD)                                IIIB1320
*                                                                        IIIB1330
* 11 - MAXIMUM TIME A REQUEST MAY WAIT FOR ANY VEHICLE ELEMENT           IIIB1340
*          AFTER THE EARLIEST POSSIBLE MATE AND HOOKUP INITIATION.       IIIB1350
*           (MAXIMUM TIME TO LAUNCH(V5))-(MINIMUM TIME TO LAUNCH(V8))    IIIB1360
*                                                                        
* 12 - TIME TO BEGINING OF FINAL COUNTDOWN FOR THE SECOND WINDOW         IIIC 120
*          (TIME TO SECOND LAUNCH WINDOW)-(LENGTH OF COUNT DOWN)         IIIC 130
*                                                                        IIIC 140
* 13 - LENGTH OF TIME BETWEEN EVEN AND ODD WINDOWS                       IIIC 150
*          (23.5 HOURS)-(TIME BETWEEN FIRST AND SECOND WINDOW)           IIIC 160
*                                                                        IIIC 170
* 14 - TIME LEFT TO BEGINING OF FINAL COUNTDOWN FOR LAUNCH AT THE        IIIC 180
*          CURRENT WINDOW.                                               IIIC 190
*          (LENGTH OF TIME BETWEEN FIRST WINDOW AND CURRENT WINDOW)+(TIME IIIC 200
*           THE FIRST WINDOW WAS MISSED BY)-(LENGTH OF FINAL COUNTDOWN)  IIIC 210
*                                                                        IIIC 220
* 15 - DIVIDE WINDOW NUMBER BY 2 DISCARD QUOTIENT AND RETAIN REMAINDER   IIIC 230
*          IF V15=0 THE WINDOW IS EVEN. IF V15=1 THE WINDOW IS ODD       IIIC 240
* ------------------------                                               IIIA1760
 1     VARIABLE     MP11-P1+P2                                           IIIB1361
 2     VARIABLE     V5-M1                                                IIIA1780
 3     VARIABLE     V2-V6                                                IIIA1790
 4     VARIABLE     P7+V2                                                IIIC 250
 5     VARIABLE     K5325                                                IIIA1810
 6     VARIABLE     K83                                                  IIIA1820
 7     VARIABLE     MP11-P1+P4                                           IIIB1362
 8     VARIABLE     K3925                                                IIIA1840
 9     VARIABLE     X21-C1-V5                                            IIIA1850
10     VARIABLE     K500                                                 IIIA1860
11     VARIABLE     V5-V8                                                IIIA1870
12     VARIABLE     V4-V6                                                IIIC 260
13     VARIABLE     K979-P7                                              IIIC 270
```

```
14    VARIABLE     X35+V2-V6                                         IIIC 280
15    VARIABLE     X27(K2                                            IIIC 290
*-----------------------------------------------------------------------IIIA1880
*                                                                       IIIB1370
*                                                                       IIIB1380
*-----------------------------------------------------------------------IIIB1390
*                                                                       IIIB1400
*                        PARAMETER DEFINITIONS                          IIIB1410
*                                                                       IIIB1420
*  1 - INDEPENDENT TOTAL FLIGHT TIME OF EACH BOOSTER AND ORBITER        IIIB1430
*  2 - TIME IN STORAGE FOR THE BOOSTER OF THE CURRENT MISSION           IIIB1440
*  3 - LENGTH OF CURRENT ORBITER MISSION                                IIIB1450
*  4 - MISSION TYPE UNTIL REDEFINED AS TIME IN STORAGE FOR THE ORBITER  IIIB1460
*      OF THE CURRENT MISSION                                           IIIB1470
*  5 - TIME TO THE CURRENT LAUNCH                                       IIIB1480
*  6 -                                                                  IIIB1490
*  7 - TIME BETWEEN LAUNCH OPPORTUNITIES FOR CURRENT LAUNCH             IIIB1500
*  8 - TIME FROM LANDING TO STORAGE ENTRY                               IIIB1510
*  9 - DEFINES PAD TO BE SEIZED FOR CURRENT MISSION                     IIIB1520
* 10 - INITIALIZATION OF THE NUMBER OF BOOSTERS,ORBITERS AND LUTS       IIIB1530
* 11 - IS MARKED SUCH THAT THE LEAVE STORAGE TO LAUNCH TIME CAN BE      IIIB1540
*      TABULATED. THE AMOUNT OF TIME EACH BOOSTER AND ORBITER           IIIB1550
*      SPENDS OUT OF STORAGE IS ALSO TABULATED BY THIS PARAMETER.       IIIB1560
*-----------------------------------------------------------------------IIIB1570
*                                                                       IIIA1890
*                                                                       IIIA1900
* ----------------------------                                          IIIA1910
* START NEW PROBLEM AND READ IN FLEET SIZE                              IIIA1920
*                                                                       IIIA1930
1     GENERATE     0     1               2                              IIIA1940
*                  BOOSTER  ORBITERS                LUT                 IIIA1950
2     HELP         K7    X11    X10      3          X12                 IIIA1960
*                                                                       IIIA1970
* ENTER X10 NUMBER OF ORBITERS INTO STORAGE 13                          IIIA1980
*                                                                       IIIA1990
3     ASSIGN       10    X10             4                              IIIA2000
4     ENTER        13    P10             5                              IIIA2010
*                                                                       IIIA2020
* ENTER X11 NUMBER OF BOOSTERS INTO STORAGE 4                           IIIA2030
5     ASSIGN       10    X11             6                              IIIA2040
6     ENTER        4     P10             7                              IIIA2050
*                                                                       IIIA2060
*ENTER X12 NUMBER OF LUTS IN STORAGE 16                                 IIIA2070
*                                                                       IIIA2080
7     ASSIGN       10    X12             8                              IIIA2090
8     ENTER        16    P10             15                             IIIB1571
* ----------------------------                                          IIIA2120
*                                                                       IIIA2130
*                                                                       IIIA2140
* ----------------------------                                          IIIA2150
*                                                                       IIIA2160
* GENERATE SUFFICIENT TIME TO MEET THE FIRST MISSION REQUEST            IIIA2170
*                                                                       IIIA2180
* INITIALIZE SAVEX 16 THROUGH 21 WITH FIRST TRANSACTION START TIME      IIIA2200
*                                                                       IIIA2210
*                                                                       IIIA2220
```

```
 9   SAVEX      16   X11             10                      IIIA2230
10   SAVEX      17   X12             11                      IIIA2240
11   SAVEX      19   X13             12                      IIIA2250
12   SAVEX      20   X15             13                      IIIA2260
*INITIALIZE TIME TO LAUNCH CUMMULATOR WITH SIMULATION START TIME IIIA2270
13   SAVEX      21   C1       BOTH  14    15                 IIIA2280
*                                                            IIIA2290
* DETERMINE IF FIRST LAUNCH REQ . HAS ENOUGH TIME TO MEET ITS LAUNCH IIIA2300
*                                                            IIIA2310
14   COMPARE    X18  GE   V5         16                      IIIA2320
*                                                            IIIA2330
* CALL HELP & DETERMINE A MORE REALISTIC FIRST LAUNCH REQ.   IIIA2340
*                                                            IIIA2350
15   HELP       K1   X11  X12        9          X13   X15    IIIA2360
*                                                            IIIA2370
* RETURN AND REINITIALIZE SAVEX 16 THROUGH 21                IIIA2380
*                                                            IIIA2390
16   LOGIC      R3                   17                      IIIA2400
17   TERMINATE                                               IIIA2410
*-----------------------------------------------------------IIIA2420
*                                                            IIIA2430
*                                                            IIIA2440
*                                                            IIIA2450
*                                                            IIIA2460
*                                                            IIIA2470
*                                                            IIIA2480
*                                                            IIIA2490
*                                                            IIIA2500
*                                                            IIIA2510
*-----------------------------------------------------------IIIA2520
* THE SIMULATION TRANSACTIONS ARE GENERATED FOLLOWING THIS POINT IIIA2530
*-----------------------------------------------------------IIIA2540
*                                                            IIIA2550
*                                                            IIIA2560
*                                                            IIIA2570
*                                                            IIIA2580
*                                                            IIIA2590
*                                                            IIIA2600
*                                                            IIIA2610
*                                                            IIIA2620
*                                                            IIIA2630
*-----------------------------                               IIIA2640
* GENERATE LAUNCHES BASED ON PROJECTED TRAFFIC REQUIREMENT   IIIA2650
*                                                            IIIA2660
18   GENERATE                        19                      IIIA2670
19   GATE       LR3                  20                      IIIA2680
20   LOGIC      S3                   21                      IIIA2690
*                                                            IIIA2700
*COUNT THE NUMBER OF LAUNCH REQUESTS                         IIIA2710
*                                                            IIIA2720
21   SAVEX      5+   K1              22                      IIIA2730
*                                                            IIIA2740
*CALL HELP TO GENERATE ATTRIBUTES OF NEXT LAUNCH             IIIA2750
*                                                            IIIA2760
22   HELP       K1   X11  X12        23         X13   X15    IIIB1572
*                                                            IIIA2780
```

```
*-----------------------------                                           IIIA2790
*                                                                       IIIA2800
*                                                                       IIIA2810
*-----------------------------                                           IIIA2820
*ASSIGN ATTRIBUTES OF CURRENT LAUNCH TO TRANSACTION                     IIIA2830
*  3 - LENGTH OF THE ORBITER MISSION                                    IIIA2840
*  4 - MISSION TYPE                                                     IIIA2850
*  5 - TIME UNTIL NEXT LAUNCH                                           IIIA2860
*  7 - TIME BETWEEN LAUNCH OPPORTUNITIES                                IIIA2870
*                                                                       IIIA2880
 23   ASSIGN      3     X16                24                           IIIA2890
 24   ASSIGN      4     X17                25                           IIIA2900
 25   ASSIGN      5     X18                26                           IIIA2910
 26   ASSIGN      7     X20                27                           IIIA2920
 27   SAVEX      16     X11                28                           IIIA2930
 28   SAVEX      17     X12                29                           IIIA2940
 29   SAVEX      18     X13                30                           IIIA2950
 30   SAVEX      20     X15                31                           IIIA2960
 31   SAVEX      21+    X18                32                           IIIA2970
 32   SPLIT                                33    37                     IIIA2980
 33   SAVEX      10     X13                34                           IIIA2990
 34   ADVANCE                              35           1      FN1      IIIA3000
 35   LOGIC      R3                        36                           IIIA3010
 36   TERMINATE                                                         IIIA3020
*-----------------------------                                           IIIA3030
*                                                                       IIIA3040
*                                                                       IIIA3050
* ---------------------------                                           IIIA3060
* READY STORAGE QUEUE TIME INCLUDED IN TABLE 2 AND TABLE 22             IIIA3070
 37   QUEUE      29                        38                           IIIA3080
*                                                                       IIIA3090
*ADVANCE UNTIL TIME TO MATE THE VEHICLES                                IIIA3100
*                                                                       IIIA3110
 38   ENTER      29                        39                           IIIA3120
 39   SAVEX      10     V9                 40                           IIIA3130
 40   ADVANCE                              41           1      FN1      IIIA3140
 41   LEAVE      29                        42                           IIIA3150
 42   QUEUE      31                        43                           IIIA3160
 43   ENTER      31                        44                           IIIA3170
* ---------------------------                                           IIIA3180
 44   MARK                          BOTH   45    47                     IIIA3190
*                                                                       IIIA3200
*                                                                       IIIA3210
* ---------------------------                                           IIIA3220
*CHECK IF TOO MUCH DELAY HAS OCCURRED TO MEET THE LAUNCH                IIIA3230
*                                                                       IIIA3240
 45   COMPARE    M1    G    V11            46                           IIIA3250
 46   LEAVE      31                        250                          IIIA3260
 47   GATE       SNE4               BOTH   45    48                     IIIA3270
 48   GATE       SNE13              BOTH   45    49                     IIIA3280
 49   GATE       SNE16              BOTH   51    50                     IIIA3290
*--------------------                                                   IIIB1580
* CHECK TO SEE IF PAD IS AVAILABLE FOR THIS MISSION.                    IIIB1590
* IF NOT GO TO MISSED LAUNCH STATISTICS SECTION                         IIIB1600
*                                                                       IIIB1610
 50   LEAVE      31                        259                          IIIA3300
```

```
000447     51  COMPARE    Q21   E     K0          52                              IIIA3310
000448     52  LEAVE      31                      53                              IIIC 300
000449     53  TABULATE   6                       54                              IIIC 310
000450     * ---------------------------                                          IIIA3330
000451     *                                                                      IIIA3340
000452     *                                                                      IIIA3350
000453     *---------------------------                                           IIIA3360
000454     * COUNT THE NUMBER OF TIMES THE BOOSTER,ORBITER AND LUT ARE AVAILABLE  IIIA3370
000455     * REMOVE ONE BOOSTER, ORBITER, AND LUT FROM THEIR RESPECTIVE STORAGES. IIIB1620
000456     * AND TABULATE THE TIME THESE VEHICLE ELEMENTS SPENT IN STORAGE.       IIIB1630
000457     *                                                                      IIIA3380
000458     54  SAVEX      1+    K1                55                              IIIA3390
000459     55  LEAVE      4                       56                              IIIA3400
000460     56  HELP       K6    K1    C1          57          X10                 IIIA3410
000461     57  TABULATE   2                       58                              IIIA3420
000462     58  ASSIGN     2     X10               59                              IIIA3430
000463     59  LEAVE      13                      60                              IIIA3440
000464     60  HELP       K6    K2    C1          61          X10                 IIIA3450
000465     61  TABULATE   22                      62                              IIIA3460
000466     62  ASSIGN     4     X10               63                              IIIA3470
000467     63  LEAVE      16                      64                              IIIA3480
000468     *---------------------------                                           IIIA3490
000469     *   PARAMETER 11 MEASURES LEAVE STORAGE TO LAUNCH TIME                 IIIA3500
000470     64  MARK       11                      65                              IIIA3510
000471     *                                                                      IIIA3520
000472     *                                                                      IIIA3530
000473     * ---------------------------                                          IIIA3540
000474     * ENTER WAITING LINE FOR MATE AND HOOKUP FACILITY                      IIIA3550
000475     *                                                                      IIIA3560
000476     65  QUEUE      20                      66                              IIIA3570
000477     *MATE BOOSTER,ORBITER AND LUT                                     IDST IIIA3580
000478     66  ENTER      20                      67                              IIIA3590
000479     *   TRIANGULAR       MIN   MODE                    MAX                 IIIA3600
000480     67  HELP       K3    K1800 K2000       68          K2200 X10           IIIC 311
000481     68  ADVANCE                            69          1     FN1           IIIA3620
000482     69  LEAVE      20                      70                              IIIA3630
000483     * ---------------------------                                          IIIA3640
000484     *                                                                      IIIA3650
000485     *                                                                      IIIA3660
000486     * ---------------------------                                          IIIA3680
000487     * WAITING LINE FOR PAD                                                 IIIA3690
000488     *                                                                      IIIA3700
000489     70  QUEUE      21              ALL     71   72                         IIIA3710
000490     71  COMPARE    F1    E     K0          74                              IIIA3720
000491     72  COMPARE    F2    E     K0          75                              IIIA3730
000492     73  COMPARE    F3    E     K0          76                              IIIA3740
000493     74  ASSIGN     9     K1                77                              IIIA3750
000494     75  ASSIGN     9     K2                77                              IIIA3760
000495     76  ASSIGN     9     K3                77                              IIIA3770
000496     * ---------------------------                                          IIIA3780
000497     *                                                                      IIIA3790
000498     *                                                                      IIIA3800
000499     * ---------------------------                                          IIIA3810
000500     *   SEIZE THE FIRST AVAILABLE PAD FOR THIS MISSION AND                 IIIA3820
000501     *   REMOVE THIS PAD FROM THE PAD AVAILABLE LIST                        IIIA3830
000502     *                                                                      IIIA3840
```

```
 77   SEIZE        *9                   78                              IIIA3850
 * ----------------------------                                          IIIA3860
 *                                                                       IIIA3880
 *                                                                       IIIA3890
 *----------------------------                                           IIIA3900
 *COUNTDOWN PREPARATION                                            IDST IIIA3910
 *TRANSFER LUT AND VEHICLE, INSTALL INTO PAD, AND COUNTDOWN PREPS        IIIA3920
 *                                                                       IIIA3930
 78   ENTER        22                   79                              IIIA3940
 *    TRIANGULAR        MIN    MODE                   MAX                IIIA3950
 79   HELP         K3   K2000  K2500          80      K3000  X10         IIIC 312
 80   ADVANCE                                 81      1      FN1         IIIA3970
 81   LEAVE        22                         82                         IIIC 320
 * ---------------------------                                           IIIA3990
 *                                                                       IIIA4000
 *                                                                       IIIA4010
 *--------------------------------------------------------------------- IIIC 330
 *         DETERMINE WHICH PATH THROUGH THE LAUNCH LOGIC IS TO BE TAKEN  IIIC 340
 *         ESTABLISH TIME LEFT TO FIRST LAUNCH WINDOW                    IIIC 350
 82   SAVEX        10   V2          ALL  83   85                         IIIC 360
 *                                                                       IIIC 370
 *                                                                       IIIC 380
 *                                                                       IIIC 390
 *         IS TIME TO LAUNCH GE LENGTH OF FINAL COUNTDOWN. IF SO GO TO   IIIC 400
 *         PRIMARY PATH                                                  IIIC 410
 *                                                                       IIIC 420
 *PATH ONE                                                               IIIC 430
 83   COMPARE      V2   GE   V6          86                              IIIC 440
 *                                                                       IIIC 450
 *         TIME TO LAUNCH IS NOT GE LENGTH OF FINAL COUNTDOWN, IS THERE  IIIC 460
 *         ANOTHER LAUNCH WINDOW. IF SO GO TO SECONDARY PATH             IIIC 470
 *                                                                       IIIC 480
 *PATH TWO                                                               IIIC 490
 84   COMPARE      P7   G    KO          105                             IIIC 500
 *                                                                       IIIC 510
 *         TIME TO LAUNCH IS NOT GE LENGTH OF FINAL COUNTDOWN, AND THERE IIIC 520
 *         IS NOT ANOTHER LAUNCH WINDOW. COUNT NUMBER OF VEHICLES REMOVED IIIC 530
 *         FROM THE PAD FOR LACK OF ANOTHER LAUNCH WINDOW AND GO TO THE  IIIC 540
 *         VEHICLE REMOVAL PATH.                                         IIIC 550
 *                                                                       IIIC 560
 *PATH THREE                                                             IIIC 570
 85   SAVEX        24+  K1                   210                         IIIC 580
 *--------------------------------------------------------------------- IIIC 590
 *                                                                       IIIC 600
 *                                                                       IIIC 610
 *--------------------------------------------------------------------- IIIC 620
 *                                                                       IIIC 630
 *         THIS IS THE PRIMARY PATH THROUGH THE LAUNCH LOGIC.            IIIC 640
 *                                                                       IIIC 650
 *                                                                       IIIC 660
 *                                                                       IIIC 670
 *         ESTABLISH LENGTH OF HOLD TO BEGINING OF FINAL COUNTDOWN       IIIC 680
 86   SAVEX        10   V3                   87                          IIIC 690
 *         TABULATE LENGTH OF HOLD TO BEGINING OF FINAL COUNTDOWN        IIIC 700
 87   TABULATE     4                         88                          IIIC 710
 *         SET FIRST LAUNCH WINDOW INDICATOR                             IIIC 720
```

```
 88   SAVEX       27    K1                     89                               IIIC 730
 *         ESTABLISH A BASE POINT OF ZERO FOR CALCULATING TIME BETWEEN ,         IIIC 740
 *         FIRST AND SUBSEQUENT WINDOWS IN CASE THIS VEHICLE HAS A FINAL         IIIC 750
 *         COUNTDOWN FAILURE AND CAN BE REPAIRED ON THE PAD.                     IIIC 760
 89   SAVEX       35    KC                     90                               IIIC 770
 *         ADVANCE TO BEGINING OF FINAL COUNTDOWN                                IIIC 780
 90   ADVANCE                                  91           1      FN1          IIIC 790
 *-----------------------------                                                  IIIC 800
 *                                                                               IIIC 810
 *                                                                               IIIC 820
 *-----------------------------                                                  IIIC 830
 * FINAL COUNTDOWN                                                          IDST IIIC 840
 *                                                                               IIIC 850
 *         BEGIN FINAL COUNTDOWN                                                 IIIC 860
 *                                                                               IIIC 870
 91   ENTER       23                           92                               IIIC 880
 *         ESTABLISH LENGTH OF FINAL COUNTDOWN                                   IIIC 890
 92   HELP        K11   K83                    93                  X10          IIIC 900
 *         ADVANCE TO END OF FINAL COUNTDOWN                                     IIIC 910
 93   ADVANCE                                  94           1      FN1          IIIC 920
 *         LEAVE FINAL COUNTDOWN                                                 IIIC 930
 94   LEAVE       23                     .05   95    100                        IIIC 940
 *-----------------------------                                                  IIIC 950
 *                                                                               IIIC 960
 *                                                                               IIIC 970
 *-----------------------------                                                  IIIC 980
 *         FAILURE HAS NOT OCCURED DURING FINAL COUNTDOWN. COLLECT               IIIC 990
 *         APPROPRIATE STATISTICS AND INITIATE LAUNCH OF VEHICLE.                IIIC 995
 *         (95 PERCENT PROBABILITY)                                              IIIC1000
 *                                                                               IIIC1010
 *         TABULATE WHICH WINDOW LAUNCH OCCURS AT                                IIIC1020
 95   TABULATE    9                      BOTH  96    98                         IIIC1030
 *         DID LAUNCH OCCUR AT PRIMARY WINDOW                                    IIIC1040
 96   COMPARE     X27   E     K1               97                               IIIC1050
 *         COUNT NUMBER OF LAUNCHES AT THE PRIMARY WINDOW                        IIIC1060
 97   SAVEX       28+   K1                     118                              IIIC1070
 *         LAUNCH DID NOT OCCUR AT THE PRIMARY WINDOW                            IIIC1080
 *         TABULATE AMOUNT OF TIME AFTER PRIMARY WINDOW LAUNCH OCCURED           IIIC1090
 98   TABULATE    8                            99                               IIIC1100
 *         COUNT NUMBER OF LAUNCHES AT A SUBSEQUENT WINDOW                       IIIC1110
 99   SAVEX       26+   K1                     118                              IIIC1120
 *-----------------------------                                                  IIIC1130
 *                                                                               IIIC1140
 *                                                                               IIIC1150
 *-----------------------------                                                  IIIC1160
 *         FAILURE HAS OCCURED DURING FINAL COUNTDOWN (5 PERCENT PROB.)          IIIC1170
 *         REMOVE 50 PERCENT OF THESE VEHICLES FROM PAD. ATTEMPT TO LAUNCHIIIC1180
 *         50 PERCENT AT A LATER WINDOW.                                         IIIC1190
100   ADVANCE                            .50   101   102                        IIIC1200
 *         COUNT NUMBER OF VEHICLES REMOVED FROM PAD DUE TO A FINAL              IIIC1210
 *         COUNTDOWN FAILURE.                                                    IIIC1220
101   SAVEX       22+   K1                     210                              IIIC1230
 *         COUNT NUMBER OF TIMES A VEHICLE WAS REPAIRED ON THE PAD               IIIC1240
102   SAVEX       29+   K1               BOTH  103   104                        IIIC1250
 *         IS THERE ANOTHER LAUNCH OPPORTUNITY                                   IIIC1260
103   COMPARE     P7    G     K0               112                              IIIC1270
```

```
*         THERE IS NOT ANOTHER LAUNCH OPPORTUNITY
*         COUNT NUMBER OF VEHICLES REMOVED DUE TO NO FURTHER
*         LAUNCH WINDOW. THEN GO TO VEHICLE REMOVAL PATH.
     104  SAVEX       24+    K1                 210
*         THERE IS ANOTHER LAUNCH OPPORTUNITY
*         SET SECOND LAUNCH WINDOW INDICATOR
*-----------------------------------------------------------------------
*
*
*-----------------------------------------------------------------------
*
*         THIS THE SECONDARY PATH THROUGH THE LAUNCH LOGIC. ONLY VEHICLES
*         WHICH CANNOT LAUNCH ON THE PRIMARY WINDOW FLOW THROUGH THIS
*         SECTION
*
     105  SAVEX       27     K2                 106
*         TABULATE AMOUNT OF TIME LEFT TO FIRST WINDOW (MAY BE NEGATIVE).
     106  TABULATE    7                 BOTH    107    111
*         IS THERE TIME TO LAUNCH AT THE SECOND WINDOW
     107  COMPARE     V4     GE    V6           108
*         THERE IS TIME TO LAUNCH AT THE SECOND WINDOW
*         ESTABLISH LENGTH OF HOLD TO BEGINING OF FINAL COUNTDOWN
*         FOR THE SECOND WINDOW.
     108  SAVEX       10     V12                109
*         ESTABLISH LENGTH OF TIME BETWEEN FIRST AND SECOND WINDOW
     109  SAVEX       35     P7                 110
*         TABULATE HOLD TO BEGINING OF FINAL COUNTDOWN FOR THE
*         SECOND WINDOW
     110  TABULATE    5                         90
*
*
*         THERE IS NOT TIME TO LAUNCH AT THE SECOND WINDOW
*         ESTABLISH LENGTH OF TIME BETWEEN FIRST AND SECOND WINDOW
     111  SAVEX       35     P7                 112
*         INCREMENT WINDOW INDICATOR TO NEXT WINDOW
     112  SAVEX       27+    K1         BOTH    113    115
*         DETERMINE IF THE WINDOW BEING TRIED IS EVEN OR ODD
     113  COMPARE     V15    G     KO           114
*         THE WINDOW IS ODD.
*         ESTABLISH LENGTH OF TIME BETWEEN FIRST AND SUBSEQUENT
*         ODD WINDOWS
     114  SAVEX       35+    V13        BOTH    116    112
*
*
*         THE WINDOW IS EVEN
*         ESTABLISH LENGTH OF TIME BETWEEN FIRST AND SUBSEQUENT
*         EVEN WINDOWS
     115  SAVEX       35+    P7         BOTH    116    112
*         IS THERE TIME TO LAUNCH AT THIS WINDOW. IF NOT TRY NEXT WINDOW.
     116  COMPARE     V14    GE    KO           117
*         THERE IS TIME TO LAUNCH
*         ESTABLISH LENGTH OF HOLD TO BEGINING OF FINAL COUNTDOWN
     117  SAVEX       10     V14                110
*-----------------------------------------------------------------------
*
*
```

