MEMORANDUM
RM-4663-PR
JANUARY 1969

PLANET: PART V-
REPORTS AND ANALYSIS LIBRARY
B. J. Voosen, S. Glaseman, R. J. Young and Judy Judd

This research is supported by the United States Air Force under Project RAND Con-
tact No. F1162-67-C-0015 monitored by the Directorate of Operational Requirements
and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF.
Views or conclusions contained in this study should not be interpreted as representing
the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

The RAND Corporation
This study is presented as a competent treatment of the subject, worthy of publication. The Rand Corporation vouches for the quality of the research, without necessarily endorsing the opinions and conclusions of the authors.

Published by The RAND Corporation
PREFACE AND SUMMARY

PLANET (Planned Logistics Analysis and Evaluation Technique) is a series of four computer simulation models designed to examine the hardware-configuration/operations/logistics support interactions of a variety of weapon systems in a single or multibase environment. Its purpose is to help the manager gain an understanding of the operation of his system and find a rationale for allocating resources effectively and efficiently.

The PLANET complex comprises five computer programs:

1) The Availability and Base Cadre Simulator (ABC) furnishes the framework for the logistics resources assigned to a support base or bases.

2) The Bench Repair Simulator (BR) processes the reparables through the base repair shops or diverts them to a depot, thus converting the reparables to serviceables.

3) The Depot Transportation Simulator (DT) processes the movement of reparables from the base(s) to the depot(s) or factory and return.

4) The Depot Repair and Overhaul Simulator (DR&O) simulates the functions in a repair or overhaul facility.

5) The Reports and Analysis Library consists of twelve different output programs.

The simulators can be used separately to examine specific areas of the logistics system, or conjointly to simulate the complete weapon-system operation from the site or point of demand through to the depot.

The Reports and Analysis Library described here contains the twelve report programs with operating instructions. All reports are designed for use by managerial personnel. The manager may select from the library those programs best suited for analysis of his particular problem. Even though the output programs cover a wide spectrum of problem areas, it can be anticipated that additional outputs will in some cases be required. Either the output programs can be modified to incorporate any additional data required, or new programs can be written with relative ease.
Although PLANET is designed as an advanced planning tool, it can be used to assess periodically whether the logistics support planned will maintain a system or equipment effectively and economically. These assessments combine the relevant reliability, maintainability, and performance parameters for the weapon system. This enables the systematic development of an integrated logistics support plan for systems and equipment at all levels of maintenance for its programmed life cycle.

While PLANET is programmed in SIMSCRIPT, the user need not be a skilled SIMSCRIPT programmer to conduct a simulation. We have included the necessary step-by-step instructions as well as the necessary SIMSCRIPT instructions to permit managers to assemble the data in a form acceptable to the models.
In general, computer simulation is a way of using a computer to produce a reasonable likeness of the behavior of a system under study. Simulation models are only representations of reality. Of necessity, the likeness of the system under study is "scaled down" to manageable size for the computer. Simulation models, therefore, are based on the designer’s concept of what the key elements of the system are, and how they operate and interact on the system.

This being so, one cannot say a priori that one model is better than another. A manager should always strive for the lowest-cost model that suits his purpose. Since computer simulation models generally are explanatory, the analyst must first determine if a particular model sufficiently represents his system. In short, the analyst must first understand the model and then strike a compromise between realism and simplicity.

The size and complexity of the problem that the system manager would like to examine increase as a function of the interrelationships to be considered. It is very difficult, for example, to visualize the impact of a change in sortie rate on the personnel requirements in a depot overhaul facility, or even the effect of a change in reliability/maintainability parameters on the operational capability if support shortages exist at the higher echelons of maintenance. In short, while problems can be bounded and scaled down to manageable size, it is often desirable to view the analysis in a broader context to observe more of the interactions.

PLANET was developed as a logistics prediction and estimating tool. Its purpose is to help the manager gain an understanding of the operation of his system and find a rationale for allocating resources efficiently. Real world observations of a system help serve the same end, but simulations permit more varied, controlled and complete ranges of experience, usually at far less cost and much earlier in the life of the weapon system.

Coincident with the PLANET development program, DOD Directive 4100.35 dated June 19, 1964 was issued. The following is quoted from
that directive:

The primary objective of this Directive is to assure that the development of effective logistic support for systems and equipments is systematically planned, acquired, and managed as an integrated whole (by interlocking the elements of logistic support) to obtain maximum material readiness and optimum cost effectiveness.

Integrated Logistic Support - Integrated Logistic support is a composite of the elements necessary to assure the effective and economical support of a system or equipment at all levels of maintenance for its programmed life cycle. It is characterized by the harmony and coherence obtained between each of its elements and levels of maintenance.

We believe that simulation models such as PLANET can be used to develop an "Integrated Logistics Support" plan for a spectrum of weapon systems.

The history of the development of PLANET might be of interest. Prior to the actual coding of the programs, approximately one man-year was devoted to the problem of how best to structure the models. It was obvious that the bulk of the computers envisioned by the projected release date (1966) would have memories of 32K words. Although larger computers were being proposed at the time, we had no guarantee that very large computers (greater than 96K) would be readily available to prospective users. Therefore the problem of how to structure the programs to be useful, regardless of computer size, had to be faced.

In addition, we wanted to structure the simulation programs so that the internal logic of the simulator could be modified for special applications with relative ease. This required that the family of weapon systems that the model is designed to imitate be as broad as possible while the program itself be segmented into as many small subroutines as practicable.