```
*                                                                     IIIC1830
*-----------------------------                                        IIIA4570
*                                                                     IIIA4580
*                                                                     IIIA4590
*                                                                     IIIA4900
*                                                                     IIIA4910
*-----------------------------                                        IIIA4920
* FAILURE HAS NOT OCCURED DURING TIME ON PAD. INITIATE LAUNCH OF SHUTTLEIIIA4930
*                                                                     IIIA4940
 118  SPLIT                      119   120                            IIIA4950
 119  MATCH       128            140                                  IIIA4960
 120  SPLIT                      121   142                            IIIA4970
 121  SPLIT                      136   122                            IIIA4980
*-----------------------------                                        IIIA4990
*                                                                     IIIA5000
*                                                                     IIIA5010
*---------------------------------------------------------------------IIIC1840
*           LUT REMOVAL AND PAD REFURBISHMENT PATH                    IIIC1850
*           THIS PATH IS SIMULTANEOUS WITH THE FLIGHT AND REFURBISH PATH IIIC1860
*                                                                     IIIC1870
*                                                                     IIIC1880
*REMOVE POST LAUNCH LUT FROM PAD                                 IDST IIIA5030
 122  ENTER       17             123                                  IIIA5040
 123  HELP        K3    K1250 K1500  124         K1750 X10            IIIA5050
 124  ADVANCE                    125             1     FN1            IIIA5060
 125  LEAVE       17             126                                  IIIA5070
 126  SPLIT                      127   130                            IIIA5080
 127  ASSEMBLE    2              128                                  IIIA5090
 128  MATCH       119            129                                  IIIA5100
 129  TERMINATE                                                       IIIA5110
*-----------------------------                                        IIIA5120
*                                                                     IIIA5130
*                                                                     IIIA5140
*-----------------------------                                        IIIA5150
*TRANSPORT LUT TO MAINT.,REFURBISH/TEST,TRANSPORT TO STORAGE     IDST IIIA5160
 130  QUEUE       18             131                                  IIIA5170
 131  ENTER       18             132                                  IIIC1881
 132  HELP        K3    K2700 K3000  133         K3300 X10            IIIA5190
 133  ADVANCE                    134             1     FN1            IIIA5200
 134  LEAVE       18             135                                  IIIA5210
 135  ENTER       16    K1       129                                  IIIA5220
*-----------------------------                                        IIIA5230
*                                                                     IIIA5340
*                                                                     IIIA5350
*-----------------------------                                        IIIA5360
*REFURBISH LAUNCH PAD                                            IDST IIIA5370
*                                                                     IIIA5380
 136  ENTER       27             137                                  IIIA5390
*     TRIANGULAR  MIN   MODE                     MAX                  IIIC1882
 137  HELP        K3    K950  K1000  138         K1050 X10            IIIA5410
 138  ADVANCE                    139             1     FN1            IIIA5420
 139  LEAVE       27             127                                  IIIA5430
 140  RELEASE     *9             141                                  IIIA5440
 141  TERMINATE                                                       IIIA5450
*---------------------------------------------------------------------IIIC1890
*                                                                     IIIC1900
```

```
*        FLIGHT AND REFURBISH PATH                                        IIIC1910
*                                                                         IIIA5470
*                                                                         IIIA5480
 142  TABULATE     10                   143                               IIIA5490
*                                                                         IIIA5500
*-----------------------------                                            IIIA5510
*LIFTOFF THROUGH SEPARATION                                          IDST IIIA5520
*                                                                         IIIA5530
 143  ENTER        28                   144                               IIIA5540
 144  HELP         K11    K2            145                  X10          IIIA5550
 145  ADVANCE                           146            1     FN1          IIIA5560
 146  LEAVE        28                   147                               IIIA5570
*-----------------------------                                            IIIA5580
*                                                                         IIIA5590
*                                                                         IIIA5600
*-----------------------------                                            IIIA5610
*                                                                         IIIA5620
*PARAMETER 1 IS USED TO CUMULATE THE TOTAL INFLIGHT TIME FOR              IIIA5630
*COLLECTING STATISTICS ON GROUND PROCESSING TIME                          IIIA5640
*                                                                         IIIA5650
 147  ASSIGN       1+     K2            148                               IIIA5660
*                                                                         IIIA5670
* SEPARATION IN FLIGHT OF BOOSTER AND ORBITER                             IIIA5680
* COUNT THE NUMBER OF SUCESSFUL LAUNCHES                                  IIIA5690
*                                                                         IIIA5700
 148  SAVEX        30+    K1            149                               IIIA5710
 149  SPLIT                             150     180                       IIIA5720
*                                                                         IIIA5730
*-----------------------------                                            IIIA5740
*                                                                         IIIA5750
*                                                                         IIIA5760
*-----------------------------                                            IIIA5770
*BOOSTER COAST,REENTRY,FLYBACK AND LANDING                           IDST IIIA5780
*                                                                         IIIA5790
 150  HELP         K3     K64    K70    151            K76   X10          IIIA5800
 151  ASSIGN       1+     FN1           152                               IIIA5810
 152  ADVANCE                           153            1     FN1          IIIA5820
*                                                                         IIIA5830
*-----------------------------                                            IIIA5840
*                                                                         IIIA5850
*PARAMETER 8 COLLECTS THE TIME FROM LANDING UNTIL ENTRY INTO STORAGE       IIIA5860
 153  MARK         8                    154                               IIIA5870
*                                                                         IIIA5880
*                                                                         IIIA5890
*-----------------------------                                            IIIA5900
*                                                                         IIIA5910
*                                                                         IIIA5920
*-----------------------------                                            IIIA5930
*DELAY FOR TRANSPORTING FROM DOWNRANGE LANDING                            IIIA5940
*                                                                         IIIA5950
*                         MIN    NOM                   MAX                IIIA5960
 154  HELP         K3     K1000  K1500  155            K2000 X10          IIIA5970
 155  SAVEX        10     K0            156                               IIIA5980
 156  ADVANCE                           157            1     FN1          IIIA5990
*-----------------------------                                            IIIA6000
*                                                                         IIIA6010
```

```
*                                                                            IIIA6020
*----------------------------                                                IIIA6030
* WAITING LINE FOR BOOSTERS GOING INTO SAFING FACILITY                       IIIA6040
*                                                                            IIIA6050
 157  QUEUE        1                    158                                  IIIA6060
*                                                                            IIIA6070
*----------------------------                                                IIIA6080
*                                                                            IIIA6090
*                                                                            IIIA6100
*BOOSTER SAFING                                                         IDST IIIA6110
*                                                                            IIIA6120
 158  ENTER        1                    159                                  IIIA6130
*     TRIANGULAR       MIN   MODE                      MAX                   IIIC1911
 159  HELP        K3   K900  K1000      160            K1100 X10             IIIA6150
 160  ADVANCE                           161            1       FN1           IIIA6160
 161  LEAVE        1                    162                                  IIIA6170
*----------------------------                                                IIIA6180
*                                                                            IIIA6190
*                                                                            IIIA6200
*TRANSPORT BOOSTER TO MAINTENANCE                                       IDST IIIA6210
*                                                                            IIIA6220
*     TRIANGULAR       MIN   MODE                      MAX                   IIIC1912
 162  HELP        K3   K50   K100       163            K150  X10             IIIA6240
 163  ADVANCE                           164            1     FN1             IIIA6250
* ---------------------------                                                IIIA6260
*                                                                            IIIA6270
*                                                                            IIIA6280
*----------------------------                                                IIIA6290
*BOOSTER MAINTENANCE                                                    IDST IIIA6300
*                                                                            IIIA6310
 164  QUEUE        2                    165                                  IIIA6320
 165  ENTER        2                    166                                  IIIA6330
*     TRIANGULAR       MIN   MODE                      MAX                   IIIC1917
 166  HELP        K3   K2250 K2500      167            K2750 X10             IIIA6350
 167  ADVANCE                     .25   170     168    1     FN1             IIIA6360
*                                                                            IIIA6370
*BOOSTER ADDITIONAL MAINTENANCE                                         IDST IIIA6380
*                                                                            IIIA6390
*     EXPONENTIAL      MIN   MEAN                                            IIIC1913
 168  HELP        K8   K500  K1000      169                  X10             IIIA6410
 169  ADVANCE                           170            1     FN1             IIIA6420
 170  LEAVE        2                    171                                  IIIA6430
* ---------------------------                                                IIIA6440
*                                                                            IIIA6450
*                                                                            IIIA6460
*----------------------------                                                IIIA6470
*TRANSPORT BOOSTER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE           IDST IIIA6480
*                                                                            IIIA6490
 171  QUEUE        3                    172                                  IIIA6500
 172  ENTER        3                    173                                  IIIA6510
 173  HELP        K3   K1900 K2000      174            K2100 X10             IIIA6520
 174  ADVANCE                           175            1     FN1             IIIA6530
 175  LEAVE        3                    176                                  IIIA6540
*                                                                            IIIA6550
* BOOSTER STORAGE WHEN WAITING FOR ORBITER OR LAUNCH REQUIREMENT             IIIA6560
*                                                                            IIIA6570
```

```
       176  ENTER        4     K1                    177                                 IIIA6580
       177  HELP         K5    K1     C1             178                                 IIIA6590
       178  TABULATE     3                           179                                 IIIA6600
       179  TABULATE     1                           260                                 IIIA6610
      *-----------------------------                                                     IIIA6620
      *                                                                                  IIIA6630
      *                                                                                  IIIA6640
      * ORBITER BOOST INTO ORBIT,PERFORM MISSION AND RETURN                              IIIA6650
      *                                                                                  IIIA6660
       180  ENTER        14                          181                                 IIIA6670
       181  SAVEX        10    P3                    182                                 IIIA6680
       182  ASSIGN       1+    FN1                   183                                 IIIA6690
       183  ADVANCE                                  184         1      FN1              IIIA6700
       184  LEAVE        14                          185                                 IIIA6710
      *-----------------------------                                                     IIIA6720
      *                                                                                  IIIA6730
      *                                                                                  IIIA6740
      *PARAMETER 8 COLLECTS THE TIME FROM LANDING UNTIL ENTRY INTO STORAGE               IIIA6750
       185  MARK         8                           186                                 IIIA6760
      *                                                                                  IIIA6770
      *                                                                                  IIIA6780
      *                                                                                  IIIA6790
      *-----------------------------                                                     IIIA6800
      *ORBITER SAFING                                                               IDST IIIA6810
      *                                                                                  IIIA6820
       186  QUEUE        10                          187                                 IIIA6830
       187  ENTER        10                          188                                 IIIA6840
      *     TRIANGULAR         MIN    MODE                       MAX                     IIIC1914
       188  HELP         K3    K900   K1000          189         K1100  X10              IIIA6860
       189  ADVANCE                                  190         1      FN1              IIIA6870
       190  LEAVE        10                          191                                 IIIA6880
      *-----------------------------                                                     IIIA6890
      *                                                                                  IIIA6900
      *                                                                                  IIIA6910
      *-----------------------------                                                     IIIA6920
      *TRANSPORT ORBITER TO MAINTENANCE                                             IDST IIIA6930
      *                                                                                  IIIA6940
      *     LOGNORMAL          MEAN   STDDEV                                             IIIA6950
       191  HELP         K3    K50    K100           192         K150   X10              IIIA6960
       192  ADVANCE                                  193         1      FN1              IIIA6970
      *-----------------------------                                                     IIIA6980
      *                                                                                  IIIA6990
      *                                                                                  IIIA7000
      * ---------------------------                                                      IIIA7010
      *ORBITER MAINTENANCE                                                          IDST IIIA7020
      *                                                                                  IIIA7030
       193  QUEUE        11                          194                                 IIIA7040
       194  ENTER        11                          195                                 IIIA7050
      *     TRIANGULAR         MIN    MODE                       MAX                     IIIC1915
       195  HELP         K3    K2250  K2500          196         K2750  X10              IIIA7070
       196  ADVANCE                           .25    199   197   1      FN1              IIIA7080
      *                                                                                  IIIA7090
      *ORBITER ADDITIONAL MAINTENANCE                                               IDST IIIA7100
      *                                                                                  IIIA7110
      *     EXPONENTIAL        MIN    MEAN                                               IIIC1916
       197  HELP         K8    K500   K1000          198                X10              IIIA7130
```

```
000895    198  ADVANCE                                  199        1     FN1            IIIA7140
000896    199  LEAVE        11                          200                             IIIA7150
000897    * ----------------------------                                                IIIA7160
000898    *                                                                             IIIA7170
000899    *                                                                             IIIA7180
000900    *-----------------------------                                                IIIA7190
000901    *TRANSPORT ORBITER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE           IDST  IIIA7200
000902    * ORBITER VEHICLE SYSTEM TEST FACILITY                                        IIIA7210
000903    *                                                                             IIIA7220
000904    200  QUEUE        12                          201                             IIIA7230
000905    201  ENTER        12                          202                             IIIA7240
000906    202  HELP         K3     K1900 K2000          203        K2100 X10            IIIA7250
000907    203  ADVANCE                                  204        1     FN1            IIIA7260
000908    204  LEAVE        12                          205                             IIIA7270
000909    *                                                                             IIIA7280
000910    * ORBITER STORAGE WHEN WAITING FOR BOOSTER OR LAUNCH REQUIREMENT              IIIA7290
000911    *                                                                             IIIA7300
000912    205  ENTER        13     K1                   206                             IIIA7310
000913    206  HELP         K5     K2    C1             207                             IIIA7320
000914    207  TABULATE     21                          208                             IIIA7330
000915    208  TABULATE     23                          209                             IIIA7340
000916    209  TERMINATE                                                                IIIA7350
000917    *-----------------------------                                                IIIA7360
000918    *                                                                             IIIA7370
000919    *                                                                             IIIA7380
000920    *-----------------------------------------------------------------------------IIIC1920
000921    *       VEHICLE REMOVAL PATH                                                  IIIC1930
000922    *       ONLY VEHICLES WHICH ARE TO BE REMOVED FROM THE PAD FLOW               IIIC1940
000923    *       THROUGH THIS SECTION.                                                 IIIC1950
000924    *                                                                             IIIC1960
000925    *                                                                             IIIC1970
000926    * REMOVE BOOSTER, ORBITER AND LUT FROM THE PAD                          IDST  IIIC1980
000927    *                                                                             IIIC1990
000928    210  ENTER        25                          211                             IIIC2000
000929    *     TRIANGULAR         MIN   MODE                      MAX                  IIIC2010
000930    211  HELP         K3     K1800 K2000          212        K2200 X10            IIIC2020
000931    212  ADVANCE                                  213        1     FN1            IIIC2030
000932    213  LEAVE        25                          214                             IIIC2040
000933    214  RELEASE      *9                          215                             IIIC2050
000934    *-----------------------------                                                IIIC2060
000935    *                                                                             IIIC2070
000936    *                                                                             IIIC2080
000937    *-----------------------------                                                IIIC2090
000938    * TRANSPORT VEHICLE, SEPARARTE ORBITER, TRANSPORT ORBITER TO MAINT IDST       IIIC2100
000939    *                                                                             IIIC2110
000940    215  ENTER        30                          216                             IIIC2120
000941    *     TRIANGULAR         MIN   MODE                      MAX                  IIIC2130
000942    216  HELP         K3     K550  K750           217        K950  X10            IIIC2140
000943    217  ADVANCE                                  218        1     FN1            IIIC2150
000944    218  LEAVE        30                          219                             IIIC2160
000945    *                                                                             IIIC2170
000946    *-----------------------------                                                IIIC2180
000947    *                                                                             IIIC2190
000948    *                                                                             IIIC2200
000949    *-----------------------------                                                IIIC2210
000950    * SEPARATE ORBITER FROM BOOSTER/LUT AND SEND ORBITER TO MAINTENANCE           IIIC2220
```

```
000951    *                                                                 IIIC2230
000952     219  SPLIT                              220   237                IIIC2240
000953    *                                                                 IIIC2250
000954    *----------------------------                                     IIIC2260
000955    *                                                                 IIIC2270
000956    *                                                                 IIIC2280
000957    *----------------------------                                     IIIC2290
000958    * SEPARATE BOOSTER FROM LUT, TRANSPORT BOOSTER TO MAINTENANCE   IDST IIIC2300
000959     220  ENTER        19                    221                      IIIC2310
000960     221  HELP         K3    K550  K750      222        K950  X10     IIIC2320
000961     222  ADVANCE                            223        1     FN1     IIIC2330
000962     223  LEAVE        19                    224                      IIIC2340
000963     224  SPLIT                              225   130                IIIC2350
000964    *----------------------------                                     IIIC2360
000965    *BOOSTER MAINTENANCE ON RETURN FROM PAD                      IDST IIIA7400
000966     225  QUEUE        2                     226                      IIIA7410
000967     226  ENTER        2                     227                      IIIA7420
000968    *     EXPONENTIAL        MIN   MEAN                               IIIC2361
000969     227  HELP         K8    K500  K7500     228              X10     IIIA7440
000970     228  ADVANCE                            229        1     FN1     IIIA7450
000971     229  LEAVE        2                     230                      IIIA7460
000972    *----------------------------                                     IIIA7470
000973    *                                                                 IIIA7480
000974    *                                                                 IIIA7490
000975    *----------------------------                                     IIIA7500
000976    * BOOSTER VEHICLE SYSTEM  TEST FACILITY ON RETURN FROM PAD   IDST IIIA7510
000977     230  QUEUE        3                     231                      IIIA7520
000978     231  ENTER        3                     232                      IIIA7530
000979    *                        MIN   NOM                  MAX           IIIA7540
000980     232  HELP         K3    K1900 K2000     233        K2100 X10     IIIA7550
000981     233  ADVANCE                            234        1     FN1     IIIA7560
000982     234  LEAVE        3                     235                      IIIA7570
000983     235  ENTER        4     K1              236                      IIIA7580
000984     236  HELP         K5    K1    C1        260                      IIIA7590
000985    *----------------------------                                     IIIA7600
000986    *                                                                 IIIA7610
000987    *                                                                 IIIA7620
000988    *----------------------------                                     IIIA7630
000989    *ORBITER MAINTENANCE ON RETURN FROM PAD                      IDST IIIA7640
000990     237  QUEUE        11                    238                      IIIA7650
000991     238  ENTER        11                    239                      IIIA7660
000992    *     EXPONENTIAL        MIN   MEAN                               IIIC2362
000993     239  HELP         K8    K500  K7500     240              X10     IIIA7680
000994     240  ADVANCE                            241        1     FN1     IIIA7690
000995     241  LEAVE        11                    242                      IIIA7700
000996    * ----------------------------                                    IIIA7710
000997    *                                                                 IIIA7720
000998    *                                                                 IIIA7730
000999    * ----------------------------                                    IIIA7740
001000    * ORBITER VEHICLE SYS TEST FACILITY ON RETURN FROM PAD       IDST IIIA7750
001001     242  QUEUE        12                    243                      IIIA7760
001002     243  ENTER        12                    244                      IIIA7770
001003    *                        MIN   NOM                  MAX           IIIA7780
001004     244  HELP         K3    K1900 K2000     245        K2100 X10     IIIA7790
001005     245  ADVANCE                            246        1     FN1     IIIA7800
001006     246  LEAVE        12                    247                      IIIA7810
```

```
001007     247  ENTER       13                       248                     IIIA7820
001008     248  HELP        K5     K2     C1         249                     IIIA7830
001009     249  TERMINATE                                                    IIIA7840
001010    *--------------------------------------------------------------------IIIC2370
001011    *                                                                  IIIA7860
001012    *                                                                  IIIA7870
001013    * ----------------------------                                     IIIA7880
001014    *     COUNT NUMBER OF LAUNCHES MISSED BECAUSE OF UNAVAILABILITY    IIIA7890
001015    *        OF A BOOSTER AND/OR ORBITER AND/OR LUT                    IIIA7900
001016    *                                                                  IIIA7910
001017     250  SAVEX       5+     K1            BOTH 251    255             IIIA7920
001018     251  GATE        SE4                  BOTH 252    254             IIIA7930
001019     252  GATE        SE13                      253                    IIIA7940
001020    *                                                                  IIIA7950
001021    *NO BOOSTER OR ORBITER AVAILABLE                                   IIIA7960
001022    *                                                                  IIIA7970
001023     253  SAVEX       2+     K1                 260                    IIIA7980
001024    *                                                                  IIIA7990
001025    *NO BOOSTER AVAILABLE                                              IIIA8000
001026    *                                                                  IIIA8010
001027     254  SAVEX       3+     K1                 260                    IIIA8020
001028    *                                                                  IIIA8030
001029    *     EITHER NO ORBITER OR NO LUT OR BOTH                          IIIA8040
001030    *                                                                  IIIA8050
001031    *                                                                  IIIA8060
001032     255  SAVEX       7+     K1            BOTH 256    258             IIIA8070
001033     256  GATE        SE13                      257                    IIIA8080
001034    *                                                                  IIIA8090
001035    *     NO ORBITER AVAILABLE                                         IIIA8100
001036    *                                                                  IIIA8110
001037     257  SAVEX       4+     K1                 260                    IIIA8120
001038    *                                                                  IIIA8130
001039    *     NO LUT AVAILABLE WHEN HARDWARE IS AVAILABLE                  IIIA8140
001040    *                                                                  IIIA8150
001041     258  SAVEX       8+     K1                 260                    IIIA8160
001042    *                                                                  IIIA8170
001043    * COUNT NUMBER OF MISSED LAUNCHES  PAD(S) UNAVAILABLE              IIIA8180
001044     259  SAVEX       25+    K1                 260                    IIIA8190
001045    *                                                                  IIID  30
001046    *                                                                  IIID  40
001047    *----------------------------                                      IIID  50
001048    * INSERT HELP CARDS IN THIS BLOCK TO REQUEST CONFIDENCE INTERVALS.  IIID  60
001049    * THESE ARE THE FIVE DUMMY ADVANCE CARDS USED AS PLACE HOLDERS AND  IIID  70
001050    * ARE REPLACED BY HELP CARDS WHEN REQUESTING CONFIDENCE INTERVALS.  IIID  80
001051    *                                                                  IIID  90
001052     260  ADVANCE                               261                    IIID 100
001053     261  ADVANCE                               262                    IIID 110
001054     262  ADVANCE                               263                    IIID 120
001055     263  ADVANCE                               264                    IIID 130
001056     264  ADVANCE                               265                    IIID 140
001057    *                                                                  IIID 150
001058    *----------------------------                                      IIID 160
001059    *                                                                  IIID 170
001060     265  COMPARE     N276   LE     V10    BOTH 266    276             IIIA8200
001061     266  COMPARE     V10    E      N276        267                    IIIA8210
001062    *                                                                  IIIA8220
```

```
001063      * PRINT OUT SUMMARY TABLE                                                    IIIA8230
001064      *                                                                            IIIA8240
001065       267  HELP        K10                      268        TB10  ST14             IIIA8250
001066       268  HELP        K10    TB1    TB21       269        TB2   TB22             IIIA8260
001067       269  HELP        K10    QX1    QX10       270        QX2   QX11             IIIB1633
001068       270  HELP        K10    QX3    QX12       271        QX20  QX18             IIIB1636
001069       271  HELP        K10    QX21              272                               IIIA8280
001070       272  HELP        K10    SR1    SR10       273        SR2   SR11             IIIA8290
001071       273  HELP        K10    SR3    SR12       274                               IIIA8300
001072       274  HELP        K10    SR20   SR18       275                               IIIA8310
001073       275  HELP        K10    FR1    FR2        276        FR3                    IIIA8320
001074       276  TERMINATE   R                                                          IIIA8330
001075      *------------------------------------------------------------------------    IIIB1640
001076      *                                                                            IIIB1650
001077      *                              TABLE DEFINITIONS                             IIIB1660
001078      *                                                                            IIIB1670
001079      *  1 - TIME FROM LANDING TO STORAGE ENTRY (BOOSTER)                          IIIB1680
001080      *  2 - STORAGE TIME   (BOOSTER)                                              IIIB1690
001081      *  3 - GROUND PROCESSING TIME  (BOOSTER)                                     IIIB1700
001082      *  4 - FLOATING HOLD TIME BETWEEN COMPLETION OF ALL PRECOUNTDOWN             IIIB1710
001083      *      OPERATIONS AND THE INITIATION OF FINAL COUNTDOWN FOR THE FIRST        IIIC2380
001084      *      WINDOW                                                                IIIC2390
001085      *  5 - FLOSTING HOLD TIME BETWEEN COMPLETION OF ALL PRECOUNTDOWN             IIIC2400
001086      *      OPERATIONS AND THE INITIATION OF FINAL COUNTDOWN,FOR ALL WINDOWS      IIIC2410
001087      *      SUBSEQUENT TO THE FIRST                                               IIIC2420
001088      *  6 - THE AMOUNT OF TIME A SUCCESSFUL REQUEST SPENT WAITING FOR             IIIC2430
001089      *      VEHICLE ELEMENTS.                                                     IIIC2440
001090      *  7 - THE AMOUNT OF TIME LEFT TO THE FIRST LAUNCH WINDOW . NEGATIVE         IIIC2450
001091      *      VALUES IN THE TABLE INDICATE THE AMOUNT OF TIME PAST THE WINDOW       IIIC2460
001092      *      THE PROGRAM IS AT.                                                    IIIC2470
001093      *  8 - AMOUNT OF TIME AFTER PRIMARY WINDOW LAUNCH OCCURED AT                 IIIC2480
001094      *  9 - INDICATES THE MUMBER OF LAUNCHES AT EACH WINDOW                       IIIC2490
001095      * 10 - ELAPSED TIME OF VEHICLE ELEMENTS FROM THEIR STORAGE EXIT TO           IIIB1780
001096      *      THEIR LIFTOFF                                                         IIIB1790
001097      * 11 -                                                                       IIIB1800
001098      * 12 -                                                                       IIIB1810
001099      * 13 -                                                                       IIIB1820
001100      * 14 -                                                                       IIIB1830
001101      * 15 -                                                                       IIIB1840
001102      * 16 -                                                                       IIIB1850
001103      * 17 -                                                                       IIIB1860
001104      * 18 -                                                                       IIIB1870
001105      * 19 -                                                                       IIIB1880
001106      * 20 -                                                                       IIIB1890
001107      * 21 - TIME FROM LANDING TO STORAGE ENTRY (ORBITER)                          IIIB1900
001108      * 22 - STORAGE TIME (ORBITER)                                                IIIB1910
001109      * 23 - GROUND PROCESSING TIME (ORBITER)                                      IIIB1920
001110      *------------------                                                          IIIB1930
001111        1   TABLE       MP8    5000   500    20                                    IIIA8340
001112        2   TABLE       X10    000    500    30                                    IIIA8350
001113        3   TABLE       V1     9000   500    30                                    IIIA8360
001114        4   TABLE       X10    000    100    30                                    IIIA8370
001115        5   TABLE       X10    000    100    30                                    IIIC2500
001116        6   TABLE       M1     000    100    70                                    IIIC2510
001117        7   TABLE       X10    -5000  100    55                                    IIIC2520
001118        8   TABLE       X35    000    100    50                                    IIIC2530
```

```
001119       9    TABLE       X27   000   1     15                            IIIC2540
001120      10    TABLE       MP11  0000  500   30                            IIIA8380
001121     *11                                                                IIIC2550
001122     *12                                                                IIIC2560
001123     *13                                                                IIIC2570
001124     *14                                                                IIIC2580
001125     *15                                                                IIIC2590
001126     *16                                                                IIIC2600
001127     *17                                                                IIIC2610
001128     *18                                                                IIIC2620
001129     *19                                                                IIIC2630
001130     *20                                                                IIIC2640
001131      21    TABLE       MP8   5000  500   20                            IIIA8390
001132      22    TABLE       X10   000   500   30                            IIIA8400
001133      23    TABLE       V7    9000  500   30                            IIIA8410
001134     *-----------------------------------------------------------------IIIB1940
001135     *                                                                  IIIB1950
001136     *                   NUMBER OF SIMULATIONS TO BE PERFORMED          IIIB1960
001137     *                                                                  IIIB1970
001138     *      NOTE- SEE VARIABLE 10                                       IIIB1980
001139     *-----------------------------------------------------------------IIIB1990
001140 NEW        START       501
```

APPENDIX B. SUPPLEMENTAL FORTRAN INSTRUCTIONS DESCRIPTION AND LISTING

As the SOS Model evolved, it became apparent that additional Fortran subprograms were required to supplement the basic instructions provided by the standard GPSS II language. These subprograms, which are referred to as the user written routines, are utilized during the dynamic simulation of the Shuttle operations cycle by means of the GPSS II HELP block. The subprograms utilized in the SOS Model are listed after the descriptions in this Appendix. The function of the Fortran subprogram HELP is described below:

- HELP serves as the interfacing routine between the GPSS II instructions and the user written subprograms. The HELP subprogram is accessed by a GPSS II HELP block containing a control variable located in the X field. The HELP subprogram serves four general functions in support of the SOS Model: 1. generating random deviates from the available probability distribution; 2. collecting statistics on the time of stage storage; 3. transferring control to other special purpose subprograms; 4. providing scheduling data for each launch. A description of each segment of subprogram HELP is shown in Table B-I. Transfer to the segments from within the HELP subprogram is accomplished by utilizing the control variable in a Fortran computed GO TO statement and selecting the proper HELP segment.

The subprograms described below are responsible for the generation of the random numbers used in the user written subprograms.

- DRAND and DRAND1 serve as two independent random number generators. The random number generation is accomplished by modulo division. The random number seeds for these two subprograms are defined by variables ISEED and ISEED1, respectively. (No support routines called.)

The following subprograms are used in determining deviates from various statistical distributions to be used by the GPSS II instructions to accurately simulate the Shuttle operations cycle.

TABLE B-I. SUBPROGRAM HELP SEGMENT DESCRIPTION

Segment	Description
1	Generates the attributes of the next launch including mission type, time between launches, Orbiter mission length, and the time between the first and second launch opportunity. (Support routine called: MISSON.)
2	Calls a special purpose subprogram that prepares information obtained from GPSS II system variables for use in the Confidence Interval Option. (Support routine called: DUMMY.)
3	Generates the deviates from a triangular distribution. (Support routine called: TRIAG1.)
4	Generates the deviates from a normal distribution. (Support routine called: RNORM.)
5	Saves the clock time when a stage has completed processing and has entered storage. (No support routine called.)
6	Computes the elapsed time a stage has spent in storage based on a first in first out basis. (No support routines called.)
7	Prepares the model for a new case study by reading the input variables and the comment card title. (No support routines called.)
8	Generates the deviates for a two-parameter exponential distribution. (Support routine called: EXPO.)
9	Generates the deviates for a lognormal distribution. (Support routine called: RLOGN.)
10	Outputs the summary table. Repeated calls are required to complete the entire table. (Support routine called: PRINT.)
11	Generates the deviates for a constant distribution. (No support routines called.)
12	Generates the deviates for a uniform distribution. (Support routine called: UNFRM1.)
13	Generates the deviates for a beta distribution. (Support routines called: BETADP and PERTXF.)

- BETADP generates the deviates for a beta distribution. (Support routines called: GAM and PERTXF.)
- DICRET produces deviates from a discrete probability distribution. This subprogram permits the model to generate random values from data that has either a 1/3 or 1/2 probability of selection. (Support routine called: DRAND.)
- EXPO generates the deviates for a two parameter exponential distribution. (Support routine called: DRAND1.)
- GAM generates the deviates for a gamma distribution. (Support routine called: DRAND1.)
- PERTXF converts three input parameters, namely the minimum expected time, the most likely expected time and the maximum expected time to the mean and standard deviation of a beta distribution. (No support routines called.)
- RLOGN calculates the deviates for a lognormal distribution by exponentiating the normal deviate. The mean and standard deviation of the lognormal distribution are obtained from the calling subprogram. (Support routine called: RNORM.)
- RNORM calculates the deviates for a normal distribution. The mean and standard deviation of the routine are obtained from the calling subprogram. (Support routine called: DRAND1.)
- TRIAG and TRIAG1 produce the random deviates for a triangular distribution function. These two routines are identical and independent. (Support routines called: DRAND and DRAND1, respectively.)
- UNFRM and UNFRM1 produce random deviates from a uniform distribution. The two routines are identical and independent. (Support routines called: DRAND and DRAND1, respectively.)

It should be noted that subprograms DRAND, TRIAG, and UNFRM are used exclusively by the MISSON subprogram to provide a separate and unique sequence of random numbers.

The subprogram described below is responsible for scheduling the Shuttle launches.

- MISSON is responsible for determining the parameters associated with each Shuttle launch. MISSON is used to determine a realistic time between launches for the Shuttle system. For each launch, the mission type is defined with the possible types including Shuttle payload placement, space station logistics, tug placement and fuel supplies, Department of Defense, and others, including space rescue. The MISSON subprogram also determines the Orbiter mission length which depends on mission type, the date of the next launch, and the time between the first and second launch opportunity. (Support routines called: DICRET, TRIAG, UNFRM.)

The following subprograms are not active during the dynamic simulation of the Shuttle operations cycle, but were developed for the convenience of the user. These subprograms are used to produce the TABLE1 output which summarizes the activity time distributions defined in the System Model Description of the SOS Model.

- OUTPUT determines the type of distribution for each Shuttle processing event and outputs this information in tabular form. (Support routine called: REALN).
- REALN converts alphanumeric data into the real data used by the OUTPUT routine. (No support routines called.)
- TABLE1 scans the System Model Description for the titles of the various processing operations and for their associated time distributional parameters. (Support routine called: OUTPUT.)

The following subprograms are used when confidence intervals are requested for specific statistics generated during the dynamic simulation of the Shuttle operations cycle. Reference 3 gives detailed information on the Confidence Interval Option.

- DUMMY controls the sampling rate used in obtaining statistical observations for those statistics specified by the Confidence Interval Option. The subprogram also assures the proper collection of the statistics specified by the user. (Support routine called: SAMPLE.)
- SAMPLE makes statistical observations during the simulation on the statistics specified by the user to have confidence intervals calculated for them. At the conclusion of the simulation, a summary table is outputted that contains a descriptive heading and a corresponding confidence interval for each of the statistics specified by the user to be collected. (Support routine called: STDV.)

- STAT is used to read and store descriptive headings to be used in the output of confidence interval statistics. Each card of the System Model Description is scanned for the appropriate headings. This subprogram is not active during the actual simulation. (No support routines called.)

- STDV calculates the sample standard deviation for the statistics specified for confidence interval calculations. (No support routines called.)

The subprogram described below is responsible for a summary output of the statistics collected during the simulation of the Shuttle operations cycle.

- PRINT is a collection of various write statements which print out a summary table of various simulation parameters at the end of each computer run. The values of these parameters are transmitted to the routine through two different transfer methods. The first method transfers the values of certain GPSS II system variables (table means, facility utilization and queue times) to PRINT as arguments of the subroutine. These arguments are defined on the HELP blocks at the end of the model. The other method permits the user to retrieve the output parameters through the values stored in the GPSS II common block variables. These tables are subscripted arrays and are defined in the GPSS II reference manual. The routine is executed by a series of call statements that subsequently execute each statement of the subprogram. (Support routine called: SAMPLE.)

The GPSS II Mainline (GPSS2) routine is the only GPSS II symbolic element deck available for modification by the SOS Model user. It serves as the mainline master control for the complete GPSS II program. The GPSS2 routine is a software package available upon request from the operators of the UNIVAC 1108 EXEC VIII System. The routine is described as follows:

- GPSS2 contains the option to specify the size of the network and its associated variables by means of the Fortran parameter statements contained within the routine. The total program core size is determined by the limits of the model description as defined by the values assigned to the Fortran parameter statements. Complete instructions and other information on making changes is contained in the comments of the routine listing. (Major GPSS II routines USES, BLOCKD, SETERR, INPROC, EXECUT, PUTOFF, ASSEMB; other support routines called: RCONWN, CSFREQ, LETTER, STAT, and TABLE1.)

BETADP DATE 063072 PAGE 1

```
420223*TPF$.BETADP
      FUNCTION BETADP(J,SMEAN,SVAR)                                      BETA  10
      COMMON /HEPL/ PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)       BETA  20
      A = SMEAN                                                          BETA  30
      B = SVAR                                                           BETA  40
      X = GAM(A)                                                         BETA  50
      BETADP=X/(X+GAM(B) )                                               BETA  60
      BETADP=BETADP*(PARAM(J,3)-PARAM(J,2))+PARAM(J,2)                   BETA  70
      RETURN                                                             BETA  80
      END                                                                BETA  90

@HDG,P      DICRET

@PRT,S      DICRET
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.DICRET
      FUNCTION DICRET(JP)
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      RNUM=DRAND(ISEED)
      IF(PARAM(JP,2).EQ.0.)GO TO 20
      IF(RNUM.GE.0.6666)GO TO 15
      IF(RNUM.GE.0.3333)GO TO 10
      DICRET=PARAM(JP,1)
      RETURN
   10 DICRET=PARAM(JP,2)
      RETURN
   15 DICRET=PARAM(JP,3)
      RETURN
   20 IF(RNUM.LE.0.5)GO TO 30
      DICRET=PARAM(JP,3)
      RETURN
   30 DICRET=PARAM(JP,1)
      RETURN
      END

@HDG,P     DRAND

@PRT,S     DRAND
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.DRAND
            FUNCTION DRAND(NR)                                GPST2500
            NR=NR*316231                                      GPST2510
            IF(NR)5,5,6                                       GPST2520
          5 NR=NR+34359738367                                 GPST2530
          6 RN=NR                                             GPST2540
            DRAND=RN/343597384.E2                             GPST2550
            RETURN                                            GPST2560
            END                                               GPST2570

@HDG,P      DRAND1

@PRT,S    DRAND1
FURPUR HA18-06/30-13:31
```

```
DRANDI                                                       DATE 063072          PAGE    1

420223*TPF$.DRANDI
              FUNCTION DRANDI(NRI)
              NRI=NRI*316231
              IF(NRI)5,5,6
            5 NRI=NRI+34359738367
            6 RN=NRI
              DRANDI=RN/343597384.E2
              RETURN
              END

@HDG,P       DUMMY

@PRT,S     DUMMY
FURPUR HA18-06/30-13:31
```

420223*TPFS.DUMMY

```
      SUBROUTINE DUMMY(K1,K2,K3,K4,K5)                                  DUMY  10
      COMMON/STAT2/NFLTS                                                DUMY  20
      COMMON/STAT3/ISN                                                  DUMY  30
      COMMON/EKSES/JEKS(1)                                              DUMY  40
      COMMON/NODE4/JN4(1)                                               DUMY  50
C                                                                       DUMY  60
C     NENTR DETERMINES THE NUMBER OF TERMINATED SIMULATIONS             DUMY  70
      NENTR=MGTH2U(JN4(276))+2                                          DUMY  80
C                                                                       DUMY  90
C     DETERMINE IF 100 SIMULATIONS HAVE BEEN COMPLETED                  DUMY 100
C     IF NOT, RETURN                                                    DUMY 110
      IF(NENTR.LT.100) RETURN                                           DUMY 120
C                                                                       DUMY 130
C     DETERMINE IF 100 OR MORE FLIGHTS (IN INCREMENTS OF 50 FLIGHTS)    DUMY 140
C     HAVE OCCURRED                                                     DUMY 150
C     IF NOT, RETURN                                                    DUMY 160
      IF (MOD(NENTR,50).NE.0) RETURN                                    DUMY 170
C                                                                       DUMY 180
C                                                                       DUMY 190
C                                                                       DUMY 200
C     NFLTS DETERMINES THE NUMBER OF SIMULATIONS THAT HAVE BEEN         DUMY 210
C     COMPLETED                                                         DUMY 220
      NFLTS=JEKS(2)+JEKS(3)+JEKS(4)+JEKS(8)+JEKS(22)+JEKS(23)+JEKS(24)  DUMY 230
     1 +JEKS(25)+JEKS(30)                                               DUMY 240
C                                                                       DUMY 250
C                                                                       DUMY 260
C     DETERMINE IF THE STATISTICAL IDENTIFICATION NUMBER IS LEGAL       DUMY 270
C     IF NOT, RETURN                                                    DUMY 280
      IF(K2.LT.0.OR.K2.GT.5) RETURN                                     DUMY 290
C                                                                       DUMY 300
C     IF OPTION 1 HAS BEEN SPECIFIED, IDENTIFY THE MAXIMUM NUMBER       DUMY 310
C     OF STATISTICS TO BE COLLECTED                                     DUMY 320
      IF(K2.EQ.0) ISN=9                                                 DUMY 330
C                                                                       DUMY 340
C     DETERMINE THE CONFIDENCE INTERVAL IDENTIFICATION NUMBER           DUMY 350
      IF(K2-2)2,1,1                                                     DUMY 360
C                                                                       DUMY 370
C                                                                       DUMY 380
C     IF OPTION 1 HAS BEEN REQUESTED, COLLECT STATISTICS FOR THE OPERATIDUMY 390
C     EVALUATION PARAMETERS                                             DUMY 400
    2 K2T=K2                                                            DUMY 410
      K3T=K3                                                            DUMY 420
      K4T=K4                                                            DUMY 430
      K5T=K5                                                            DUMY 440
      K2=1                                                              DUMY 450
      K3=1                                                              DUMY 460
      K4=JEKS(30)                                                       DUMY 470
      K5=1                                                              DUMY 480
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 490
C                                                                       DUMY 500
      K2=2                                                              DUMY 510
      K4=NFLTS-JEKS(30)                                                 DUMY 520
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 530
C                                                                       DUMY 540
      K2=3                                                              DUMY 550
```

```
      K4=JEKS(26)                                                       DUMY 560
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 570
C                                                                       DUMY 580
      K2=4                                                              DUMY 590
      K4=JEKS(3)                                                        DUMY 600
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 610
C                                                                       DUMY 620
      K2=5                                                              DUMY 630
      K4=JEKS(4)                                                        DUMY 640
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 650
C                                                                       DUMY 660
      K2=6                                                              DUMY 670
      K4=JEKS(2)                                                        DUMY 680
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 690
C                                                                       DUMY 700
      K2=7                                                              DUMY 710
      K4=JEKS(8)                                                        DUMY 720
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 730
C                                                                       DUMY 740
      K2=8                                                              DUMY 750
      K4=JEKS(25)                                                       DUMY 760
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 770
C                                                                       DUMY 780
      K2=9                                                              DUMY 790
      K4=JEKS(22)+JEKS(23)+JEKS(24)                                     DUMY 800
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 810
C                                                                       DUMY 820
C     IF OPTION 1 HAS BEEN REQUESTED, RETURN                            DUMY 830
      IF(K2T.EQ.0) RETURN                                               DUMY 840
C                                                                       DUMY 850
C     IF OPTION 2 HAS BEEN REQUESTED, COLLECT STATISTICS FOR REQUEST 1  DUMY 860
      K2=10                                                             DUMY 870
      K3=K3T                                                            DUMY 880
      K4=K4T                                                            DUMY 890
      K5=K5T                                                            DUMY 900
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 910
      RETURN                                                            DUMY 920
C                                                                       DUMY 930
C     IF OPTION 2 HAS BEEN REQUESTED, COLLECT STATISTICS FOR THE        DUMY 940
C     REMAINING REQUESTS                                                DUMY 950
    1 K2=K2+9                                                           DUMY 960
      CALL SAMPLE(K1,K2,K3,K4,K5)                                       DUMY 970
      RETURN                                                            DUMY 980
      END                                                               DUMY 990
```

@HDG,P EXPO

@PRT,S EXPO

FURPUR HA18-06/30-13:31

```
420223*TPFS.EXPO
      FUNCTION EXPO(J)
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      RNUM=DRAND1(ISEED1)
      EXPO=PARAM(J,1)-PARAM(J,2)*ALOG(RNUM)
      RETURN
      END

@HDG,P     GAM

@PRT,S     GAM
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.GAM
      FUNCTION GAM(ALPHA)
      K = ALPHA
      FK = K
      GAM = 0.
      IF(K)7,7,8
    8 IF(K.GT.20) GO TO 30
      PROD = 1.0
      DO 10 I=1,K
   10 PROD = PROD*DRAND1(ISEED1)
      GAM = -ALOG(PROD)
    7 D = ALPHA-FK
      IF(D-.015)12,12,11
   11 IF(D-.985)13,20,20
   20 W = 1.0
      GO TO 22
   13 A = 1.0/D
      B = 1.0/(1.-D)
   14 X = DRAND1(ISEED1)**A
      Y = DRAND1(ISEED1)**B+X
      IF(Y-1.)15,15,14
   15 W = X/Y
   22 Y = -ALOG(DRAND1(ISEED1))
      GAM =GAM+W*Y
   12 RETURN
   30 GAM = ALPHA
      RETURN
      END

@HDG,P     GPSS2

@PRT,S    GPSS2
FURPUR HA18-06/30-13:31
```

420223*1PF$.GPSS2

```
C       GPSS II    VERSION F     LEVEL 3
C
C     THE GENERAL PURPOSE SYSTEMS SIMULATOR II FINAL USER MANUAL IS
C     UP-4129, AND THE FINAL GPSS II CODING FORM IS UP-4131, BOTH
C     OBTAINABLE FROM HOLYOKE, MASS., VIA A SALES HELP REQUISITION
C     THROUGH THE LOCAL UNIVAC MANAGER.
C
C*****THE FOLLOWING NOTES ARE INTENDED TO SUPPLEMENT OR
C     UPDATE INFORMATION PRESENTED IN THE GPSS USER MANUAL.*****
C
C         1. EACH TIME AN OVERLAY OCCURS A WARNING TO THIS EFFECT
C            IS PRINTED, BUT EXECUTION IS NOT INHIBITED
C         2. IF JOBTAPE AND WRITE ARE USED THE NUMBER OF TRANS-
C            ACTIONS ON TAPE MUST BE AT LEAST ONE GREATER THAN
C            THE NUMBER REQUIRED FOR THE EXECUTION OF THE JOB
C         3. EXECUTION USING XQT MAPGPS RESULTS IN A SEGMENTATION
C            OF GPSS II.
C         4. THE STANDARD UPPER LIMITS OF FIELDS GIVEN IN APPENDIX 3
C            OF THE GPSS USER MANUAL MAY BE CHANGED. THE METHOD FOR
C            MAKING SUCH CHANGES HAS BEEN REVISED. SEE NOTE 7 FOR AN
C            EXPLANATION.
C         5. THE EXECUTIVE COMMAND, @SETC 2 , WILL CAUSE AN ASSEMBLED
C            DECK TO BE PUNCHED OUT.
C         6. UPON DETECTION OF AN END COMMAND, ALL JOBTAPES WHICH WERE
C            REFERENCED BY A WRITE BLOCK WILL HAVE AN END-OF-FILE PLACED
C            ON THEM. REWIND OF THESE TAPES WILL NOT OCCUR.
C        ***************************************************
C        *7. THE METHOD USED TO CHANGE TABLE SIZES FOR THIS *
C        *   VERSION IS TO CHANGE THE VALUES GIVEN IN THE   *
C        *   FORTRAN PARAMETER STATEMENTS WHICH FOLLOW. THE *
C        *   TABLE TYPE REFERRED TO BY  A  FORTRAN PARAMETER*
C        *   STATEMENT  IS GIVEN IN THE LINE PRECEDING THAT *
C        *   STATEMENT.                                     *
C        ***************************************************
C
C**BLOCKS
      PARAMETER NBMAX = 400
C
C**FACILITIES
      PARAMETER NFMAX = 15
C
C**STORAGES
      PARAMETER NSMAX = 50
C
C**QUEUES
      PARAMETER NQMAX = 50
C
C**USER CHAINS                                                   USCH
      PARAMETER NUCMAX = 1
C                                                                USCH
C**LOGIC SWITCHES
      PARAMETER NLSMAX = 25
C
C**SAVEX LOCATIONS
      PARAMETER NSLMAX = 50
```

```
C
C**FUNCTIONS
      PARAMETER NFNMAX = 10
C
C**TABLES AND QTABLES (COMBINED TOTAL)
      PARAMETER NTQMAX = 30
C
C**VARIABLE STATEMENTS
      PARAMETER NVMAX = 30
C
C**COMMON CORE AREA
      PARAMETER NCCMAX = 4500
C
C**TRANSACTIONS (MAXIMUM ALLOWED IN SYSTEM AT ANY GIVEN TIME)
      PARAMETER NTMAX = 100
C
C**PARAMETERS PER TRANSACTION (MUST BE LESS THAN OR EQUAL TO 30)
      PARAMETER  NPRMAX=20
C*********THE FOLLOWING CARDS ARE NOT TO BE DELETED, NOR CHANGED
C            UNLESS SO NOTED IN COMMENT CARDS.*********
      PARAMETER  NPRM=((NPRMAX+1)/2)*2 ,
     A          NPX2 = NPRM/2-NPRM/4-NPRM/6-NPRM/10+NPRM/12
     B                 -NPRM/14+NPRM/20-NPRM/22-NPRM/26
     C                 +NPRM/28+NPRM/30 ,
     D          NPX4 = NPRM/4-NPRM/8-NPRM/12-NPRM/20+NPRM/24-NPRM/28 ,
     E          NPX6 = NPRM/6-NPRM/12-NPRM/18-NPRM/30 ,
     F          NPX8 = NPRM/8-NPRM/16-NPRM/24 ,
     G          NPX10= NPRM/10-NPRM/20-NPRM/30 ,
     H          NP2  = NPX2*NTMAX +1 -NPX2 ,
     I          NP4  = NPX4*NTMAX +1 -NPX4 ,
     J          NP6  = NPX6*NTMAX +1 -NPX6
      PARAMETER NP8  =NPX8*NTMAX +1 -NPX8 ,
     1          NP10 = NPX10*NTMAX +1 -NPX10 ,
     2          NP12 = (NPRM/12-NPRM/24)*NTMAX +1 -(NPRM/12-NPRM/24),
     3          NP14 = (NPRM/14-NPRM/28)*NTMAX +1 -(NPRM/14-NPRM/28),
     4          NP16 = (NPRM/16)*NTMAX +1 -NPRM/16 ,
     5          NP18 = (NPRM/18)*NTMAX +1 - NPRM/18 ,
     6          NP20 = (NPRM/20)*NTMAX +1 - NPRM/20 ,
     7          NP22 = (NPRM/22)*NTMAX +1 - NPRM/22 ,
     8          NP24 = (NPRM/24)*NTMAX +1 - NPRM/24 ,
     9          NP26 = (NPRM/26)*NTMAX +1 - NPRM/26 ,
     A          NP28 = (NPRM/28)*NTMAX +1 - NPRM/28 ,
     B          NP30 = (NPRM/30)*NTMAX +1 - NPRM/30
      COMMON/NODE1/JN1(NBMAX)
      COMMON/NODE2/JN2(NBMAX)
      COMMON/NODE3/JN3(NBMAX)
      COMMON/NODE4/JN4(NBMAX)
      COMMON/NODE5/JN5(NBMAX)
      COMMON/EQ1/JF1(NFMAX)
      COMMON/EQ2/JF2(NFMAX)
      COMMON/EQ3/JF3(NFMAX)
      COMMON/EQ4/JF4(NFMAX)
      COMMON/EQ5/JF5(NFMAX)
      COMMON/EQ6/JF6(NFMAX)
      COMMON/EQ7/JF7(NFMAX)
      COMMON/STOR1/JS1(NSMAX)
```

```
      COMMON/STOR2/JS2(NSMAX)
      COMMON/STOR3/JS3(NSMAX)
      COMMON/STOR4/JS4(NSMAX)
      COMMON/STOR5/JS5(NSMAX)
      COMMON/STOR6/JS6(NSMAX)
      COMMON/STOR7/JS7(NSMAX)
      COMMON/QUE1/JQ1(NQMAX)
      COMMON/QUE2/JQ2(NQMAX)
      COMMON/QUE3/JQ3(NQMAX)
      COMMON/QUE4/JQ4(NQMAX)
      COMMON/QUE5/JQ5(NQMAX)
      COMMON/QUE6/JQ6(NQMAX)
      COMMON/LOGIX/JL1(NLSMAX)
      COMMON/EKSES/JEKS(NSLMAX)
      COMMON/FN1/JYLOCS(NFNMAX)
      COMMON/FN2/JXLOCS(NFNMAX)
      COMMON/FN3/JSLOCS(NFNMAX)
      COMMON/TAB1/JTLOCS(NTQMAX)
      COMMON/TAB2/JTMODE(NTQMAX)
      COMMON/TAB3/JLOWRS(NTQMAX)
      COMMON/TAB4/JTINCS(NTQMAX)
      COMMON/TAB5/JTLAST(NTQMAX)
      COMMON/TAB6/JTLNUM(NTQMAX)
      COMMON/TAB7/TLARG(NTQMAX)
      COMMON/TAB8/TSQR(NTQMAX)
      COMMON/TAB9/TWARG(NTQMAX)
      COMMON/TAB10/TWSQR(NTQMAX)
      COMMON/VARS/JVLOCS(NVMAX)
      COMMON/WORDS/JWORDS(NCCMAX)
      COMMON/UC1/JU1(NUCMAX)                                    USCH
      COMMON/UC2/JU2(NUCMAX)                                    USCH
      COMMON/UC3/JU3(NUCMAX)                                    USCH
      COMMON/UC4/JU4(NUCMAX)                                    USCH
      COMMON/UC5/JU5(NUCMAX)                                    USCH
      COMMON/TRAN00/JTSTAT(NTMAX)
      COMMON/TRAN1/JNDT(NTMAX)
      COMMON/TRAN2/JCHAIN(NTMAX)
      COMMON/TRAN3/JMOVE(NTMAX)
      COMMON/TRAN4/JNNWD(NTMAX)
      COMMON/TRAN5/JC1(NTMAX)
      COMMON/TRAN6/JC2(NP2)
      COMMON/TRAN7/JC3(NP4)
      COMMON/TRAN8/JC4(NP6)
      COMMON/TRAN9/JC5(NP8)
      COMMON/TRAN10/JC6(NP10)
      COMMON/TRAN12/JC7(NP12)
      COMMON/TRAN14/JC8(NP14)
      COMMON/TRAN16/JC9(NP16)
      COMMON/TRAN18/JC10(NP18)
      COMMON/TRAN20/JC11(NP20)
      COMMON/TRAN22/JC12(NP22)
      COMMON/TRAN24/JC13(NP24)
      COMMON/TRAN26/JC14(NP26)
      COMMON/TRAN28/JC15(NP28)
      COMMON/TRAN30/JC16(NP30)
      COMMON K(100)
```

```
      COMMON LPRI(128),LPRIOR(128)
      COMMON ICHAR(70),KTYPE(43),KGATE(12),KCONTR(7),KSV(50),
     1 KCOMP(6),KSELEC(7),LX(6)
      COMMON/STAT3/ISN
      COMMON/STATS/NHEAD
      DIMENSION NHEAD(14,11)
      DIMENSION FWORDS(1)
      EQUIVALENCE(JWORDS(1),FWORDS(1))
      EQUIVALENCE(K(1),KASYM1),(K(2),KASYM2),(K(3),KNODES),(K(4),KEQS),
     1 (K(5),KSTORS),(K(6),KQUES),(K(7),KVARS),(K(8),KLOGIX),
     2 (K(9),KEKSES),(K(10),KFNS),(K(11),KTABS),(K(12),KWORDS),
     3 (K(13),KTRANS),(K(14),KRAND),(K(15),KASMBL),(K(16),KIT),
     4 (K(17),KOT),(K(86),KUSERS)                                   USCH
      EQUIVALENCE(K(79),KPARAM),(K(71),INOFLO),(K(72),INDEND),
     1 (K(55),IFATAL)
C
C*****THE FOLLOAING NOTES PROVIDES TO THE USER OF THE MSFC-SOS MODEL
C     THE NECESSARY INFORMATION CONCERNING @SETC COMMAND CONTROL OPTIONS.
C   1. A @SETC 3 COMMAND PLACED BEFORE THE @XQT CONTROL CARD WILL
C        ELIMINATE THE PRINTOUT OF THE MODEL LISTING. ONLY SUMMARY
C        RESULTS AND STANDARD GPSS OUTPUT WILL BE PRINTED.
C   2. A @SETC 4 COMMAND PLACED BEFORE THE @XQT CONTROL CARD WILL
C        GENERATE A TABLE CONTAINING THE DISTRIBUTION TIME PARAMETERS
C        FOR THE DIFFERENT PROCESSING OPERATIONS.
C        NOTE: THAT THE MODEL MUST BE ADDED TO THE RUN STREAM IN ORDER
C        TO PRODUCE THIS OUTPUT TABLE.
C        EXAMPLE DECK SET-UP:      @SETC 4
C                                  @XQT GPSSE8.MAPGPS
C                                  @ADD MODEL.
C                                  @ADD MODEL.
C                                  COMMENT CARD FOR CASE 1
C                                  DATA CARD 1
C                                           END
C   3. A @SETC 5 COMMAND PLACED BEFORE THE @XQT CONTROL CARD WILL
C        DETERMINE THE HEADINGS TO BE USED IN THE CONFIDENCE INTERVAL
C        STATISTICS COLLECTED BY THE MODEL. THIS COMMAND IS USED ONLY
C        WHEN CONFIDENCE INTERVAL STATISTICS ARE DESIRED.
C        NOTE: TO COLLECT THE HEADINGS TO BE ASSOCIATED WITH THE
C        STATISTICS COLLECTED, THE MODEL MUST BE ADDED TO THE RUN STREAM.
C        EXAMPLE DECK SET-UP:      @SETC 5
C                                  @XQT GPSSE8.MAPGPS
C                                  @ADD MODEL.
C                                  @ADD MODEL.
C                                  COMMENT CARD FOR CASE 1
C                                  DATA CARD 1
C                                           END
C
      DIMENSION ICTLCD(4),JCTLCD(4),KCTLCD(4)
      DATA ICTLCD/'@ASG,T DUMMY,F          . '/,
     1      JCTLCD/'@BRKPT PRINT$/DUMMY    . '/,
     2      KCTLCD/'@BRKPT PRINT$          . '/
      CALL RCOMND(MPRINT)
      NPRINT=MGETT2(MPRINT)
      IF(NPRINT.EQ.4)CALL TABLE1
      IF(NPRINT.EQ.5) CALL STAT
      IF(NPRINT.NE.3)GO TO 5
```

```
          CALL CSFREQ(ICTLCD)
          CALL CSFREQ(JCTLCD)
        5 CALL USES
          KPARAM = NPRM
          KIT = 5
          KOT = 6
          KRAND = 1220703125
          KNODES = NBMAX
          KEQS   = NFMAX
          KSTORS = NSMAX
          KQUES  = NQMAX
          KVARS  = NVMAX
          KLOGIX = NLSMAX
          KEKSES = NSLMAX
          KFNS   = NFNMAX
          KTABS  = NTQMAX
          KWORDS = NCCMAX
          KTRANS = NTMAX
          KUSERS = NUCMAX                                        USCH
              CALL BLOCKD
              CALL SETERR
       10     CALL INPROC($20,$30)
       20 IF (INDFLO .NE. 0) CALL FLOW
C         TO PERMIT EXECUTION OF A JOB WITH ILLEGAL INPUT CARDS, OR TO
C         PERMIT EXECUTION FOLLOWING A RESET OR CLEAR CARD WHERE A GPSS II
C         ERROR HAS ALREADY OCCURRED, REMOVE THE FOLLOWING CARD
          IF (IFATAL .NE. 0) GO TO 10
          IF (INDEND .NE. 0) GO TO 10
          IF(NPRINT.NE.3)GO TO 25
          CALL CSFREQ(KCTLCD)
          CALL LETTER
       25     CALL EXECUT
              CALL PUTOUT
              GO TO 10
       30     CALL ASSEMB
              GO TO 10
          END
```

```
@HDG,P     HELP

@PRT,S     HELP
FURPUR HA18-06/30-13:31
```

420223*TPFS.HELP

```
      SUBROUTINE HELP(K1,K2,K3,K4,K5)                                  HELP  10
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)       HELP  20
      COMMON /HLPPRT/NBOOS,NORBI,NLUTS,TITLE(20),NSEED,NSEED1
      COMMON/ISTAT/NCONF
   25 GO TO  (1,2,3,4,5,6,7,8,9,10,11,12,13),K1
C                                                                      HELP 110
C*****GENERATE REQUIREMENTS FOR NEXT LAUNCH                            HELP 120
C                                                                      HELP 130
    1 CONTINUE                                                         HELP 140
      CALL      MISSON(NP1,NP2,NP3,NP4,NP5,MODEL)
      K2=NP1                                                           HELP 160
      K3=NP2                                                           HELP 170
      K4=NP3                                                           HELP 180
      K5=NP5
      RETURN                                                           HELP 200
    2 CONTINUE                                                         HELP 210
C
C*****COLLECT STATISTICAL DATA FOR CONFIDENCE INTERVALS
C
      CALL DUMMY(K1,K2,K3,K4,K5)
      RETURN                                                           HELP 230
C                                                                      HELP 240
C*****GENERATE DEVIATE FOR TRIANGULAR DISTRIBUTION                     HELP 250
C                                                                      HELP 260
    3 PARAM(1,1)=K2                                                    HELP 270
      PARAM(1,2)=K3                                                    HELP 280
      PARAM(1,3)=K4                                                    HELP 290
      K5=TRIAG1(1)                                                     HELP 300
      RETURN                                                           HELP 310
C                                                                      HELP 320
C*****GENERATE DEVIATE FOR NORMAL DISTRIBUTION                         HELP 330
C                                                                      HELP 340
    4 PARAM(1,1)=K2                                                    HELP 350
      PARAM(1,2)=K3                                                    HELP 360
      K4=RNORM(1)                                                      HELP 370
      K5=K4                                                            HELP 380
      RETURN                                                           HELP 390
C                                                                      HELP 400
C*****STORE THE TIME THAT THE BOOSTER (K2=1) OR ORBITER (K2=2) WAS     HELP 410
C*****PUT IN STORAGE.                                                  HELP 420
C                                                                      HELP 430
    5 IS(K2)=IS(K2)+1                                                  HELP 440
      K=IS(K2)                                                         HELP 450
      ISTORE(K2,K)=K3                                                  HELP 460
      RETURN                                                           HELP 470
C                                                                      HELP 480
C*****COMPUTE THE TIME THAT THE BOOSTER OR ORBITER WAS PUT IN STORAGE  HELP 490
C*****BASED ON FIRST-IN-FIRST-OUT.                                     HELP 500
    6 K4=K3-ISTORE(K2,1)                                               HELP 510
      IS(K2)=IS(K2)-1                                                  HELP 520
      K=IS(K2)                                                         HELP 530
      IF(K.EQ.0)RETURN                                                 HELP 540
      DO 20 I=1,K                                                      HELP 550
   20 ISTORE(K2,I)=ISTORE(K2,I+1)                                      HELP 560
      RETURN                                                           HELP 570
```

```
      7 CONTINUE                                                        HELP 580
C                                                                       HELP 590
C*****INITIALIZE MISSON SUBROUTINE AND PREPARE FOR  NEW CASE            HELP 600
C                                                                       HELP 610
      LL=0                                                              HELP 620
C                                                                       HELP 650
C*****READ CASE STUDY COMMENT CARD
C                                                                       HELP 670
      READ(5,101)(TITLE(I),I=1,20)                                      HELP 700
  101 FORMAT(20A4)                                                      HELP 710
C                                                                       HELP 720
C*****READ THE VERSION CODE OF THE TRAFFIC MODEL TO BE SIMULATED
C*****READ IN THE NUMBER OF BOOSTERS, ORBITERS & LUTS
C*****READ IN THE LEVEL OF SIGNIFICANCE
C*****READ IN THE RANDOM NUMBER SEED FOR MISSION MODEL PROGRAM(ISEED)
C*****READ IN THE RANDOM NUMBER SEED FOR DISTRIBUTION GENERATORS(ISEED1)
C                                                                       HELP 740
      READ(5,103)MODEL,NBOOS,NORBI,NLUTS,NCONF,ISEED,ISEED1
  103 FORMAT(I1,I4,3I5,40X,2I10)
      IS(1)=NBOOS                                                       HELP 770
      IS(2)=NORBI                                                       HELP 780
      K2=NBOOS                                                          HELP 790
      K3=NORBI                                                          HELP 800
      K4=NLUTS
      NSEED=ISEED
      NSEED1=ISEED1
      RETURN                                                            HELP 810
    8 CONTINUE                                                          HELP 820
C                                                                       HELP 830
C*****GENERATE DEVIATE  FOR A TWO PARAMETER EXPONENTIAL DISTRIBUTION    HELP 840
C                                                                       HELP 850
      PARAM(1,1)=K2                                                     HELP 860
      NBETA=K3-K2
      PARAM(1,2)=NBETA
      K5=EXPO(1)                                                        HELP 880
      RETURN                                                            HELP 890
    9 CONTINUE                                                          HELP 900
C                                                                       HELP 910
C*****GENERATE DEVIATE  FOR THE LOG NORMAL DISTRIBUTION                 HELP 920
C                                                                       HELP 930
      F2=FLOAT(K2)/1000.                                                HELP 940
      F3=FLOAT(K3)/1000.                                                HELP 950
      PARAM(1,2)=SQRT(ALOG(F3**2/F2**2+1.))                             HELP 960
      PARAM(1,1)=ALOG(F2)-.5*(PARAM(1,2))**2                            HELP 970
      K5=RLOGN(1)*1000                                                  HELP 980
      RETURN                                                            HELP 990
   10 CONTINUE                                                          HELP1000
C                                                                       HELP1010
C*****PRINT OUT OF SUMMARY TABLE                                        HELP1020
C                                                                       HELP1030
      NCALL=NCALL+1                                                     HELP1040
      CALL      PRINT(NCALL,K2,K3,K4,K5)
      RETURN                                                            HELP1060
   11 CONTINUE
C
C*****THIS FUNCTION PERMITS THE ADVANCE BLOCK TO BE INCREMENTED BY A
```

```
C*****CONSTANT
C*****TRANSFER THE VALUE OF ARGUMENT K2 TO ARGUMENT K5
C
C      GPSS OUTPUT CAN BE OBTAINED FROM EITHER THE K4 OR K5 FIELDS
       K5=K2
       K4=K2
       RETURN
    12 CONTINUE
C
C*****GENERATE DEVIATE FOR UNIFORM DISTRIBUTION
C
       A = K2
       B = K3
C      GPSS OUTPUT CAN BE OBTAINED FROM EITHER THE K4 OR K5 FIELDS
       K4 = UNFRM1(A,B)
       K5=K4
       RETURN
    13 CONTINUE
C
C*****GENERATE DEVIATE FOR BETA DISTRIBUTION
C
       PARAM(1,1) = K3
       PARAM(1,2) = K2
       PARAM(1,3) = K4
       CALL PERTXF(1,SMEAN,SVAR)
       K5 = BETADP(1,SMEAN,SVAR)
       RETURN
       END                                              HELP1070

@HDG,P    MAP

@PKT,S    MAP
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.MISSON
      SUBROUTINE MISSON(NP1,NP2,NP3,NP4,NP5,K)
C
C*****THIS PROGRAM  GENERATES TRAFFIC REQUIREMENTS FOR THE SHUTTLE
C     OPERATIONAL MISSIONS.
C*****THE SELECTION OF NEXT SCHEDULED LAUNCH IS BASED ON THE FREQUENCY
C     OF MISSION OCCURANCE.
C
      REAL LATIME
      DIMENSION OPPORT(5,3),MILGTH(5,3)
      DIMENSION LCONS(5),LDATE(5),INTERL(5),LFREQ(5,5)
      COMMON /HEFL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      COMMON /MISPRT/NELTA1,NELTA2,NELTA3,NELTA4,NELTA5
C
C*****VARIABLE K DEFINES THE TRAFFIC MODEL TO BE SIMULATED
C            K=1 - 20 LAUNCHES/YEAR
C            K=2 - 35 LAUNCHES/YEAR
C            K=3 - 45 LAUNCHES/YEAR
C            K=4 - 55 LAUNCHES/YEAR
C            K=5 - 75 LAUNCHES/YEAR
C*****ARRAY LFREQ CONTAINS THE FREQUENCY REQUIREMENTS FOR THE 5 TRAFFIC
C     MODELS.    EACH ROW DEFINES A NEW MODEL.(FLIGHTS/YEAR)
      DATA LFREQ/10,8,0,0,2,
     A 14,6,2,12,1,
     B 16,8,4,16,1,
     C 20,8,8,19,2,
     D 35,8,6,19,5/
C*****DATA ON THE MINUMUM TIME BETWEEN LAUNCH REQUESTS(DAYS)
      DATA LCONS/5,3,3,2,1/
C*****DATA FOR THE TIME UNTIL SECOND LAUNCH OPPORTUNITY(HOURS)
      DATA OPPORT/8.33,8.33,8.33,8.33,0.,
     A 14.17,0.,0.,14.17,8.33,
     B 24.0,14.17,14.17,24.0,14.17/
C*****DATA ON THE RANGE OF TYPICAL ORBITER MISSION LENGTHS(DAYS)
      DATA MILGTH/2,1,4,1,1,
     A 3,2,5,3,5,
     B 5,3,6,5,20/
C*****VARIABLE NTYPE IS THE NUMBER OF DIFFERENT TYPES OF MISSIONS
      NTYPE=5
C*****VARIABLE I DEFINES THE MISSION TYPE
C            I=1 - SHUTTLE PAYLOAD PLACEMENT
C            I=2 - SPACE STATION LOGISTICS
C            I=3 - TUG P/L PLACEMENT & FUEL RESUPPLY
C            I=4 - DEPARTMENT OF DEFENSE
C            I=5 - SPACE RESCUE & OTHER TYPES
C
C
C******************GO THRU THIS LEG ON THE FIRST CALL ONLY**************
C
      IF(LL.NE.0)GO TO 2
C*****VARIABLE LL IS SET = 1 WHEN EXECUTION PASSES THRU THIS LEG
      LL = 1
      SAVE = 0.0
C*****DETERMINE THE FIRST LAUNCH DATE FOR EACH OF THE MISSION TYPES
C*****EVENLY SPACED LAUNCH DATES HAVE BEEN ASSUMMED FOR THE FIRST 5
C     LAUNCHES.
```

```
      LR=LFREQ(1,K)+LFREQ(2,K)+LFREQ(3,K)+LFREQ(4,K)+LFREQ(5,K)
      LFIRST=260/LR
      DO 7 I=1,NTYPE
      LDATE(I)=I*LFIRST*2
C*****IF MISSION TYPE IS NOT DEFINED, SET THE LAUNCH DATE = 99999999
      IF(LFREQ(I,K).EQ.0)LDATE(I)=99999999
C*****MISSIONS ARE TO BE SCHEDULED AT THE SAME FREQUENCY WITH 10 PERCENT
C     DEVIATION
      INTERL(I)=.50*FLOAT(260/LFREQ(I,K))
      IF(I.EQ.1)INTERL(1)=INTERL(1)+2
    7 CONTINUE
C*******************************************************************
    2 CONTINUE
C
C*****CALCULATE THE NEXT LAUNCH TRAFFIC REQUIREMENT
C
      NMIN=MINO(LDATE(1),LDATE(2),LDATE(3),LDATE(4),LDATE(5))
      DO 20 I=1,NTYPE
      IF(LDATE(I).EQ.NMIN)GO TO 21
   20 CONTINUE
      MERROR=1
      WRITE(6,100)MERROR
  100 FORMAT(1X,6HERROR ,I1)
C
C*****ASSIGN LAUNCH HOUR FOR MISSION ASSIGNMENT
C
   21 NP2=I
      TIME=LDATE(I)
       HOUR = UNFRM(0.,24.)
      LATIME =  TIME +  HOUR/24.
      NP3=LATIME*1000.-SAVE
      NP3=IABS(NP3)
C
C*****GENERATE THE HISTOGRAM FOR THE TIME BETWEEN LAUNCH REQUESTS
C
      IF(NP3.LT.2000)NELTA1=NELTA1+1
      IF(NP3.GE.2000.AND.NP3.LT.3000)NELTA2=NELTA2+1
      IF(NP3.GE.3000.AND.NP3.LT.5000)NELTA3=NELTA3+1
      IF(NP3.GE.5000.AND.NP3.LT.10000)NELTA4=NELTA4+1
      IF(NP3.GE.10000.AND.NP3.LT.20000)NELTA5=NELTA5+1
      SAVE=LATIME*1000.
      NP4=SAVE
C
C*****CALCULATE THE FIRST LAUNCH OPPORTUNITY  AFTER THE NOMINAL LAUNCH TIME
C
      DO 40 J=1,3
      PARAM(I,J)=OPPORT(I,J)
   40 CONTINUE
      XOPPOR = DICRET(I)
      NP5=XOPPOR/24.*1000.
C
C*****SELECT THE MISSION LENGTH OF THE ORBITER
C
      DO 50 J=1,3
      PARAM(I,J)=MILGTH(I,J)
   50 CONTINUE
```

```
      NPI=TRIAG(I)*1000.
C
C*****CALCULATE THE DATE OF THE NEXT LAUNCH
C
      NCOUNT=0
   60 CONTINUE
      NCOUNT=NCOUNT+1
      A=-INTERL(I)
      B=INTERL(I)
      NVAR=UNFRM(A,B)
      NDAY=LDATE(I)+260/LFREQ(I,K)+NVAR
      DO 65 J=1,NTYPE
C
C*****AFTER 50 RESCHEDULES , THE CONSTRAINT MAY BE OVERRIDDEN
C
      IF(NCOUNT.EQ.50)WRITE(6,107)I
  107 FORMAT(////'  *TRAFFIC MODEL VIOLATED MINUMUM TIME BETWEEN LAUNCH
     1REQUESTS CONSTRAINT*     TYPE=',I3)
      IF(NCOUNT.EQ.50)GO TO 68
C
C*****DETERMINE IF TIME BETWEEN LAUNCH REQUESTS MEETS THE MINUMUM REQ.
C     IF NOT- TRY TO SCHEDULE ANOTHER LAUNCH DATE
C
      NTEST=NDAY-LDATE(J)
      IF(IABS(NTEST).LE.LCONS(K))GO TO 60
   65 CONTINUE
   68 CONTINUE
      LDATE(I)=NDAY
      RETURN
      END

@HDG,P     OUTPUT

@PRT,S     OUTPUT
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.OUTPUT
      SUBROUTINE OUTPUT (ICOM,LOC,NAME,IX,IY,IZ,ISELM,NBA,NBB,MEAN,MOD)
C
C  SUBROUTINE TO DETERMINE TYPE OF DISTRIBUTION AND WRITE
C  CORRESPONDING DATA.
C
      DIMENSION NAME(2),ICOM(11)
      DATA K9/2HK9/,K3/2HK3/,K4/2HK4/,K8/2HK8/
      DATA K11/3HK11/,K12/3HK12/,K13/3HK13/
C
C  TEST TO DETERMINE IF DISTRIBUTION IS LOGNORMAL (K9)
      IF(IX.EQ.K9) GO TO 1
C
C  TEST TO DETERMINE IF DISTRIPUTION IS TRIANGULAR (K3)
      IF(IX.EQ.K3) GO TO 2
C
C  TEST TO DETERMINE IF DISTRIBUTION IS NORMAL (K4)
      IF(IX.EQ.K4) GO TO 3
C
C  TEST TO DETERMINE IF DISTRIBUTION IS EXPONENTIAL (K8)
      IF(IX.EQ.K8) GO TO 4
C
C  TEST TO DETERMINE IF DISTRIBUTION IS CONSTANT(K11)
       IF (IX.EQ.K11) GO TO 5
C      TEST TO DETERMINE IF DISTRIBUTION IS UNIFORM (K12)
       IF (IX.EQ.K12) GO TO 6
C
C      TEST TO DETERMINE IF DISTRIBUTION IS BETA (K13)
       IF (IX.EQ.K13) GO TO 7
C
       RETURN
    1  CONTINUE
C
C  CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C  REAL NUMBERS.
       CALL  REALN (RES1,IY)
       CALL  REALN (RES2,IZ)
C
       RES1 = RES1/1000.0
       RES2 = RES2/1000.0
C
C  WRITE RESULTS FOR LOGNORMAL DISTRIBUTION
       WRITE (6,90) ICOM,RES1,RES2
       WRITE(6,84)
       RETURN
    2  CONTINUE
C
C  CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C  REAL NUMBERS.
       CALL  REALN (RES1,IY)
       CALL  REALN (RES2,IZ)
       CALL  REALN (RES3,MEAN)
C
       RES1 = RES1/1000.0
       RES2 = RES2/1000.0
       RES3 = RES3/1000.0
```

```
C
C  WRITE RESULTS FOR TRIANGULAR DISTRIBUTION
      WRITE (6,91) ICOM,RES1,RES2,RES3
      WRITE(6,84)
      RETURN
    3 CONTINUE
C
C  CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C  REAL NUMBERS.
      CALL  REALN (RES1,IY)
      CALL  REALN (RES2,IZ)
C
      RES1 = RES1/1000.0
      RES2 = RES2/1000.0
C
C  WRITE RESULTS FOR NORMAL DISTRIBUTION
      WRITE (6,92) ICOM,RES1,RES2
      WRITE(6,84)
      RETURN
    4 CONTINUE
C
C  CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C  REAL NUMBERS.
      CALL  REALN (RES1,IY)
      CALL  REALN (RES2,IZ)
C
      RES1 = RES1/1000.0
      RES2 = RES2/1000.0
C
C  WRITE RESULTS FOR EXPONENTIAL DISTRIBUTION
      WRITE (6,93) ICOM,RES1,RES2
      WRITE(6,84)
      RETURN
  5    CONTINUE
C
C  CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C  REAL NUMBERS
        CALL REALN (RES3,IY)
C
        RES3=RES3/1000.0
C
C  WRITE RESULTS FOR CONSTANT DISTRIBUTION
        WRITE (6,94)ICOM,RES3
        WRITE (6,84)
        RETURN
     6 CONTINUE
C
C       CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
C       REAL NUMBERS
        CALL REALN(RES1,IY)
        CALL REALN(RES2,IZ)
        RES3=(RES1+RES2)/2000.0
        RES1=RES1/1000.0
        RES2=RES2/1000.0
C
C       WRITE RESULTS FOR UNIFORM DISTRIBUTION
```

```
                WRITE(6,95) ICOM,RES1,RES3,RES2
                WRITE (6,84)
                RETURN
              7 CONTINUE
      C
      C         CALL SUBROUTINE REALN TO CONVERT ALPHANUMERIC NUMBERS TO
      C         REAL NUMBERS
                CALL REALN(RES1,IY)
                CALL REALN(RES2,IZ)
                CALL REALN(RES3,MEAN)
                RES1=RES1/1000.0
                RES2=RES2/1000.0
                RES3=RES3/1000.0
      C
      C         WRITE RESULTS FOR BETA DISTRIBUTION
                WRITE (6,96) ICOM,RES1,RES2,RES3
                WRITE (6,84)
                RETURN
      C
             84 FORMAT (1X,1HI,67X,1HI,13X,1HI,14X,1HI,17X,1HI,13X,1HI)
      C
          90    FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HLOGNORMAL  ,
               1 1X,1HI,1X,5HMEAN=,F7.4,1X,1HI,1X,8HST.DEV =,F7.4,1X,1HI,13X,1HI)
      C
          91    FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HTRIANGULAR ,
               1 1X,1HI,1X,5HMIN =,F7.4,1X,1HI,1X,8HMODE    =,F7.4,
               2 1X,1HI,1X,4HMAX=,F7.4,1X,1HI)
      C
          92    FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HNORMAL     ,
               1 1X,1HI,1X,5HMEAN=,F7.4,1X,1HI,1X,8HST.DEV =,F7.4,1X,1HI,13X,1HI)
      C
          93    FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HEXPONENTIAL,
               1 1X,1HI,1X,5HMIN =,F7.4,1X,1HI,1X,8HMEAN    =,F7.4,1X,1HI,13X,1HI)
      C
           94    FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HCONSTANT   ,
               1 1X,1HI,1X,5HMEAN=, F7.4,1X,1HI,17X,1HI,13X,1HI)
             95 FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HUNIFORM    ,
               1 1X,1HI,1X,5HMIN =,F7.4,1X,1HI,1X,8HMEAN    =,F7.4,
               2 1X,1HI,1X,4HMAX=,F7.4,1X,1HI)
      C
             96 FORMAT(1X,1HI,1X,A5,2A6,8A6,1X,1HI,1X,11HBETA       ,
               1 1X,1HI,1X,5HMIN =,F7.4,1X,1HI,1X,8HMEAN    =,F7.4,
               2 1X,1HI,1X,4HMAX=,F7.4,1X,1HI)
                END
```

@HDG,P PERTXF

@PRT,S PERTXF

FURPUR HA18-06/30-13:31

```
420223*TPFS.PERTXF
             SUBROUTINE PERTXF(J,SMEAN,SVAR)
             COMMON /HEPL/ PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
             SVAR = (PARAM(J,3)-PARAM(J,2))**2./36.
             SMEAN = (PARAM(J,1)*4. + PARAM(J,2) + PARAM(J,3))/6.0
      C
      C*** CALCULATE THE MEAN AND VARIANCE FOR THE GIVEN PARAMETERS
      C***
      C***            PARAM(J,1) CONTAINS THE MODE
      C***            PARAM(J,2) CONTAINS THE LOWER LIMIT
      C***            PARAM(J,3) CONTAINS THE UPPER LIMIT
      C***            BMEAN CONTAINS THE NORMALIZED MEAN OVER THE INTERVAL (0,1
      C***            BVAR  CONTAINS THE NORMALIZED VARIANCE OVER THE INTERVAL
      C***
      C
             BMEAN =     (SMEAN  -PARAM(J,2))/(PARAM(J,3)-PARAM(J,2))
             BVAR = SVAR/(PARAM(J,3)-PARAM(J,2))**2.0
             SMEAN = BMEAN*(BMEAN*(1.0-BMEAN)/BVAR-1.0)
             SVAR = SMEAN*((1.0-BMEAN)/BMEAN)
             RETURN
             END

@HDG,P      PRINT

@PRT,S      PRINT
FURPUR HA18-06/30-13:31
```

420223•TPFS.PRINT

```
      SUBROUTINE PRINT(NCALL,K2,K3,K4,K5)
C
C**************THIS ROUTINE PRINTS OUT THE SUMMARY RESULTS TABLE*******
C
      REAL MAHUF,LUTREF
      INTEGER P
      INTEGER GLOB
      COMMON /HLPPRT/NBOOS,NORBI,NLUTS,TITLE(20),NSEED,NSEED1
      COMMON /MISPRT/NELTA1,NELTA2,NELTA3,NELTA4,NELTA5
      COMMON/EQ2/JF2(1)
      COMMON/STOR1/JS1(1)
      COMMON/STOR2/JS2(1)
      COMMON/QUE5/JQ5(1)
      COMMON/QUE6/JQ6(1)
      COMMON/EKSES/JEKS(1)
      COMMON/WAY/P
      COMMON/STAT3/ISN
      P=6
      GO TO (10,20,30,40,50,60,70,80,90),NCALL
   10 CONTINUE
      WRITE(P,100)
  100 FORMAT(1H1,30X,'***TABLE II. SUMMARY RESULTS****')
      WRITE(P,102)(TITLE(I),I=1,20)
  102 FORMAT(///10X,'CASE STUDY: ',20A4)
      WRITE(P,103)
  103 FORMAT( ///10X,'INITIAL CONDITIONS:',T60,'BOOSTER',T70,'ORBITER')
      WRITE(P,104)NBOOS ,NORBI
  104 FORMAT(15X,'NUMBER OF VEHICLE STAGES   ',T63,I2,T73,I2)
      MSAB=MGTH1U(JS2(1))+MGTH2U(JS1(1))
      MSAO=MGTH1U(JS2(10))+MGTH2U(JS1(10))
      WRITE(P,170)MSAB,MSAO
  170 FORMAT(15X,'SERVICE CAPACITY OF SAFING AREA',T63,I2,T73,I2)
      MFB=MGTH1U(JS2(2))+MGTH2U(JS1(2))
      MFO=MGTH1U(JS2(11))+MGTH2U(JS1(11))
      WRITE(P,105)MFB,MFO
  105 FORMAT(15X,'SERVICE CAPACITY OF MAINTENANCE FACILITY'    ,T63,I2,
     1 T73,I2)
      MCOTB=MGTH1U(JS2(3))+MGTH2U(JS1( 3))
      MCOTO=MGTH1U(JS2(12))+MGTH2U(JS1(12))
      WRITE(P,176)MCOTB,MCOTO
  176 FORMAT(15X,'SERVICE CAPACITY OF CHECKOUT & TEST FACILITY'  ,T63,
     1 I2,T73,I2)
      MHF =MGTH1U(JS2(20))+MGTH2U(JS1(20))
      WRITE(P,171)MHF
  171 FORMAT(15X,'SERVICE CAPACITY OF MATE & HOOK-UP FACILITY',T68,I2)
      LUTRF=MGTH1U(JS2(18))+MGTH2U(JS1(18))
      WRITE(P,172)LUTRF
  172 FORMAT(15X,'SERVICE CAPACITY OF LUT REFURBISHMENT FACILITY',T68,
     1  I2)
      WRITE(P,173)NLUTS
  173 FORMAT(15X,'NUMBER OF LUTS-',T40,I2)
C*************************************************************************
      NPADS=1
      IF(JF2(2).GT.0)NPADS=2
      IF(JF2(3).GT.0)NPADS=3
```

```
 13 WRITE(P,106)NPADS
106 FORMAT(15X,'NUMBER OF LAUNCH PADS-',T40,I2)
    K2=JEKS(30)+JEKS(2)+JEKS(3)+JEKS(4)+JEKS(22)+JEKS(23)+JEKS(24)+
   1  JEKS(25)+JEKS(8)
    WRITE(P,174)
174 FORMAT(///10X,'TRAFFIC MODEL STATISTICS:')
    NFR=260000./FLOAT(JEKS(21))*FLOAT(K2)
    WRITE(P,157)NFR
157 FORMAT(15X,'MISSION MODEL LAUNCH RATE (FLIGHTS/YEAR)-',I6)
    NLR=NELTA1+NELTA2+NELTA3+NELTA4+NELTA5
    PELTA1=FLOAT(NELTA1)*100/NLR
    PELTA2=FLOAT(NELTA2)*100/NLR
    PELTA3=FLOAT(NELTA3)*100/NLR
    PELTA4=FLOAT(NELTA4)*100/NLR
    PELTA5=FLOAT(NELTA5)*100/NLR
    WRITE(P,158)PELTA1,PELTA2,PELTA3,PELTA4,PELTA5
158 FORMAT(15X,'DISTRIBUTION OF TIME BETWEEN LAUNCH REQUESTS-',
   A/16X,F5.1,'% OF LAUNCHES ARE SCHEDULED 1 DAY APART',
   B/16X,F5.1,'% OF LAUNCHES ARE SCHEDULED 2 DAYS APART',
   C/16X,F5.1,'% OF LAUNCHES ARE SCHEDULED 3 DAYS TO 1 WEEK APART',
   D/16X,F5.1,'% OF LAUNCHES ARE SCHEDULED 1 WEEK TO 2 WEEKS APART',
   E/16X,F5.1,'% OF LAUNCHES ARE SCHEDULED 2 WEEKS TO 1 MONTH APART')
    WRITE(P,107)K2
107 FORMAT(///10X,'OPERATIONAL EVALUATION PARAMETERS FOR ',I4,' SIMULA
   1TIONS:')
    PLR=FLOAT(JEKS(30))*100./K2
    WRITE(P,152)JEKS(30),PLR
152 FORMAT(15X,'THE NUMBER OF SUCCESSFUL LAUNCH REQUESTS ARE',I4,
   1 ' OR',F5.1,' PERCENT')
    NMLR=K2-JEKS(30)
    PNMLR=FLOAT(NMLR)*100./K2
    WRITE(P,108)NMLR,PNMLR
108 FORMAT(15X,'THE NUMBER OF UNSUCCESSFUL REQUESTS ARE',I4,
   1 ' OR',F5.1,' PERCENT')
    GLOBP =(FLOAT(JEKS(26))/FLOAT(JEKS(30)))*100.
    GLOB = JEKS(26)
    WRITE(P,373)GLOB,GLOBP
373 FORMAT(15X,'THE NUMBER OF VEHICLES LAUNCHED AT A SUBSEQUENT WINDOW
   1 IS',I3,' OR',F5.1,'PERCENT')
    WRITE(P,596)
596 FORMAT(             16X,'REASONS FOR REQUEST FAILURES-')
    NBUNA=          JEKS(3)
    PBUNA=FLOAT(NBUNA)*100./K2
    WRITE(P,109)NBUNA,PBUNA
109 FORMAT(17X,'BOOSTER UNAVAILABILITY',T50,I4,' TIMES OR',F5.1,' PERC
   1ENT')
    NOUNA=          JEKS(4)
    POUNA=FLOAT(NOUNA)*100./K2
    WRITE(P,110)NOUNA,POUNA
110 FORMAT(17X,'ORBITER UNAVAILABILITY',T50,I4,' TIMES OR',F5.1,' PERC
   1ENT')
    PUNA=FLOAT(JEKS(2))*100./K2
    WRITE(P,156)JEKS(2),PUNA
156 FORMAT(17X,'BOTH STAGES UNAVAILABILE',T50,I4,' TIMES OR',F5.1,' PE
   1RCENT')
    LFAIL=JEKS(8)
```

```
      PLFAIL=FLOAT(LFAIL)*100./K2
      WRITE(P,175)LFAIL,PLFAIL
  175 FORMAT(17X,'LUT UNAVAILABILITY',T50,I4,' TIMES OR',F5.1,' PERCENT
     1 ')
      NPUAU=JEKS(25)
      PPUAU=FLOAT(NPUAU)*100./K2
      WRITE(P,150)NPUAU,PPUAU
  150 FORMAT(17X,'PAD UNAVAILABILITY',T50,I4,' TIMES OR', F5.1,' PERCENT
     1')
      NFAIL=JEKS(22)+JEKS(24)+JEKS(23)
      PFAIL=FLOAT(NFAIL)*100/K2
      WRITE(P,151)NFAIL,PFAIL
  151 FORMAT(17X,'FAILURES AND REMOVAL FROM PAD',T50,I4,' TIMES OR'
     1 ,F5.1,' PERCENT')
      WRITE(P,111)
  111 FORMAT(///10X,'AVERAGE PHASE TIMES PER VEHICLE ELEMENT:', T60,
     1 'BOOSTER',T70,'ORBITER')
      TLSTL=K4/1000.
      TIMEIO=K5/1000.
      RETURN
C**********************************************************************
   20 CONTINUE
      TLTSB=K2/1000.
      TLTSO=K3/1000.
      WRITE(P,113)TLTSB,TLTSO
  113 FORMAT(15X,'LANDING TO STORAGE(DAYS)            ',T61,F5.2,T71,F5.2)
      STORB=K4/1000.
      STORO=K5/1000.
      WRITE(P,114)STORB,STORO
  114 FORMAT(15X,'STORAGE(DAYS)                       ',T61,F5.2,T71,F5.2)
      WRITE(P,161)TLSTL,TLSTL
  161 FORMAT(15X,'LEAVE STORAGE TO LAUNCH(DAYS)',T61,F5.2,T71,F5.2)
      WRITE(P,160)TIMEIO
  160 FORMAT(15X,'MEAN ON-ORBIT MISSION TIME(DAYS)',T71,F5.2)
      TOTALB=TLTSB+STORB+TLSTL
      TOTALO=TLTSO+STORO+TLSTL+TIMEIO
      WRITE(P,112)TOTALB,TOTALO
  112 FORMAT(15X,'TURNAROUND- LAUNCH TO LAUNCH(DAYS)',T61,F5.2,T71,F5.2)
      RETURN
C**********************************************************************
   30 CONTINUE
      WRITE(P,180)
  180 FORMAT(1H1)
      WRITE(P,115)
  115 FORMAT(///10X,'WAITING STATISTICS:                         ',
     1 T60,'BOOSTER',T70,'ORBITER')
      SAFB = K2/1000.
      SAFO = K3/1000.
      PDESB = FLOAT(JQ5(1)-JQ6(1))*100./JQ5(1)
      PDESO = FLOAT(JQ5(10)-JQ6(10))*100./JQ5(10)
      WRITE(P,300)PDESB,PDESO,SAFB,SAFO
  300 FORMAT(15X,'PERCENT OF VEHICLES DELAYED FOR SAFING',T62,F5.2,T72,F
     15.2,/15X,'AVERAGE QUEUE TIME FOR SAFING(DAYS)',T62,F5.2,T72,F5.2 )
      PQMB=FLOAT(JQ5(2)-JQ6(2))*100./JQ5(2)
      PQMO=FLOAT(JQ5(11)-JQ6(11))*100./JQ5(11)
      WRITE(P,154)PQMB,PQMO
```

```
  154 FORMAT(15X,'PERCENT OF VEHICLES DELAYED FOR MAINT.',T62,F5.2,T72,
     1 F5.2)
      TMQB=K4/1000.
      TMQO=K5/1000.
      WRITE(P,116)TMQB,TMQO
  116 FORMAT(15X,'AVERAGE QUEUE TIME FOR MAINTENANCE(DAYS)',T62,F5.2,T72
     1,F5.2)
      RETURN
C****************************************************************************
   40 CONTINUE
      PCHEKB = FLOAT(JQ5(3)-JQ6(3))*100./JQ5(3)
      PCHEKO = FLOAT(JQ5(12)-JQ6(12))*100./JQ5(12)
      CHECKB = K2/1000.
      CHECKO = K3/1000.
      WRITE(P,330)PCHEKB,PCHEKO,CHECKB,CHECKO
  330 FORMAT(
     215X,'PERCENT OF VEHICLES DELAYED FOR CHECKOUT&TEST',T62,F5.2,T72,
     3 F5.2,/15X,'AVERAGE QUEUE TIME FOR CHECKOUT & TEST(DAYS)',T62,F5.2
     42,T72,F5.2)
C     WRITE(P,117)       DUMMY WRITE STATEMENT
C 117 FORMAT(15X,'AVERAGE QUEUE TIME FOR                 ',T61,F5.2,T71,
C    1 F5.2)
      MAHUF = K4/1000.
      LUTREF = K5/1000.
      PMAHU = FLOAT(JQ5(20)-JQ6(20))*100./JQ5(20)
      PLUT  = FLOAT(JQ5(18)-JQ6(18))*100./JQ5(18)
      WRITE(P,301)PMAHU,MAHUF,PLUT,LUTREF
  301 FORMAT(15X,'PERCENT OF VEHICLES DELAYED FOR M & H',T65,F5.2,/15X,'
     1AVERAGE QUEUE TIME FOR MATE & HOOKUP(DAYS)',T65,F5.2,/15X,'PERCENT
     2 OF LUTS DELAYED FOR REFURBISHMENT',T65,F5.2,/15X,'AVERAGE QUEUE T
     3IME FOR LUT REFURBISH FACILITY',T65,F5.2)
      RETURN
C**********************************************************************
   50 CONTINUE
      PQP=FLOAT(JQ5(21)-JQ6(21))*100./JQ5(21)
      WRITE(P,155)PQP
  155 FORMAT(15X,'PERCENT OF VEHICLES DELAYED FOR PAD ACCESS',T65,F5.2)
      TQP=K2/1000.
      WRITE(P,118)TQP
  118 FORMAT(15X,'AVERAGE QUEUE TIME FOR PAD(DAYS)        ',T65,F5.2)
      RETURN
C**********************************************************************
   60 CONTINUE
      SAUB=K2/10.
      SAUO=K3/10.
      WRITE(P,119)SAUB,SAUO
  119 FORMAT(///10X,'FACILITY UTILIZATION:',T60,'BOOSTER',T70,'ORBITER',
     1 /15X,'SAFING AREA(PERCENT)',T61,F5.2,T71,F5.2)
      SMUB=K4/10.
      SMUO=K5/10.
      WRITE(P,120)SMUB,SMUO
  120 FORMAT(15X,'MAINTENANCE(PERCENT)           ',T61,F5.2,T71,F5.2)
      RETURN
C**********************************************************************
   70 CONTINUE
      COTUB=K2/10.
```

```
      COTUO=K3/10.
      WRITE(P,177)COTUB,COTUO
  177 FORMAT(15X,'CHECKOUT & TEST(PERCENT)',T61,F5.2,T71,F5.2)
      RETURN
C*******************************************************************
   80 CONTINUE
      PMHU=K2/10.
      WRITE(P,121)PMHU
  121 FORMAT(15X,'MATE & HOOK-UP(PERCENT)',T45,F5.2)
      PLRU=K3/10.
      WRITE(P,122)PLRU
  122 FORMAT(15X,'LUT REFURBISHMENT(PERCENT)',T45,F5.2)
      RETURN
C****************************************************************************
   90 CONTINUE
      PDU1=K2/10.
      WRITE(P,123)PDU1
  123 FORMAT(/10X,'PAD UTILIZATION:',/
     1        15X,'LAUNCH PAD 1 (PERCENT)',T45,F5.2)
      IF (NPADS-2) 61,62,62
   62 PDU2=K3/10.
      WRITE(P,124)PDU2
  124 FORMAT(15X,'LAUNCH PAD 2 (PERCENT)',T45,F5.2)
      IF (NPADS-3) 61,1003,1003
 1003 PDU3=K4/10.
      WRITE(P,1004) PDU3
 1004 FORMAT (15X,'LAUNCH PAD 3 (PERCENT) ',T45,F5.2)
   61 CONTINUE
C*****WRITE OUT CASE STUDY COMMENT CARD
      WRITE(P,199)(TITLE(I),I=1,20)
  199 FORMAT(1H1,///'******',20A4,'******')
      WRITE(P,200)NBOOS ,NORBI
  200 FORMAT(////,10X,'THE FLEET SIZE IS',I2,' BOOSTERS &'I2,' ORBITERS
     1')
      WRITE(P,201)NSEED,NSEED1
  201 FORMAT(10X,'RANDOM NUMBER SEED FOR TRAFFIC MODEL SIMULATOR = ',I10
     1 ,/10X,'RANDOM NUMBER SEED FOR DISTRIBUTION FUNCTIONS = ',I10)
C
C     DETERMINE IF TABLE III OUTPUT IS REQUIRED
      IF(ISN.LT.9) RETURN
      CALL SAMPLE(K1,K2,K3,K4,K5)
C
      RETURN
C*******************************************************************
      END
```

@HDG,P PROCS

420223*TPFS.REALN

```
      SUBROUTINE REALN (RES,IN)
C
C  SUBROUTINE TO CHANGE ALPHANUMERIC NUMBERS TO REAL NUMBERS
C
C  SUBSCRIPTED VARIABLE IA WILL CONTAIN 1 CHARACTER IN EACH POSITION
C
      DIMENSION IA(6)
C
C*****************************************************************
C
C  FUNCTION FLD IS A LIBRARY FUNCTION WHICH ENABLES USER TO OBTAIN
C  INDIVIDUAL BITS FROM AN ALPHANUMERIC WORD AND STORE
C  THEM IN ANOTHER WORD.
C
C  FLD (I,K,E)  I = STARTING BIT    K = BIT WIDTH    E = WORD NAME
C
C*****************************************************************
C
C  IN POSITION 1 OF IA STORE BITS  0 TO  5 OF WORD IN
      IA(1) = FLD ( 0,6,IN)
C
C  IN POSITION 2 OF IA STORE BITS  6 TO 11 OF WORD IN
      IA(2) = FLD ( 6,6,IN)
C
C  IN POSITION 3 OF IA STORE BITS 12 TO 17 OF WORD IN
      IA(3) = FLD (12,6,IN)
C
C  IN POSITION 4 OF IA STORE BITS 18 TO 23 OF WORD IN
      IA(4) = FLD (18,6,IN)
C
C  IN POSITION 5 OF IA STORE BITS 24 TO 29 OF WORD IN
      IA(5) = FLD (24,6,IN)
C
C  IN POSITION 6 OF IA STORE BITS 30 TO 35 OF WORD IN
      IA(6) = FLD (30,6,IN)
C
C  TEST TO DETERMINE IF 1ST CHARACTER OF WORD IS A K
      IF (IA(1).NE.16) GO TO 1
C
C  TEST TO DETERMINE IF 6TH CHARACTER OF WORD IS A BLANK
      IF (IA(6).NE. 5) GO TO 10
C
C  TEST TO DETERMINE IF 5TH CHARACTER OF WORD IS A BLANK
      IF (IA(5).NE. 5) GO TO 11
C
C  TEST TO DETERMINE IF 4TH CHARACTER OF WORD IS A BLANK
      IF (IA(4).NE. 5) GO TO 12
C
C  TEST TO DETERMINE IF 3RD CHARACTER OF WORD IS A BLANK
      IF (IA(3).NE. 5) GO TO 13
C
C  TEST TO DETERMINE IF 2ND CHARACTER OF WORD IS A BLANK
      IF (IA(2).NE. 5) GO TO 14
      GO TO 1
C
```

```
C*******************************************************************
C
C  TO CHANGE ALPHANUMERIC NUMBER TO A REAL NUMBER SUBSTRACT 48.
C
C  ALPHANUMERIC ZERO IS REPRESENTED BY CHARACTER 48 WHICH
C  OCCUPIES 6 BITS IN AN ALPHANUMERIC WORD.
C
C  ALPHANUMERIC ONE IS REPRESENTED BY CHARACTER 49 ETC.
C
C*******************************************************************
C
   10 CONTINUE
C
C  EQUATION TO CHANGE 5 CHARACTER ALPHANUMERIC NUMBER TO 5
C  CHARACTER INTEGER
      IRES = (IA(6)-48) + (IA(5)-48)*10 + (IA(4)-48)*100
     1 + (IA(3)-48)*1000 + (IA(2)-48)*10000
C
C  EQUATION TO CHANGE INTEGER TO REAL NUMBER FOR TABLE
      RES = IRES
      RETURN
   11 CONTINUE
C
C  EQUATION TO CHANGE 4 CHARACTER ALPHANUMERIC NUMBER TO 4
C  CHARACTER INTEGER
      IRES = (IA(5)-48) + (IA(4)-48)*10 + (IA(3)-48)*100
     1 + (IA(2)-48)*1000
C
C  EQUATION TO CHANGE INTEGER TO REAL NUMBER FOR TABLE
      RES = IRES
      RETURN
   12 CONTINUE
C
C  EQUATION TO CHANGE 3 CHARACTER ALPHANUMERIC NUMBER TO 3
C  CHARACTER INTEGER
      IRES = (IA(4)-48) + (IA(3)-48)*10 + (IA(2)-48)*100
C
C  EQUATION TO CHANGE INTEGER TO REAL NUMBER FOR TABLE
      RES = IRES
      RETURN
   13 CONTINUE
C
C  EQUATION TO CHANGE 2 CHARACTER ALPHANUMERIC NUMBER TO 2
C  CHARACTER INTEGER
      IRES = (IA(3)-48) + (IA(2)-48)*10
C
C  EQUATION TO CHANGE INTEGER TO REAL NUMBER FOR TABLE
      RES = IRES
      RETURN
   14 CONTINUE
C
C  EQUATION TO CHANGE 1 CHARACTER ALPHANUMERIC NUMBER TO 1
C  CHARACTER INTEGER
      IRES = IA(2)-48
C
C  EQUATION TO CHANGE INTEGER TO REAL NUMBER FOR TABLE
```

```
                RES = IRES
                RETURN
              1 CONTINUE
          C
          C  EQUATION WILL RESULT IN ERROR, PRINTING STARS
                RES = 9999999.9
                RETURN
                END

@HDG,P    RLOGN

@PRT,S    RLOGN
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.RLOGN
          FUNCTION RLOGN (J)                                          LGNS  10
      C                                                               LGNS  10
      C*****THE PARAMETERS USED WITH RLOGN ARE THE MEAN AND STANDARD DEVIATIONLGNS  20
      C*****OF A NORMAL DISTRIBUTION                                  LGNS  30
      C
          VA= RNORM (J)                                               LGNS  40
          RLOGN=EXP(VA)                                               LGNS  50
          RETURN
          END

@HDG,P      RNORM

@PRT,S    RNORM
FURPUR HA18-06/30-13:31
```

```
420223*TPFS.RNORM
          FUNCTION RNORM (J)                                              NRLS  10
          COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
          RA=DRANDI(ISEED1)
          RB=DRANDI(ISEED1)
          V=(-2.0*ALOG(RA))**0.5*COS (6.283*RB)                           NRLS  80
          RNORM  = V*PARAM (J,2) + PARAM (J,1)                            NRLS  90
          RETURN                                                          NRLS 150
          END                                                             NRLS 160

@HDG,P      SAMPLE

@PRT,S      SAMPLE
FURPUR HA18-06/30-13:31
```

```
420223*TPF$.SAMPLE
      COMPILER(DATA=SHORT)                                        SAMP  10
      SUBROUTINE SAMPLE(K1,K2,K3,K4,K5)                           SAMP  20
      COMMON /ISTAT/ NCONF                                        SAMP  30
      COMMON/STAT2/NFLTS                                          SAMP  40
      COMMON/STATS/NHEAD                                          SAMP  50
      COMMON/WAY/P                                                SAMP  60
      COMMON/STAT3/ISN                                            SAMP  70
      COMMON K(100)                                               SAMP  80
      EQUIVALENCE (K(28),NCLOCK)                                  SAMP  90
      INTEGER P                                                   SAMP 100
      DIMENSION T(32,2),SAVE(14,6),NHEAD(14,11)                   SAMP 110
C                                                                 SAMP 120
C                                                                 SAMP 130
C    THESE ARE VALUES OF THE STUDENT T VARIABLE                   SAMP 140
C                                                                 SAMP 150
      DATA((T(J,K),K=1,2),J=1,18)/                                SAMP 160
     112.706,63.657,                                              SAMP 170
     2 4.303, 9.925,                                              SAMP 180
     3 3.182, 5.841,                                              SAMP 190
     4 2.776, 4.604,                                              SAMP 200
     5 2.571, 4.032,                                              SAMP 210
     6 2.447, 3.707,                                              SAMP 220
     7 2.365, 3.499,                                              SAMP 230
     8 2.306, 3.355,                                              SAMP 240
     9 2.262, 3.250,                                              SAMP 250
     * 2.228, 3.169,                                              SAMP 260
     1 2.201, 3.106,                                              SAMP 270
     2 2.179, 3.055,                                              SAMP 280
     3 2.160, 3.012,                                              SAMP 290
     4 2.145, 2.977,                                              SAMP 300
     5 2.131, 2.947,                                              SAMP 310
     6 2.120, 2.921,                                              SAMP 320
     7 2.110, 2.898,                                              SAMP 330
     8 2.101, 2.878/                                              SAMP 340
      DATA((T(J,K),K=1,2),J=19,31)/                               SAMP 350
     9 2.093, 2.861,                                              SAMP 360
     * 2.086, 2.845,                                              SAMP 370
     1 2.080, 2.831,                                              SAMP 380
     2 2.074, 2.819,                                              SAMP 390
     3 2.069, 2.807,                                              SAMP 400
     4 2.064, 2.797,                                              SAMP 410
     5 2.060, 2.787,                                              SAMP 420
     6 2.056, 2.779,                                              SAMP 430
     7 2.052, 2.771,                                              SAMP 440
     8 2.048, 2.763,                                              SAMP 450
     9 2.045, 2.756,                                              SAMP 460
     * 2.042, 2.750,                                              SAMP 470
     1 1.960, 2.576/                                              SAMP 480
C                                                                 SAMP 490
C                                                                 SAMP 500
C     STORE HEADINGS TO BE USED IN PRINTING                       SAMP 510
C                                                                 SAMP 520
      DATA((NHEAD(I,J),J=1,11),I=1,9)/                            SAMP 530
C                                                                 SAMP 540
     162HTHE PERCENTAGE OF SUCCESSFUL LAUNCH REQUESTS            ,SAMP 550
```

```
      262HTHE PERCENTAGE OF UNSUCCESSFUL LAUNCH REQUESTS              ,SAMP 560
      362HTHE PERCENTAGE OF VEHICLES LAUNCHED AT A SUBSEQUENT WINDOW  ,SAMP 570
      462H      BOOSTER UNAVAILABILITY                                ,SAMP 580
      562H      ORBITER UNAVAILABILITY                                ,SAMP 590
      662H      BOTH STAGES UNAVAILABLE                               ,SAMP 600
      762H      LUT UNAVAILABILITY                                    ,SAMP 610
      862H      PAD UNAVAILABILITY                                    ,SAMP 620
      962H      FAILURES AND REMOVAL FROM PAD                         /SAMP 630
C                                                                      SAMP 640
C                                                                      SAMP 650
C      FIELD DEFINITIONS                                               SAMP 660
C                                                                      SAMP 670
C      K1 IS A DUMMY                                                   SAMP 680
C                                                                      SAMP 690
C      K2 CORRESPONDS TO THE 'Y' FIELD ENTRY ON THE HELP CARD AND IS   SAMP 700
C         THE IDENTIFICATION NUMBER OF THE STATISTIC BEING COLLECTED.  SAMP 710
C                                                                      SAMP 720
C      K3 CORRESPONDS TO THE 'Z' FIELD ENTRY ON THE HELP CARD AND IS   SAMP 730
C         THE TYPE OF STATISTIC BEING COLLECTED                        SAMP 740
C         K3=1 FOR A NUMBER                                            SAMP 750
C         K3=2 FOR PERCENTAGE UTILIZATION                              SAMP 760
C         K3=3 FOR AVERAGE TIME                                        SAMP 770
C                                                                      SAMP 780
C      K4 CORRESPONDS TO THE 'MEAN' FIELD ENTRY ON THE HELP CARD AND   SAMP 790
C         IS THE GPSS SYSTEM VARIABLE THAT REPRESENTS                  SAMP 800
C         1) NUMBER OF ENTRIES                                         SAMP 810
C         2) AVERAGE UTILIZATION                                       SAMP 820
C         3) AVERAGE TIME IN A FACILITY, STORAGE OR QUEUE              SAMP 830
C                                                                      SAMP 840
C      K5 CORRESPONDS TO THE 'MOD' FIELD ENTRY ON THE HELP CARD AND    SAMP 850
C         IS THE GPSS SYSTEM VARIABLE (WHEN REQUIRED) THAT REPRESENTS  SAMP 860
C         THE NUMBER OF ENTRIES INTO THE BLOCK TYPE SPECIFIED          SAMP 870
C         IN THE 'MEAN' FIELD                                          SAMP 880
C                                                                      SAMP 890
C                                                                      SAMP 900
C      ARRAY ELEMENT DEFINITIONS                                       SAMP 910
C                                                                      SAMP 920
C      SAVE(I,1) = PREVIOUS READING                                    SAMP 930
C      SAVE(I,2) = SUM X'S                                             SAMP 940
C      SAVE(I,3) = SUM X SQUARED'S                                     SAMP 950
C      SAVE(I,4) = K4 READING FOR THE FIRST 100 SIMULATIONS            SAMP 960
C      SAVE(I,5) = 1/2 WIDTH OF THE CONFIDENCE INTERVAL                SAMP 970
C      SAVE(I,6) = PREVIOUS K5 READING                                 SAMP 980
C                                                                      SAMP 990
C                                                                      SAMP1000
C                                                                      SAMP1010
C                                                                      SAMP1020
C      CONTINUE WITH STATISTICAL SAMPLING                              SAMP1030
C                                                                      SAMP1040
C      DETERMINE IF A REQUEST FOR OUTPUT HAS OCCURRED                  SAMP1050
       IF(P.EQ.6) GO TO 60                                             SAMP1060
C                                                                      SAMP1070
C      EXECUTE THIS PORTION OF THE PROGRAM ONLY ONCE                   SAMP1080
       IF(LKT.GE.1) GO TO 90                                           SAMP1090
       LKT = 1                                                         SAMP1100
       NCI=NCONF                                                       SAMP1110
```

```
C                                                                       SAMP1120
C     IDENTIFY THE LEVEL OF SIGNIFICANCE                                SAMP1130
      IF(NCI.EQ.99) NCONF=2                                             SAMP1140
      IF(NCI.NE.99) NCONF=1                                             SAMP1150
   90 CONTINUE                                                          SAMP1160
C                                                                       SAMP1170
C     CHANGE FIXED POINT VARIABLES K4 AND K5 TO FLOATING POINT          SAMP1180
      FK4=FLOAT(K4)                                                     SAMP1190
      FK5=FLOAT(K5)                                                     SAMP1200
C                                                                       SAMP1210
C     SPECIFY THE K5 VALUE BASED ON THE TYPE OF STATISTIC BEING         SAMP1220
C     COLLECTED AS SPECIFIED BY K3                                      SAMP1230
      IF(K3.EQ.1) FK5=1.                                                SAMP1240
      IF(K3.EQ.2) FK5=FLOAT(NCLOCK)/1000.                               SAMP1250
C                                                                       SAMP1260
C     ADJUST THE K4 VALUE BASED ON THE TYPE OF STATISTIC BEING          SAMP1270
C     COLLECTED AS SPECIFIED BY K3                                      SAMP1280
      IF(K3.GE.2) FK4=FK4*FK5/1000.                                     SAMP1290
C                                                                       SAMP1300
C     COUNT THE NUMBER OF OBSERVATIONS THAT HAVE BEEN MADE              SAMP1310
      IF(K2.LE.1) NOB=NOB+1                                             SAMP1320
C                                                                       SAMP1330
C                                                                       SAMP1340
C     COLLECT AND RETAIN THE STATISTICS FROM THE FIRST 100 SIMS         SAMP1350
      IF(NOB.LT.2) SAVE(K2,4)=FK4                                       SAMP1360
      IF(NOB.LT.2) SAVE(K2,6)=FK5                                       SAMP1370
      IF(NOB.EQ.1) RETURN                                               SAMP1380
C                                                                       SAMP1390
C                                                                       SAMP1400
C     DETERMINE THE CHANGE IN MEAN FROM THE CURRENT OBSERVATION AND THE SAMP1410
C     100TH OBSERVATION                                                 SAMP1420
      XK2=FK4-SAVE(K2,4)                                                SAMP1430
C                                                                       SAMP1440
C     DETERMINE AND RETAIN THE DIFFERENCES IN MEANS BETWEEN THE CURRENT SAMP1450
C     OBSERVATION AND THE PREVIOUS OBSERVATION TO OBTAIN A 'NEW' MEAN   SAMP1460
      XI = XK2 - SAVE(K2,1)                                             SAMP1470
      SAVE(K2,1) = XK2                                                  SAMP1480
      XK2 = XI                                                          SAMP1490
C                                                                       SAMP1500
C     DETERMINE THE INCREMENT IN BLOCK ENTRIES THAT HAVE OCCURRED       SAMP1510
C     SINCE THE PREVIOUS OBSERVATION                                    SAMP1520
      BINCR=FK5-SAVE(K2,6)                                              SAMP1530
C                                                                       SAMP1540
C     RETAIN THE TOTAL NUMBER OF BLOCK ENTRIES                          SAMP1550
      SAVE(K2,6)=FK5                                                    SAMP1560
C                                                                       SAMP1570
C                                                                       SAMP1580
C     ADJUST THE 'NEW' MEAN BASED ON THE TYPE OF STATISTIC BEING        SAMP1590
C     COLLECTED AS SPECIFIED BY K3                                      SAMP1600
      IF(K3.EQ.1) XK2 = XK2*2.                                          SAMP1610
      IF(K3.EQ.2) XK2 = XK2*100/BINCR                                   SAMP1620
      IF(K3.EQ.3) XK2 = XK2/BINCR                                       SAMP1630
C                                                                       SAMP1640
C                                                                       SAMP1650
C                                                                       SAMP1660
C     DETERMINE THE T STATISTIC VALUE AND THE ESTIMATED                 SAMP1670
```

```
      C     POPULATION STANDARD DEVIATION                                      SAMP1680
            SAVE(K2,2) = SAVE(K2,2) + XK2                                      SAMP1690
            SAVE(K2,3) = SAVE(K2,3) + (XK2*XK2)                                SAMP1700
      C                                                                        SAMP1710
      C     IF AN INSUFFICIENT NUMBER OF OBSERVATIONS HAS OCCURRED, RETURN     SAMP1720
            N = NOB                                                            SAMP1730
            IF(N.LE.2) RETURN                                                  SAMP1740
      C                                                                        SAMP1750
      C     DETERMINE THE DEGREES OF FREEDOM FOR THE STUDENT T DISTRIBUTION    SAMP1760
            NDFREE = N-2                                                       SAMP1770
            IF(NDFREE.GT.31) NDFREE = 31                                       SAMP1780
      C                                                                        SAMP1781
      C     DETERMINE AND RETAIN THE ESTIMATED DEVIATION                       SAMP1782
            CALL STDV(SAVE(K2,2),SAVE(K2,3),N,STDEV)                           SAMP1790
            SAVE(K2,5)=T(NDFREE,NCONF)*STDEV                                   SAMP1800
      C                                                                        SAMP1810
            RETURN                                                             SAMP1820
      C                                                                        SAMP1830
      C     -------------------------------------------------------------------SAMP1840
      C                                                                        SAMP1850
      C                                                                        SAMP1860
      C                                                                        SAMP1870
      C                                                                        SAMP1880
      C                                                                        SAMP1890
         60 CONTINUE                                                           SAMP1900
      C                                                                        SAMP1910
      C     BEGIN OUTPUT OF CONFIDENCE INTERVALS                               SAMP1920
      C                                                                        SAMP1930
      C                                                                        SAMP1940
      C                                                                        SAMP1950
      C                                                                        SAMP1960
      C     OUTPUT FOR TABLE III HEADINGS                                      SAMP1970
      C                                                                        SAMP1980
            WRITE(P,100)                                                       SAMP1990
        100 FORMAT(1H1,40X,'***TABLE III. STATISTICAL RESULTS***')             SAMP2000
            WRITE(P,110)                                                       SAMP2010
        110 FORMAT(///35X,'CONFIDENCE INTERVALS ON REQUESTED STATISTICS')      SAMP2020
            WRITE(P,120)                                                       SAMP2030
        120 FORMAT(/25X,'WARNING - CONFIDENCE INTERVALS WITH LIMITS OF 0 OR 10SAMP2040
           10 ARE PROBABLY')                                                   SAMP2050
            WRITE(P,130) NCI                                                   SAMP2060
        130 FORMAT(35X,'NOT',I3,' PERCENT CONFIDENCE INTERVALS')               SAMP2070
            WRITE(P,140) N                                                     SAMP2080
        140 FORMAT(/35X,'STATISTICS BASED ON',I3,' OBSERVATIONS.'///)          SAMP2090
            WRITE(P,160) NCI                                                   SAMP2100
        160 FORMAT(T83,I2,' PERCENT CONFIDENCE INTERVALS'                      SAMP2110
           1/T79,'LOWER LIMIT',7X,'MEAN',6X,'UPPER LIMIT')                     SAMP2120
            WRITE(P,150) NFLTS                                                 SAMP2130
        150 FORMAT(///5X,'STATISTICAL CONFIDENCE INTERVALS FOR ',I4,' SIMULATISAMP2140
           1ONS.')                                                             SAMP2150
      C                                                                        SAMP2160
      C                                                                        SAMP2170
      C     OUTPUT FOR THE OPERATIONAL EVALUATION PARAMETERS                   SAMP2180
      C                                                                        SAMP2190
            DO 50 I=1,9                                                        SAMP2200
            AVG=SAVE(I,2)/FLOAT(N)                                             SAMP2210
```

```
                XLOW=AVG-SAVE(I,5)                                          SAMP2220
                HIGH=AVG+SAVE(I,5)                                          SAMP2230
                IF(HIGH.GE.100.) HIGH=100.                                  SAMP2240
                IF(XLOW.LE.0.) XLOW=0.0                                     SAMP2250
                WRITE(P,200)(NHEAD(I,J),J=1,11),XLOW,AVG,HIGH               SAMP2260
          200   FORMAT(10X,11A6,T81,F6.2,8X,F6.2,8X,F6.2)                   SAMP2270
      C                                                                     SAMP2280
                IF(I.NE.3) GO TO 50                                         SAMP2290
                WRITE(P,190)                                                SAMP2300
          190   FORMAT(/12X,'REASONS FOR REQUEST FAILURES:')                SAMP2310
      C                                                                     SAMP2320
           50   CONTINUE                                                    SAMP2330
      C                                                                     SAMP2340
      C                                                                     SAMP2350
      C         DETERMINE IF ANY USER REQUESTED STATISTICS HAVE BEEN REQUESTED  SAMP2360
                IF(ISN.LE.9) GO TO 500                                      SAMP2370
      C                                                                     SAMP2380
      C                                                                     SAMP2390
      C         OUTPUT FOR THE USER REQUESTED STATISTICS                    SAMP2400
      C                                                                     SAMP2410
                WRITE(P,210)                                                SAMP2420
          210   FORMAT(////5X,'THE FOLLOWING ARE THE USER REQUESTED STATISTICS'/)  SAMP2430
                DO 52 I=10,ISN                                              SAMP2440
                AVG=SAVE(I,2)/FLOAT(N)                                      SAMP2450
                XLOW=AVG-SAVE(I,5)                                          SAMP2460
                HIGH=AVG+SAVE(I,5)                                          SAMP2470
                IF(HIGH.GE.100.) HIGH=100.0                                 SAMP2480
                IF(XLOW.LE.0.) XLOW=0.0                                     SAMP2490
                WRITE(P,200)(NHEAD(I,J),J=1,11),XLOW,AVG,HIGH               SAMP2500
           52   CONTINUE                                                    SAMP2510
      C                                                                     SAMP2520
          500   WRITE(P,999)                                                SAMP2530
          999   FORMAT(1H1)                                                 SAMP2540
      C                                                                     SAMP2550
                RETURN                                                      SAMP2560
                END                                                         SAMP2570
```

@HDG,P STAT

@PRT,S STAT
FURPUR HA18-06/30-13:31

420223•TPFS.STAT

```
      SUBROUTINE STAT                                                   STAT  10
      COMMON/STATS/NHEAD                                                STAT  20
      DIMENSION NHEAD(14,11),ICOM(11)                                   STAT  30
      COMMON/STAT3/ISN                                                  STAT  40
      DATA KEY/4HCONF/,JEND/5HTART /                                    STAT  50
C                                                                       STAT  60
C                                                                       STAT  70
C     BEGIN SEARCH FOR A STATISTICAL REQUEST                            STAT  80
    1 CONTINUE                                                          STAT  90
      READ(5,100) ICOM,IREM,NUM                                         STAT 100
  100 FORMAT(1X,10A6,A5,A4,A1)                                          STAT 110
C                                                                       STAT 120
C                                                                       STAT 130
C     IS THIS THE LAST CARD                                             STAT 140
C     IF THIS IS THE LAST CARD, RETURN                                  STAT 150
      IF(ICOM(2).EQ.JEND) RETURN                                        STAT 160
C                                                                       STAT 170
C                                                                       STAT 180
C     NO,THIS IS NOT THE LAST CARD                                      STAT 190
C     IS THIS A STATISTICAL REQUEST                                     STAT 200
C     IF NOT, READ ANOTHER CARD                                         STAT 210
      IF(KEY.NE.IREM) GO TO 1                                           STAT 220
C                                                                       STAT 230
C                                                                       STAT 240
C     THIS CARD IS A STATISTICAL REQUEST                                STAT 250
C     CONVERT THE ALPHANUMERIC INPUT (NUM) TO NUMERIC                   STAT 260
      NUMI=0                                                            STAT 270
      NUMI=FLD(0,6,NUM)                                                 STAT 280
C                                                                       STAT 290
C                                                                       STAT 300
C     DETERMINE THE STATISTICAL REQUEST IDENTIFICATION NUMBER           STAT 310
      NUM=NUMI-48                                                       STAT 320
C                                                                       STAT 330
C                                                                       STAT 340
C     DETERMINE IF THE NUMERIC IDENTIFICATION IS LEGAL                  STAT 350
C     IF NOT, GO TO ERROR IDENTIFICATION ROUTINE                        STAT 360
      IF(NUM.LT.1.OR.NUM.GT.5) GO TO 3                                  STAT 370
C                                                                       STAT 380
C                                                                       STAT 390
C     STORE HEADING FOR FUTURE REFERENCE                                STAT 400
      ISN=NUM+9                                                         STAT 410
      DO 2 I=1,11                                                       STAT 420
    2 NHEAD(ISN,I)=ICOM(I)                                              STAT 430
C                                                                       STAT 440
C                                                                       STAT 450
C     READ ANOTHER CARD                                                 STAT 460
      GO TO 1                                                           STAT 470
C                                                                       STAT 480
C                                                                       STAT 490
C     ------------------------------------------------------------------STAT 500
C                                                                       STAT 510
C                                                                       STAT 520
C     ERROR IDENTIFICATION ROUTINE                                      STAT 530
C                                                                       STAT 540
    3 J=J+1                                                             STAT 550
```

```
            IF(J.GT.1) GO TO 4                                               STAT 560
            WRITE(6,300)                                                     STAT 570
        300 FORMAT(1H1)                                                      STAT 580
            WRITE(6,310)                                                     STAT 590
        310 FORMAT(10X,'THE FOLLOWING HEADING(S) ARE DISREGARDED IN THE CONFIDSTAT 600
           1ENCE INTERVAL'/5X,'CALCULATIONS DUE TO AN ILLEGAL C. I. IDENTIFICASTAT 610
           2TION NUMBER')                                                    STAT 620
          4 WRITE(6,320) (ICOM(I),I=1,11)                                    STAT 630
        320 FORMAT(15X,11A6)                                                 STAT 640
      C                                                                      STAT 650
      C                                                                      STAT 660
      C     READ ANOTHER CARD                                                STAT 670
            GO TO 1                                                          STAT 680
            END                                                              STAT 690

@HDG,P      STDV

@PRT,S    STDV
FURPUR HA18-06/30-13:31
```

```
420223*TPF$.STDV
      SUBROUTINE STDV(XSAVE,XSAVE2,N,STDEV)                             STDV  10
C                                                                       STDV  20
C     ELIMINATE THE ONE OBSERVATION COUNTED FOR THE FIRST 100 SIMULATIONSTDV  30
      N = N - 1                                                         STDV  40
C                                                                       STDV  50
C     IF AN INSUFFICIENT NUMBER OF OBSERVATIONS HAS OCCURRED, RETURN    STDV  60
      IF(N.LT.2) RETURN                                                 STDV  70
C                                                                       STDV  80
C     DETERMINE THE ESTIMATED STANDARD DEVIATION                        STDV  90
      XN = FLOAT(N)                                                     STDV 100
      STDEV = ABS(XSAVE2-(XSAVE**2./XN))/(XN-1.)                        STDV 110
      STDEV = SQRT(STDEV)                                               STDV 120
C                                                                       STDV 130
C     ADJUST TO CONFORM WITH CONFIDENCE INTERVAL EQUATION               STDV 140
      STDEV = STDEV/SQRT(XN)                                            STDV 150
C                                                                       STDV 160
      RETURN                                                            STDV 170
      END                                                               STDV 180

@HDG,P       TABLE1

@PRT,S       TABLE1
FURPUR HA18-06/30-13:31
```

```
420223*TPF$.TABLE1
      SUBROUTINE TABLE1
C
C     TABLE1 IS A SUBROUTINE DESIGNED TO BE USED IN CONJUNCTION WITH
C     A GPSS PROGRAM CARD DECK.  THE FUNCTION OF THIS SUBROUTINE IS TO
C     READ EACH CARD IN A GPSS DECK AND DETERMINE IF IT IS A COMMENT
C     CARD WITH THE KEY WORD IN COLUMNS 68-71.(THE KEY WORD IS IDST)
C     AFTER IT HAS FOUND A COMMENT CARD WITH THE PROPER KEY WORD IT THEN
C     READS CARDS UNTIL IT FINDS A HELP CARD.  AFTER THE HELP CARD IS FOUND
C     THE SUBROUTINE OUTPUT IS CALLED TO PRINT OUT THE INFORMATION
C     IN TABLULAR FORM
C
      DIMENSION NAME(2),ICOM(11)
      DATA JAST/1H*/,JHELP/4HHELP/,KEY/6H IDST /,IEND/6H END  /
      DATA JEND/6HSTART /
C
      WRITE(6,80)
      WRITE(6,81)
      WRITE (6,88)
      WRITE (6,70)
      WRITE (6,84)
      WRITE(6,82)
      WRITE(6,83)
      WRITE(6,84)
      WRITE (6,70)
      WRITE(6,84)
      WRITE (6,85)
      WRITE (6,85)
C
C     BEGIN THE SEARCH FOR THE NEXT COMMENT CARD AT THIS POINT
    5 CONTINUE
      READ (5,100) IAST,ICOM,IREM
  100 FORMAT (A1,A5,2A6,9A6)
C
C     IS THIS THE LAST CARD
      IF(ICOM(2).EQ.JEND) GO TO 999
C
C     NO, THIS IS NOT THE LAST CARD
C     IS THIS CARD A COMMENT CARD
      IF (IAST .NE. JAST) GO TO 5
C
C     YES, THIS CARD IS A COMMENT CARD
C     DOES THIS COMMENT CARD HAVE A KEY
      IF (IREM .NE. KEY) GO TO 5
C
C     YES THIS COMMENT CARD HAS A KEY
   10 CONTINUE
      READ(5,100) IAST,LOC,NAME,IX,IY,IZ,ISELM,NBA,NBB,MEAN,MOD,IREM
C     WAS THE CARD JUST READ A HELP CARD
      IF (JHELP .NE. NAME(1)) GO TO 10
C
C     YES, THE CARD JUST READ WAS A HELP CARD
C     CALL OUTPUT AND PRINT APPROPRIATE INFORMATION
      CALL       OUTPUT (ICOM,LOC,NAME,IX,IY,IZ,ISELM,NBA,NBB,MEAN,MOD)
      GO TO 5
C     YES, THIS IS THE LAST CARD
```

```
      999 CONTINUE
          WRITE (6,70)
          WRITE (6,99)
C
       70 FORMAT (1X,130H--------------------------------------------------------
         1---------------------------------------------------------------------
         2-------------)
C
       80 FORMAT (1H1)
C
       81 FORMAT      (44X,'***TABLE I.  OPERATION TIME DATA IN DAYS***')
C
       82 FORMAT         (1X,1HI,29X,9HOPERATION,29X,1HI,2X,
         1 9HDISTRIBU-,2X,1HI,16X,14HINPUT DATA FOR,16X,1HI)
C
       83 FORMAT (1X,1HI,28X,11HDESCRIPTION,28X,1HI,2X,9HTION TYPE,2X,
         1 1HI,12X,23HDISTRIBUTION PARAMETERS,11X,1HI)
C
       84 FORMAT (1X,1HI,67X,1HI,13X,1HI,46X,1HI)
C
       85 FORMAT (1X,1HI,67X,1HI,13X,1HI,14X,1HI,17X,1HI,13X,1HI)
C
       88 FORMAT (//)
C
       99 FORMAT (1H1)
          RETURN
          END
```

@HDG,P   TRIAG

@PRT,S   TRIAG
FURPUR HA18-06/30-13:31

```
420223*TPF$.TRIAG
      FUNCTION TRIAG(JP)                                               GPST2110
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      RNUM = DRAND(ISEED)                                              GPST2130
      IF(RNUM-(PARAM(JP,2)-PARAM(JP,1))/(PARAM(JP,3)-PARAM(JP,1)))1,1,2 GPST2140
    1 TRIAG = PARAM(JP,1) + SQRT((PARAM(JP,3)-PARAM(JP,1))             GPST2150
     1*(PARAM(JP,2)- PARAM(JP,1))*RNUM)                                GPST2160
      RETURN                                                           GPST2170
    2 X = (PARAM(JP,3)-PARAM(JP,2))**2 -(PARAM(JP,2) - PARAM(JP,3))*   GPST2180
     1(PARAM(JP,2) - PARAM(JP,1)- (PARAM(JP,3)-PARAM(JP,1))*RNUM)      GPST2190
      TRIAG = PARAM(JP,3) - SQRT(X)                                    GPST2200
      RETURN                                                           GPST2210
      END                                                              GPST2220

@HDG,P     TRIAG1

@PRT,S     TRIAG1
FURPUR HA18-06/30-13:31
```

```
420223*TPF$.TRIAG1
      FUNCTION TRIAG1(JP)
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      RNUM = DRAND1(ISEED1)
      IF(RNUM-(PARAM(JP,2)-PARAM(JP,1))/(PARAM(JP,3)-PARAM(JP,1)))1,1,2
    1 TRIAG1 = PARAM(JP,1) + SQRT((PARAM(JP,3)-PARAM(JP,1))
     1*(PARAM(JP,2)- PARAM(JP,1))*RNUM)
      RETURN
    2 X = (PARAM(JP,3)-PARAM(JP,2))**2 -(PARAM(JP,2) - PARAM(JP,3))*
     1(PARAM(JP,2) - PARAM(JP,1)- (PARAM(JP,3)-PARAM(JP,1))*RNUM)
      TRIAG1 = PARAM(JP,3) - SQRT(X)
      RETURN
      END

@HDG,P      UNFRM

@PRT,S    UNFRM
FURPUR HA18-06/30-13:31
```

420223*TPFS.UNFRM

```
      FUNCTION UNFRM (A,B)                                             UFMS  10
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)
      RNUM=DRAND(ISEED)                                                UFMS  60
      UNFRM  = A+(B-A)*RNUM                                            UFMS  70
      RETURN                                                           UFMS  80
      END                                                              UFMS  90
```

@HDG,P UNFRM1

@PRT,S UNFRM1
FURPUR HA18-06/30-13:31

420223*TPFS.UNFRM1

```
      FUNCTION UNFRM1(A,B)                                          UNFM  10
      COMMON /HEPL/PARAM(7,3),LL,ISEED,ISEED1,ISTORE(2,10),IS(2)    UNFM  20
      RNUM=DRAND1(ISEED1)                                           UNFM  30
      UNFRM1 = A+(B-A)*RNUM                                         UNFM  40
      RETURN                                                        UNFM  50
      END                                                           UNFM  60
```

APPENDIX C. EXAMPLE TABLE1 OUTPUT

PRECEDING PAGE BLANK NOT FILMED

TABLE I. OPERATION TIME DATA IN DAYS

OPERATION DESCRIPTION	DISTRIBUTION TYPE	INPUT DATA FOR DISTRIBUTION PARAMETERS		
MATE BOOSTER,ORBITER AND LUT	TRIANGULAR	MIN = 1.8000	MODE = 2.0000	MAX= 2.2000
COUNTDOWN PREPARATION	TRIANGULAR	MIN = 1.5000	MODE = 2.0000	MAX= 2.5000
FINAL COUNTDOWN	CONSTANT	MEAN= .0830		
REMOVE POST LAUNCH LUT FROM PAD	TRIANGULAR	MIN = 1.2500	MODE = 1.5000	MAX= 1.7500
TRANSPORT LUT TO MAINT.,REFURBISH/TEST,TRANSPORT TO STORAGE	TRIANGULAR	MIN = 2.7000	MODE = 3.0000	MAX= 3.3000
REFURBISH LAUNCH PAD	TRIANGULAR	MIN = .9500	MODE = 1.0000	MAX= 1.0500
LIFTOFF THROUGH SEPARATION	CONSTANT	MEAN= .0020		
BOOSTER COAST,REENTRY,FLYBACK AND LANDING	TRIANGULAR	MIN = .0640	MODE = .0700	MAX= .0760
BOOSTER SAFING	TRIANGULAR	MIN = .9000	MODE = 1.0000	MAX= 1.1000
TRANSPORT BOOSTER TO MAINTENANCE	TRIANGULAR	MIN = .0500	MODE = .1000	MAX= .1500
BOOSTER MAINTENANCE	TRIANGULAR	MIN = 2.2500	MODE = 2.5000	MAX= 2.7500
BOOSTER ADDITIONAL MAINTENANCE	EXPONENTIAL	MIN = .5000	MEAN = 1.0000	
TRANSPORT BOOSTER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE	TRIANGULAR	MIN = 1.9000	MODE = 2.0000	MAX= 2.1000
ORBITER SAFING	TRIANGULAR	MIN = .9000	MODE = 1.0000	MAX= 1.1000
TRANSPORT ORBITER TO MAINTENANCE	TRIANGULAR	MIN = .0500	MODE = .1000	MAX= .1500
ORBITER MAINTENANCE	TRIANGULAR	MIN = 2.2500	MODE = 2.5000	MAX= 2.7500
ORBITER ADDITIONAL MAINTENANCE	EXPONENTIAL	MIN = .5000	MEAN = 1.0000	
TRANSPORT ORBITER TO SYSTEMS TEST, TEST,TRANSPORT TO STORAGE	TRIANGULAR	MIN = 1.9000	MODE = 2.0000	MAX= 2.1000
REMOVE BOOSTER, ORBITER AND LUT FROM THE PAD	TRIANGULAR	MIN = 1.8000	MODE = 2.0000	MAX= 2.2000
TRANSPORT VEHICLE, SEPARARTE ORBITER, TRANSPORT ORBITER TO MAINT	TRIANGULAR	MIN = .5500	MODE = .7500	MAX= .9500
SEPARATE BOOSTER FROM LUT, TRANSPORT BOOSTER TO MAINTENANCE	TRIANGULAR	MIN = .5500	MODE = .7500	MAX= .9500
BOOSTER MAINTENANCE ON RETURN FROM PAD	EXPONENTIAL	MIN = .5000	MEAN = 7.5000	
BOOSTER VEHICLE SYSTEM TEST FACILITY ON RETURN FROM PAD	TRIANGULAR	MIN = 1.9000	MODE = 2.0000	MAX= 2.1000
ORBITER MAINTENANCE ON RETURN FROM PAD	EXPONENTIAL	MIN = .5000	MEAN = 7.5000	
ORBITER VEHICLE SYS TEST FACILITY ON RETURN FROM PAD	TRIANGULAR	MIN = 1.9000	MODE = 2.0000	MAX= 2.1000

APPENDIX D. SAMPLE SOS MODEL OUTPUT

TABLE II. SUMMARY RESULTS

CASE STUDY: DETERMINATION OF FLEET SIZE REQUIREMENTS

INITIAL CONDITIONS:	BOOSTER	ORBITER
NUMBER OF VEHICLE STAGES	4	5
SERVICE CAPACITY OF SAFING AREA	1	1
SERVICE CAPACITY OF MAINTENANCE FACILITY	1	2
SERVICE CAPACITY OF CHECKOUT & TEST FACILITY	2	2
SERVICE CAPACITY OF MATE & HOOK-UP FACILITY	1	
SERVICE CAPACITY OF LUT REFURBISHMENT FACILITY	2	
NUMBER OF LUTS- 3		
NUMBER OF LAUNCH PADS- 2		

TRAFFIC MODEL STATISTICS:

MISSION MODEL LAUNCH RATE (FLIGHTS/YEAR)- 53

DISTRIBUTION OF TIME BETWEEN LAUNCH REQUESTS-

- .8% OF LAUNCHES ARE SCHEDULED 1 DAY APART
- 14.1% OF LAUNCHES ARE SCHEDULED 2 DAYS APART
- 48.6% OF LAUNCHES ARE SCHEDULED 3 DAYS TO 1 WEEK APART
- 35.3% OF LAUNCHES ARE SCHEDULED 1 WEEK TO 2 WEEKS APART
- 1.2% OF LAUNCHES ARE SCHEDULED 2 WEEKS TO 1 MONTH APART

OPERATIONAL EVALUATION PARAMETERS FOR 502 SIMULATIONS:

THE NUMBER OF SUCCESSFUL LAUNCH REQUESTS ARE 484 OR 96.4 PERCENT

THE NUMBER OF UNSUCCESSFUL REQUESTS ARE 18 OR 3.6 PERCENT

THE NUMBER OF VEHICLES LAUNCHED AT A SUBSEQUENT WINDOW IS 7 OR 1.4PERCENT

REASONS FOR REQUEST FAILURES-		
BOOSTER UNAVAILABILITY	5 TIMES OR	1.0 PERCENT
ORBITER UNAVAILABILITY	0 TIMES OR	.0 PERCENT
BOTH STAGES UNAVAILABILE	0 TIMES OR	.0 PERCENT
LUT UNAVAILABILITY	1 TIMES OR	.2 PERCENT
PAD UNAVAILABILITY	0 TIMES OR	.0 PERCENT
FAILURES AND REMOVAL FROM PAD	12 TIMES OR	2.4 PERCENT

AVERAGE PHASE TIMES PER VEHICLE ELEMENT:	BOOSTER	ORBITER
LANDING TO STORAGE(DAYS)	6.12	5.87
STORAGE(DAYS)	7.93	9.67
LEAVE STORAGE TO LAUNCH(DAYS)	5.32	5.32
MEAN ON-ORBIT MISSION TIME(DAYS)		3.42
TURNAROUND- LAUNCH TO LAUNCH(DAYS)	19.37	24.29

WAITING STATISTICS:

		BOOSTER	ORBITER
PERCENT OF VEHICLES DELAYED FOR SAFING		.00	3.93
AVERAGE QUEUE TIME FOR SAFING(DAYS)		.00	.39
PERCENT OF VEHICLES DELAYED FOR MAINT.		9.48	.61
AVERAGE QUEUE TIME FOR MAINTENANCE(DAYS)		2.69	.16
PERCENT OF VEHICLES DELAYED FOR CHECKOUT&TEST		.00	.00
AVERAGE QUEUE TIME FOR CHECKOUT & TEST(DAYS)		.00	.00
PERCENT OF VEHICLES DELAYED FOR M & H	.40		
AVERAGE QUEUE TIME FOR MATE & HOOKUP(DAYS)	.63		
PERCENT OF LUTS DELAYED FOR REFURBISHMENT	.20		
AVERAGE QUEUE TIME FOR LUT REFURBISH FACILITY	.00		
PERCENT OF VEHICLES DELAYED FOR PAD ACCESS	.00		
AVERAGE QUEUE TIME FOR PAD(DAYS)	.00		

FACILITY UTILIZATION:

		BOOSTER	ORBITER
SAFING AREA(PERCENT)		19.80	19.90
MAINTENANCE(PERCENT)		57.70	29.90
CHECKOUT & TEST(PERCENT)		20.30	20.30
MATE & HOOK-UP(PERCENT)	40.90		
LUT REFURBISHMENT(PERCENT)	30.50		

PAD UTILIZATION:

LAUNCH PAD 1 (PERCENT)	68.70
LAUNCH PAD 2 (PERCENT)	30.00

00000 DETERMINATION OF FLEET SIZE REQUIREMENTS 000000

THE FLEET SIZE IS 4 BOOSTERS & 5 ORBITERS
RANDOM NUMBER SEED FOR TRAFFIC MODEL SIMULATOR = 0
RANDOM NUMBER SEED FOR DISTRIBUTION FUNCTIONS = 0

CLOCK TIME REL 2428181 ABS 2428181

TRANS COUNTS

BLOCK	TRANS,	TOTAL	BLOCK	TRANS,	TOTAL	BLOCK	TRANS,	TOTAL	BLOCK	TRANS,	TOTAL	BLOCK	TRANS,	TOTAL
1	0.	1	2	0.	1	3	0.	1	4	0.	1	5	0.	1
6	0.	1	7	0.	1	8	0.	1	9	0.	1	10	0.	1
11	0.	1	12	0.	1	13	0.	1	14	0.	1	15	0.	1
16	0.	1	17	0.	1	18	1.	504	19	0.	503	20	0.	503
21	0.	503	22	0.	503	23	0.	503	24	0.	503	25	0.	503
26	0.	503	27	0.	503	28	0.	503	29	0.	503	30	0.	503
31	0.	503	32	0.	503	33	0.	503	34	1.	503	35	0.	502
36	0.	502	37	0.	503	38	0.	503	39	0.	503	40	0.	503
41	0.	503	42	0.	503	43	0.	503	44	0.	503	45	0.	6
46	0.	6	47	0.	498	48	0.	498	49	0.	497	50	0.	0
51	0.	497	52	0.	497	53	0.	497	54	0.	497	55	0.	497
56	0.	497	57	0.	497	58	0.	497	59	0.	497	60	0.	497
61	0.	497	62	0.	497	63	0.	497	64	0.	497	65	0.	497
66	0.	497	67	0.	497	68	1.	497	69	0.	496	70	0.	496
71	0.	345	72	0.	151	73	0.	0	74	0.	345	75	0.	151
76	0.	0	77	0.	496	78	0.	496	79	0.	496	80	0.	496
81	0.	496	82	0.	496	83	0.	496	84	0.	0	85	0.	0
86	0.	496	87	0.	496	88	0.	496	89	0.	496	90	0.	504
91	0.	504	92	0.	504	93	0.	504	94	0.	504	95	0.	484
96	0.	477	97	0.	477	98	0.	7	99	0.	7	100	0.	20
101	0.	12	102	0.	8	103	0.	8	104	0.	0	105	0.	0
106	0.	0	107	0.	0	108	0.	0	109	0.	0	110	0.	8
111	0.	0	112	0.	8	113	0.	0	114	0.	0	115	0.	8
116	0.	8	117	0.	8	118	0.	484	119	1.	484	120	0.	484
121	0.	484	122	0.	484	123	0.	484	124	1.	484	125	0.	483
126	0.	483	127	1.	967	128	0.	483	129	0.	978	130	0.	495
131	0.	495	132	0.	495	133	0.	495	134	0.	495	135	0.	495
136	0.	484	137	0.	484	138	0.	484	139	0.	484	140	0.	483
141	0.	483	142	0.	484	143	0.	484	144	0.	484	145	0.	484
146	0.	484	147	0.	484	148	0.	484	149	0.	484	150	0.	484
151	0.	484	152	0.	484	153	0.	484	154	0.	484	155	0.	484
156	0.	484	157	0.	484	158	0.	484	159	0.	484	160	0.	484
161	0.	484	162	0.	484	163	0.	484	164	0.	484	165	0.	484
166	0.	484	167	1.	484	168	0.	126	169	0.	126	170	0.	483
171	0.	483	172	0.	483	173	0.	483	174	0.	483	175	0.	483
176	0.	483	177	0.	483	178	0.	483	179	0.	483	180	0.	484
181	0.	484	182	0.	484	183	1.	484	184	0.	483	185	0.	483
186	0.	483	187	0.	483	188	0.	483	189	0.	483	190	0.	483
191	0.	483	192	0.	483	193	0.	483	194	0.	483	195	0.	483
196	1.	483	197	0.	124	198	0.	124	199	0.	482	200	0.	482
201	0.	482	202	0.	482	203	0.	482	204	0.	482	205	0.	482
206	0.	482	207	0.	482	208	0.	482	209	0.	482	210	0.	12

211	0,	12	212	0,	12	213	0,	12	214	0,	12	215	0,	12
216	0,	12	217	0,	12	218	0,	12	219	0,	12	220	0,	12
221	0,	12	222	0,	12	223	0,	12	224	0,	12	225	0,	12
226	0,	12	227	0,	12	228	0,	12	229	0,	12	230	0,	12
231	0,	12	232	0,	12	233	0,	12	234	0,	12	235	0,	12
236	0,	12	237	0,	12	238	0,	12	239	0,	12	240	0,	12
241	0,	12	242	0,	12	243	0,	12	244	0,	12	245	0,	12
246	0,	12	247	0,	12	248	0,	12	249	0,	12	250	0,	6
251	0,	5	252	0,	0	253	0,	0	254	0,	5	255	0,	1
256	0,	0	257	0,	0	258	0,	1	259	0,	0	260	0,	501
261	0,	501	262	0,	501	263	0,	501	264	0,	501	265	0,	501
266	0,	1	267	0,	1	268	0,	1	269	0,	1	270	0,	1
271	0,	1	272	0,	1	273	0,	1	274	0,	1	275	0,	1
276	0,	501	277	0,	0	278	0,	0	279	0,	0	280	0,	0

SAVEX NR	........VALUE	NR	........VALUE	NR	........VALUE	NR	........VALUE	NR	........VALUE
1	497	2	0	3	5	4	0	5	6
6	503	7	1	8	1	9	0	10	2549
11	3280	12	1	13	4577	14	0	15	347
16	3280	17	1	18	4577	19	0	20	347
21	2431322	22	12	23	0	24	0	25	0
26	7	27	1	28	477	29	8	30	484

FACILITY NR	AVERAGE UTILIZATION	NUMBER ENTRIES	AVERAGE TIME/TRANS	TRANS	$TRANS
1	.6875	345	4838.94	12,S	0
2	.3000	151	4824.56	0	0

STORAGE NR	MAXIMUM CONTENTS	CAPACITY	AVERAGE CONTENTS	AVERAGE UTILIZATION	TOTAL ENTRIES	TOTAL TRANS	AVERAGE ENT/TRAN	AVERAGE TIME/ENTRY	CURRENT CONTENTS
1	1	1	.20	.1988	484	484	1.00	997.31	0
2	1	1	.58	.5773	496	496	1.00	2825.98	1
3	1	2	.41	.2039	495	495	1.00	2000.08	0
4	4	9	1.62	.1805	499	496	1.01	7902.79	2
10	1	1	.20	.1993	483	483	1.00	1002.05	0
11	2	2	.60	.2998	495	495	1.00	2941.18	1
12	2	2	.41	.2038	494	494	1.00	2003.78	0
13	5	9	1.99	.2206	499	495	1.01	9660.57	2
14	3	9	.68	.0758	484	484	1.00	3424.07	1
16	3	10	.98	.0983	498	496	1.00	4794.66	1
17	2	2	.30	.1495	484	484	1.00	1499.90	1
18	2	2	.61	.3058	495	495	1.00	3000.35	0
19	1	1	.00	.0038	12	12	1.00	765.83	0
20	1	1	.41	.4094	497	497	1.00	2000.43	1
22	2	2	.41	.2064	496	496	1.00	2021.03	0
23	1	2	.02	.0086	504	504	1.00	83.00	0
25	1	2	.01	.0049	12	12	1.00	1966.58	0
27	1	2	.20	.0995	484	484	1.00	998.12	0
28	1	2	.00	.0002	484	484	1.00	2.00	0
29	1	1	.12	.1249	503	503	1.00	603.16	0
30	1	1	.00	.0037	12	12	1.00	742.50	0
31	1	1	.01	.0060	503	503	1.00	28.86	0

QUEUE NR	MAXIMUM CONTENTS	AVERAGE CONTENTS	TOTAL ENTRIES	ZERO ENTRIES	ZEROS PERCENT	AVERAGE TIME/ENTRIES ALL ENT	AVERAGE TIME/ENTRIES NON ZERO ENT	TABLE NUMBER	CURRENT CONTENTS
1	1	.00	484	484	100.00	.00	.00	0	0

2	3	.05	496	449	90.52	254.53	2686.09	0	0
3	1	.00	495	495	100.00	.00	.00	0	0
10	1	.00	483	464	96.07	15.54	395.00	0	0
11	1	.00	495	492	99.39	.96	159.00	0	0
12	1	.00	494	494	100.00	.00	.00	0	0
18	1	.00	495	494	99.80	.00	.00	0	0
20	1	.00	497	495	99.60	2.54	631.50	0	0
21	1	.00	496	496	100.00	.00	.00	0	0
29	1	.00	503	503	100.00	.00	.00	0	0
31	1	.00	503	503	100.00	.00	.00	0	0

TABLE NUMBER 1

ENTRIES IN TABLE 483   MEAN ARGUMENT 6121.199   STANDARD DEVIATION 1634.808   NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
5000	0	.00	.0	100.0	.817	-.686
5500	74	15.32	15.3	84.7	.899	-.380
6000	264	54.66	70.0	30.0	.980	-.074
6500	63	13.04	83.0	17.0	1.062	.232
7000	39	8.07	91.1	8.9	1.144	.538
7500	18	3.73	94.8	5.2	1.225	.843
8000	10	2.07	96.9	3.1	1.307	1.149
8500	0	.00	96.9	3.1	1.389	1.455
9000	4	.83	97.7	2.3	1.470	1.761
9500	1	.21	97.9	2.1	1.552	2.067
10000	0	.00	97.9	2.1	1.634	2.373
10500	1	.21	98.1	1.9	1.715	2.678
11000	2	.41	98.6	1.4	1.797	2.984
11500	0	.00	98.6	1.4	1.879	3.290
12000	1	.21	98.8	1.2	1.960	3.596
12500	0	.00	98.8	1.2	2.042	3.902
13000	0	.00	98.8	1.2	2.124	4.208
13500	0	.00	98.8	1.2	2.205	4.514
14000	0	.00	98.8	1.2	2.287	4.819
OVERFLOW	6	1.24	100.0	.0		

TABLE NUMBER 2

ENTRIES IN TABLE 497   MEAN ARGUMENT 7926.986   STANDARD DEVIATION 3142.907   NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-2.522
500	2	.40	.4	99.6	.063	-2.363
1000	3	.60	1.0	99.0	.126	-2.204
1500	6	1.21	2.2	97.8	.189	-2.045
2000	8	1.61	3.8	96.2	.252	-1.886
2500	4	.80	4.6	95.4	.315	-1.727

3000	8	1.61	6.2	93.8	.378	-1.568
3500	13	2.62	8.9	91.1	.442	-1.409
4000	6	1.21	10.1	89.9	.505	-1.249
4500	20	4.02	14.1	85.9	.568	-1.090
5000	18	3.62	17.7	82.3	.631	-.931
5500	25	5.03	22.7	77.3	.694	-.772
6000	27	5.43	28.2	71.8	.757	-.613
6500	21	4.23	32.4	67.6	.820	-.454
7000	27	5.43	37.8	62.2	.883	-.295
7500	26	5.23	43.1	56.9	.946	-.136
8000	37	7.44	50.5	49.5	1.009	.023
8500	25	5.03	55.5	44.5	1.072	.182
9000	34	6.84	62.4	37.6	1.135	.341
9500	28	5.63	68.0	32.0	1.198	.500
10000	32	6.44	74.4	25.6	1.262	.660
10500	24	4.83	79.3	20.7	1.325	.819
11000	25	5.03	84.3	15.7	1.388	.978
11500	18	3.62	87.9	12.1	1.451	1.137
12000	19	3.82	91.8	8.2	1.514	1.296
12500	12	2.41	94.2	5.8	1.577	1.455
13000	5	1.01	95.2	4.8	1.640	1.614
13500	6	1.21	96.4	3.6	1.703	1.773
14000	5	1.01	97.4	2.6	1.766	1.932
OVERFLOW	13	2.62	100.0	.0		

TABLE NUMBER 3

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
483	19392.867	3435.608	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
9000	0	.00	.0	100.0	.464	-3.025
9500	0	.00	.0	100.0	.490	-2.880
10000	0	.00	.0	100.0	.516	-2.734
10500	0	.00	.0	100.0	.541	-2.588
11000	0	.00	.0	100.0	.567	-2.443
11500	1	.21	.2	99.8	.593	-2.297
12000	3	.62	.8	99.2	.619	-2.152
12500	2	.41	1.2	98.8	.645	-2.006
13000	6	1.24	2.5	97.5	.670	-1.861
13500	5	1.04	3.5	96.5	.696	-1.715
14000	8	1.66	5.2	94.8	.722	-1.570
14500	15	3.11	8.3	91.7	.748	-1.424
15000	8	1.66	9.9	90.1	.773	-1.279
15500	15	3.11	13.0	87.0	.799	-1.133
16000	12	2.48	15.5	84.5	.825	-.988
16500	18	3.73	19.3	80.7	.851	-.842
17000	22	4.55	23.8	76.2	.877	-.696
17500	26	5.38	29.2	70.8	.902	-.551
18000	21	4.35	33.5	66.5	.928	-.405
18500	20	4.14	37.7	62.3	.954	-.260
19000	42	8.70	46.4	53.6	.980	-.114
19500	24	4.97	51.3	48.7	1.006	.031

20000	35	7.25	58.6	41.4	1.031	.177
20500	17	3.52	62.1	37.9	1.057	.322
21000	31	6.42	68.5	31.5	1.083	.468
21500	31	6.42	74.9	25.1	1.109	.613
22000	32	6.63	81.6	18.4	1.134	.759
22500	17	3.52	85.1	14.9	1.160	.904
23000	14	2.90	88.0	12.0	1.186	1.050
OVERFLOW	58	12.01	100.0	.0		

TABLE NUMBER 4

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
496	1207.008	225.094	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-5.362
100	0	.00	.0	100.0	.083	-4.918
200	0	.00	.0	100.0	.166	-4.474
300	1	.20	.2	99.8	.249	-4.029
400	0	.00	.2	99.8	.331	-3.585
500	2	.40	.6	99.4	.414	-3.141
600	2	.40	1.0	99.0	.497	-2.697
700	5	1.01	2.0	98.0	.580	-2.252
800	10	2.02	4.0	96.0	.663	-1.808
900	25	5.04	9.1	90.9	.746	-1.364
1000	38	7.66	16.7	83.3	.828	-.920
1100	58	11.69	28.4	71.6	.911	-.475
1200	89	17.94	46.4	53.6	.994	-.031
1300	97	19.56	65.9	34.1	1.077	.413
1400	80	16.13	82.1	17.9	1.160	.857
1500	43	8.67	90.7	9.3	1.243	1.302
1600	30	6.05	96.8	3.2	1.326	1.746
1700	12	2.42	99.2	.8	1.408	2.190
1800	4	.81	100.0	.0	1.491	2.634

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 5

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
8	681.625	246.855	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-2.761
100	0	.00	.0	100.0	.147	-2.356
200	0	.00	.0	100.0	.293	-1.951
300	1	12.50	12.5	87.5	.440	-1.546
400	0	.00	12.5	87.5	.587	-1.141
500	0	.00	12.5	87.5	.734	-.736
600	3	37.50	50.0	50.0	.880	-.331
700	0	.00	50.0	50.0	1.027	.074

800	0	.00	50.0	50.0	1.174	.480
900	0	.00	50.0	50.0	1.320	.885
1000	4	50.00	100.0	.0	1.467	1.290

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 6

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
497	9.835	82.668	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	489	98.39	98.4	1.6	.000	-.119
100	0	.00	98.4	1.6	10.168	1.091
200	0	.00	98.4	1.6	20.336	2.300
300	2	.40	98.8	1.2	30.503	3.510
400	0	.00	98.8	1.2	40.671	4.720
500	0	.00	98.8	1.2	50.839	5.929
600	2	.40	99.2	.8	61.007	7.139
700	0	.00	99.2	.8	71.174	8.349
800	2	.40	99.6	.4	81.342	9.558
900	1	.20	99.8	.2	91.510	10.768
1000	1	.20	100.0	.0	101.678	11.978

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 8

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
7	731.000	246.165	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-2.970
100	0	.00	.0	100.0	.137	-2.563
200	0	.00	.0	100.0	.274	-2.157
300	0	.00	.0	100.0	.410	-1.751
400	1	14.29	14.3	85.7	.547	-1.345
500	0	.00	14.3	85.7	.684	-.938
600	3	42.86	57.1	42.9	.821	-.532
700	0	.00	57.1	42.9	.958	-.126
800	0	.00	57.1	42.9	1.094	.280
900	0	.00	57.1	42.9	1.231	.687
1000	3	42.86	100.0	.0	1.368	1.093

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 9

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
484	1.014	.119	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-8.497
1	477	98.55	98.6	1.4	.986	-.121
2	7	1.45	100.0	.0	1.971	8.255

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 10

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
484	5325.473	125.264	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	0	.00	.0	100.0	.000	-42.514
500	0	.00	.0	100.0	.094	-38.522
1000	0	.00	.0	100.0	.188	-34.531
1500	0	.00	.0	100.0	.282	-30.539
2000	0	.00	.0	100.0	.376	-26.548
2500	0	.00	.0	100.0	.469	-22.556
3000	0	.00	.0	100.0	.563	-18.565
3500	0	.00	.0	100.0	.657	-14.573
4000	0	.00	.0	100.0	.751	-10.581
4500	1	.21	.2	99.8	.845	-6.590
5000	5	1.03	1.2	98.8	.939	-2.598
5500	471	97.31	98.6	1.4	1.033	1.393
6000	4	.83	99.4	.6	1.127	5.385
6500	3	.62	100.0	.0	1.221	9.376

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 21

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
482	5872.456	498.013	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
5000	0	.00	.0	100.0	.851	-1.752
5500	60	12.45	12.4	87.6	.937	-.748
6000	294	61.00	73.4	26.6	1.022	.256
6500	73	15.15	88.6	11.4	1.107	1.260
7000	35	7.26	95.9	4.1	1.192	2.264
7500	12	2.49	98.3	1.7	1.277	3.268
8000	5	1.04	99.4	.6	1.362	4.272
8500	3	.62	100.0	.0	1.447	5.276

REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER 22

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION

497	9673.145		3591.861	NON-WEIGHTED		

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
0	1	.20	.2	99.8	.000	-2.693
500	0	.00	.2	99.8	.052	-2.554
1000	0	.00	.2	99.8	.103	-2.415
1500	2	.40	.6	99.4	.155	-2.275
2000	1	.20	.8	99.2	.207	-2.136
2500	4	.80	1.6	98.4	.258	-1.997
3000	7	1.41	3.0	97.0	.310	-1.858
3500	5	1.01	4.0	96.0	.362	-1.719
4000	7	1.41	5.4	94.6	.414	-1.579
4500	9	1.81	7.2	92.8	.465	-1.440
5000	12	2.41	9.7	90.3	.517	-1.301
5500	12	2.41	12.1	87.9	.569	-1.162
6000	17	3.42	15.5	84.5	.620	-1.023
6500	21	4.23	19.7	80.3	.672	-.883
7000	25	5.03	24.7	75.3	.724	-.744
7500	14	2.82	27.6	72.4	.775	-.605
8000	28	5.63	33.2	66.8	.827	-.466
8500	23	4.63	37.8	62.2	.879	-.327
9000	26	5.23	43.1	56.9	.930	-.187
9500	18	3.62	46.7	53.3	.982	-.048
10000	35	7.04	53.7	46.3	1.034	.091
10500	29	5.84	59.6	40.4	1.085	.230
11000	28	5.63	65.2	34.8	1.137	.369
11500	26	5.23	70.4	29.6	1.189	.509
12000	22	4.43	74.8	25.2	1.241	.648
12500	16	3.22	78.1	21.9	1.292	.787
13000	24	4.83	82.9	17.1	1.344	.926
13500	14	2.82	85.7	14.3	1.396	1.065
14000	18	3.62	89.3	10.7	1.447	1.205
OVERFLOW	53	10.66	100.0	.0		

TABLE NUMBER 23

ENTRIES IN TABLE	MEAN ARGUMENT	STANDARD DEVIATION	
482	20915.421	3612.657	NON-WEIGHTED

UPPER LIMIT	OBSERVED FREQUENCY	PERCENT OF TOTAL	CUMULATIVE PERCENTAGE	CUMULATIVE REMAINDER	MULTIPLE OF MEAN	DEVIATION FROM MEAN
9000	0	.00	.0	100.0	.430	-3.298
9500	0	.00	.0	100.0	.454	-3.160
10000	0	.00	.0	100.0	.478	-3.021
10500	0	.00	.0	100.0	.502	-2.883
11000	1	.21	.2	99.8	.526	-2.745
11500	0	.00	.2	99.8	.550	-2.606
12000	0	.00	.2	99.8	.574	-2.468
12500	2	.41	.6	99.4	.598	-2.329
13000	3	.62	1.2	98.8	.622	-2.191
13500	4	.83	2.1	97.9	.645	-2.053
14000	4	.83	2.9	97.1	.669	-1.914
14500	6	1.24	4.1	95.9	.693	-1.776

15000	4	.83	5.0	95.0	.717	-1.637
15500	7	1.45	6.4	93.6	.741	-1.499
16000	7	1.45	7.9	92.1	.765	-1.361
16500	14	2.90	10.8	89.2	.789	-1.222
17000	15	3.11	13.9	86.1	.813	-1.084
17500	17	3.53	17.4	82.6	.837	-.945
18000	20	4.15	21.6	78.4	.861	-.807
18500	17	3.53	25.1	74.9	.885	-.669
19000	23	4.77	29.9	70.1	.908	-.530
19500	25	5.19	35.1	64.9	.932	-.392
20000	25	5.19	40.2	59.8	.956	-.253
20500	22	4.56	44.8	55.2	.980	-.115
21000	29	6.02	50.8	49.2	1.004	.023
21500	29	6.02	56.8	43.2	1.028	.162
22000	22	4.56	61.4	38.6	1.052	.300
22500	29	6.02	67.4	32.6	1.076	.439
23000	21	4.36	71.8	28.2	1.100	.577
OVERFLOW	136	28.22	100.0	.0		

FUTURE RANDOM NUMBER SEED IS (OCTAL) 277404621561

APPENDIX E. SAMPLE CONFIDENCE INTERVAL OPTION OUTPUT

ooo TABLE III. STATISTICAL RESULTS ooo

CONFIDENCE INTERVALS ON REQUESTED STATISTICS

WARNING - CONFIDENCE INTERVALS WITH LIMITS OF 0 OR 100 ARE PROBABLY NOT 99 PERCENT CONFIDENCE INTERVALS

STATISTICS BASED ON 8 OBSERVATIONS.

	99 PERCENT CONFIDENCE INTERVALS		
	LOWER LIMIT	MEAN	UPPER LIMIT
STATISTICAL CONFIDENCE INTERVALS FOR 500 SIMULATIONS.			
THE PERCENTAGE OF SUCCESSFUL LAUNCH REQUESTS	38.06	41.50	44.94
THE PERCENTAGE OF UNSUCCESSFUL LAUNCH REQUESTS	55.81	58.75	61.69
THE PERCENTAGE OF VEHICLES LAUNCHED AT A SUBSEQUENT WINDOW	3.37	7.50	11.63
REASONS FOR REQUEST FAILURES:			
BOOSTER UNAVAILABILITY	.00	.00	.00
ORBITER UNAVAILABILITY	.00	.00	.00
BOTH STAGES UNAVAILABLE	.00	.00	.00
LUT UNAVAILABILITY	55.69	57.75	59.81
PAD UNAVAILABILITY	.00	.00	.00
FAILURES AND REMOVAL FROM PAD	.00	1.00	2.32
THE FOLLOWING ARE THE USER REQUESTED STATISTICS			
TIME FROM LANDING TO STORAGE (BOOSTER)	5.69	5.87	6.05
ORBITER SAFING FACILITY UTILIZATION (PERCENT)	11.28	12.84	14.40
LAUNCH PAD 1 UTILIZATION (PERCENT)	40.33	42.41	44.48
BOOSTER SAFING QUEUE TIME	.00	.00	.00
ORBITER MAINTENANCE QUEUE TIME	.00	.00	.00

APPENDIX F. EXPLANATION OF INDIVIDUAL CONTROL CARDS

This appendix gives a brief operational description of the various control cards contained in the example deck setups shown in the document. Further information about these control statements is given in the MSFC Program Procedures Manual with the page numbers in parentheses. (Refer to Figure F-1 for the correct column positioning).

1. @ADD MODEL.
   This card will add the contents of file MODEL to the run stream. This card must follow the @XQT GPSSE8. MAPGPS card for each execution of the GPSS program. (6.3.1-80)

2. @ASG, T NAME, F2
   This control card assigns a temporary file location called NAME on a magnetic drum unit. The F2 represents the type of drum selected for mass storage. Note that an @ASG card must be used for each file referenced within the run. (6.3.1-29)

3. @ASG, T TAPENAME, T, SAVEXX
   This card creates a tape called TAPENAME and is maintained in the MSFC tape library for the time specified by the code XX. The code XX represents the following: (Technical Bulletin No. 17)

   02--14 day retention
   03--30 day retention
   04--90 day retention
   05--6 month retention

4. @ASG, T TAPENAME, T, 12345
   This card assigns the use of a tape drive to the run. In this example, the tape file is called TAPENAME and reel number 12345 is loaded onto the drive unit. (6.3.1-33)

5. @BRKPT PUNCH$
   This card causes the current normal punch file to be closed and queued for punching and starts a new normal punch file. (6.3.1-101)

6. @BRKPT PUNCH$/FN
   This statement causes the current normal punch file to be closed and queued for printing and routes all future output of the normal punch type to a file whose name is FN. The file whose name is FN must be currently assigned to the run. (6.3.1-102)

7. @COPIN TAPENAME, GPSSE8.
   This card causes the transfer of the program instruction from the tape called TAPENAME, to a file name GPSSE8. (6.3.1-147)

8. @COPIN TAPENAME, TPF$.
   This card causes the transfer of the program instructions from the tape, called TAPENAME, to a temporary program file. (6.3.1-148)

9. @COPOUT GPSSE8, TAPENAME
   This card is a reverse operation of the @COPIN card. The program instructions are transferred from the file GPSSE8 to the tape named TAPENAME. (6.3.1-146)

10. @COPY,G TAPENAME, MODEL.
    This control card causes the transfer of a data file (the GPSS System Model Description) from the tape named TAPENAME to a drum location named MODEL. (6.3.1-144)

11. @COPY,GM MODEL, TAPENAME
    This represents the reverse operation of the @COPY,G TAPENAME, MODEL. command. (6.3.1-144)

12. @COPY,RSA MASTER*GPSSE8., GPSSE8.
    This control card is used to command execution from a drum file. This card copies from the drum a program called MASTER*GPSSE8 into temporary location named GPSSE8. (6.3.1-144)

13. @DATA FILE2, MODEL
    This control card permits the updating of the GPSS model network contained on a file named FILE2. The new revision is stored in a file named MODEL (corrections or changes to MODEL follow this control card and FILE2 remains unchanged). (6.3.1-97)

14. @DATA,L DUMMYP, MODEL
    This control allows the revision of the DUMMYP file and creates an updated file named MODEL. The revisions to DUMMYP follow this control card. A listing of the data in MODEL is then generated. (6.3.1-97).

15. @END
This control card marks the end of the data corrections following the DATA control card. (6.3.1-98)

16. @ERS GPSSE8.
This control card will erase all contents of the program file named GPSSE8. (This card is used in decks that generate a new program file.) (6.3.1-156)

17. @FIN
Signifies the end of the control deck. (This is the last card on all deck setups.) (6.3.1-80)

18. @FOR, US GPSSE8. NAME, GPSSE8. NAME
Control card used to include a new or modified Fortran user written subroutine in the GPSS run stream. The word NAME should be replaced on the control card with the title of the subroutine. The U option allows modification of individual cards in the subroutine. An I option allows the inclusion of an entire subroutine package but not the updating of individual cards (6.3.1-103)

19. @FREE TAPENAME
Releases the tape called TAPENAME. (In cases where the information on the tape has been copied onto the drums, the tape drive unit is no longer needed and, therefore, released from the run.) (6.3.1-88)

20. @HDG, P CASE STUDY TITLE
This heading card prints any desired title on the top of each page of the printout. The page number and data are also printed. All columns on the card can be used. In this case, the title CASE STUDY TITLE is printed on top of each page of output. (6.3.1-124)

21. @MAP, LX GPSSE8. MAP, GPSSE8. MAPGPS
This card causes the collection of all program elements into the program instruction element by the name of MAPGPS. (6.6-4)

22. @MOVE TAPENAME, 2
This control card is used for positioning the tape called TAPENAME at the beginning of the selected file. (The example indicates the movement of the tape past 2 files from its present position.) (6.3.1-137)

23. @PREP GPSSE8.
This command is used to prepare a program file on FASTRAND for subsequent referencing as a library by the collector. This card must always be used before a @MAP control card. (6.3.1-153)

24. @PRT,T
This card produces a table of contents of the temporary program file (TPF$). (6.3.1-157)

25. @PRT,T GPSSE8.
This card will generate a table of contents of all elements contained in the program file GPSSE8. (6.3.1-151)

26. @REWIND TAPENAME
Executive command which causes the rewinding of the tape drive called TAPENAME. Execution of this control causes the tape to be positioned at the starting point of the first tape file. (6.3.1-138)

27. @RUN, //P SAMPLE, 999999, PORTERBIN225, 3, 150

The run is always the first control card in the 1108 deck setup. This card contains all the information used by the computer for accounting and identification purposes. The three options available on the RUN card are P (production), T (development, and A (maintenance and checkout). SAMPLE1 is the run I.D. code assigned to the particular deck for use by the computer, 999999 is the accounting job number, PORTER is the programmer's name in six characters, and 225 identifies the programmer's BIN location. The 3 is the maximum CPU time (min) for the run. The 150 represents the maximum number of pages of output produced by the program. (Technical Bulletin No. 16)

28. @SETC 2
Sets the computer condition word to the value of 2 and causes the generation of a punched deck of the absolute GPSS model. (6.3.1-118)

29. @SETC 3
Sets the computer condition word to the value of 3, thus eliminating the listing of the model network. A separate header page is generated for each case to be executed. (6.3.1-118)

30. @SETC 4
Sets the computer condition word to the value of 4, thus causing a printout indicating the distributions and their associated parameters from the GPSS model that were defined for each processing operation. (An extra control card, @ADD MODEL, must be inserted into the program deck when this control option is used.) (6.3.1-118)

31. @SETC 5
Sets the computer condition word to the value of 5, thus causing the System Model to be read for headings used in the collection of confidence interval statistics. (An extra contral card, @ADD MODEL, must be inserted into the program deck when this control option is used.) (6.3.1-118)

32. @XQT GPSSE8.MAPGPS
Control card command indicating the execution of the GPSS program instructions called MAPGPS contained on a file named GPSSE8. (6.3.1-108)

33. @XQT,LA SYS$*MSFC$.LISTIT
This card executes a special program that generates a listing of the contents of the temporary program file. (6.3.1-106)

GENERAL CARD DESIGN

1	@ADD MODEL.
2	@ASG,T NAME,F2
3	@ASG,T TAPENAME,T,SAVEXX
4	@ASG,T TAPENAME,T,12345
5	@BRKPT PUNCH$
6	@BRKPT PUNCH$/FN
7	@COPIN TAPENAME,GPSSE8.
8	@COPIN TAPENAME,TPF$.
9	@COPOUT GPSSE8,TAPENAME
10	@COPY,G TAPENAME,MODEL.
11	@COPY,GM MODEL,TAPENAME
12	@COPY,RSA MASTER*GPSSE8.,GPSSE8.
13	@DATA FILE2,MODEL
14	@DATA,L DUMMYP.MODEL
15	@END
16	@ERS GPSSE8.
17	@FIN
18	@FOR,US GPSSE8.NAME,GPSSE8.NAME
19	@FREE TAPENAME
20	@HDG,F CASE STUDY TITLE
21	@MAP,LX GPSSE8.MAP,GPSSE8.MAPGPS
22	@MOVE TAPENAME,2
23	@PREP GPSSE8.
24	@PRT,T
25	@PRT,T GPSSE8.

MSFC - Form 353 (July 1960)

FIGURE F-1. EXAMPLE CONTROL CARD FORMAT (Sheet 1 of 2)

GENERAL CARD DESIGN

	Columns 1–80
26	@REWIND TAPENAME
27	@RUN,//P SAMPLE,99999,PORTERBIN225,3,150
28	@SETC 2
29	@SETC 3
30	@SETC 4
31	@SETC 5
32	@XQT GPSSE8.MAPGPS
33	@XQT,LA SYS$*MSFC$.LISTIT

MSFC - Form 353 (July 1960)

MSFC—RSA, Ala

FIGURE F-1. EXAMPLE CONTROL CARD FORMAT (Sheet 2 of 2)