The result of the planning phase was that the simulators would be bounded in a logical order of Flightline or site, Base, Depot and a link between Base and Depot. Each segment of the total simulation package must be capable of being used on a machine with the limited capacity of 32K. In some instances, this constraint limited the amount of detail we would have liked to include. In addition, since there
appears to be a trend toward the procurement of larger (larger than 32K) machines, the models should be easily assembled into larger, more detailed simulation programs.

After many months of examining logistic systems and plans, a commonality among them appeared to emerge that indicated the feasibility of such an undertaking. Although in many instances (particularly in the comparison of aircraft and missile logistic systems) the jargon used to describe specific functions was completely different, the functions to be performed were similar. Even though the operating parameters for the simulation would be different, this meant that the same computer program logic could be used if the real world jargon could be defined into common terms.

We therefore have tried to use, as much as possible, the functional description of the various logistics actions and activities, and hope that users will be able to translate the jargon into functional terms for use in the simulation.

Coding of the programs began in January 1964. By October 1964 the first (ABC simulator) of the four simulators was available for debugging and proof testing.

For each simulator, debugging consists of tracing (using trace routines coded into the program) each event through its cycle during a simulation run.

The proof testing consisted of inputting a set of empirical data, computing each value that was to be generated by the computer by manual or analytical methods, then comparing the empirical inputs with the actual experience that occurred in the real world. This has been done for both missile and aircraft data sets.

By March 1967, all models were coded and proof testing of the last was nearing completion. There remained only the marriage of the four programs to ensure that they would in fact work together.

In total approximately nine man-years of coding effort was used to code, debug and proof test the models, and approximately 200 hours of computer operating time (this is in addition to the one man-year previously mentioned). It is easy to see, therefore, why some organizations would have difficulty in developing models of this size and complexity.
In conclusion, we hasten to point out that while PLANET is developed as a "general purpose simulation model," it is not a panacea. It is limited in purpose and scope. However, we have endeavored to structure the simulators so that as the need arises additional complexity can be added and the models thus can be expanded.
ACKNOWLEDGMENTS

To acknowledge everyone who contributed to PLANET would require many pages. We would feel negligent, however, if we did not acknowledge the efforts of all the programmers who developed the many programs contained in this Memorandum.

1. Miss Pat Love developed the Cost Effectiveness program (number 8).
2. Mr. Richard Villanueva developed the Bench Repair Capability and NRTS programs (numbers 9 and 10).
3. Mr. Al Nelson developed the Aircraft Recovery Package Report Generator (numbers 4 to 7).
4. Mr. Steven Glaseman developed the Depot Transportation output program (number 11).
5. Last but not least, Mrs. Shirley Ballinger developed the Depot Repair and Overhaul Capability output program and has been of great assistance during the early production runs of PLANET.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE AND SUMMARY</td>
<td>iii</td>
</tr>
<tr>
<td>FOREWORD</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>ix</td>
</tr>
<tr>
<td>PART I. INTRODUCTION AND INITIALIZATION INSTRUCTIONS</td>
<td>1</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>2</td>
</tr>
<tr>
<td>NOR Time Summary</td>
<td>2</td>
</tr>
<tr>
<td>Weapon System Availability</td>
<td>2</td>
</tr>
<tr>
<td>Logistics Resource Utilization</td>
<td>3</td>
</tr>
<tr>
<td>Aircraft Recovery Package</td>
<td>3</td>
</tr>
<tr>
<td>Cost/Effectiveness Program</td>
<td>4</td>
</tr>
<tr>
<td>Base Shops Maintenance Capability</td>
<td>5</td>
</tr>
<tr>
<td>NRTS Program</td>
<td>5</td>
</tr>
<tr>
<td>Transportation Capability</td>
<td>5</td>
</tr>
<tr>
<td>Depot Capability</td>
<td>5</td>
</tr>
<tr>
<td>II. Initialization</td>
<td>7</td>
</tr>
<tr>
<td>PART 2. PROGRAM LIBRARY</td>
<td>15</td>
</tr>
<tr>
<td>I. NOR Time Summary</td>
<td>16</td>
</tr>
<tr>
<td>Initialization</td>
<td>16</td>
</tr>
<tr>
<td>Output Program</td>
<td>16</td>
</tr>
<tr>
<td>Permanent Variables</td>
<td>19</td>
</tr>
<tr>
<td>Sets</td>
<td>21</td>
</tr>
<tr>
<td>II. Weapon System Availability</td>
<td>44</td>
</tr>
<tr>
<td>Initialization</td>
<td>49</td>
</tr>
<tr>
<td>Output Program</td>
<td>49</td>
</tr>
<tr>
<td>Permanent Variables</td>
<td>49</td>
</tr>
<tr>
<td>Sets</td>
<td>52</td>
</tr>
<tr>
<td>III. Logistics Resource Utilization</td>
<td>87</td>
</tr>
<tr>
<td>Initialization</td>
<td>89</td>
</tr>
<tr>
<td>Output Program</td>
<td>91</td>
</tr>
<tr>
<td>Permanent Variables</td>
<td>91</td>
</tr>
<tr>
<td>Sets</td>
<td>92</td>
</tr>
<tr>
<td>IV. Aircraft Recovery Time Distribution</td>
<td>165</td>
</tr>
<tr>
<td>V. System Recovery</td>
<td>172</td>
</tr>
<tr>
<td>VI. Work Center Recovery</td>
<td>179</td>
</tr>
<tr>
<td>The Recovery Program</td>
<td>179</td>
</tr>
<tr>
<td>Flight Program Description</td>
<td>181</td>
</tr>
<tr>
<td>VII. Failure List</td>
<td>186</td>
</tr>
<tr>
<td>VIII. Cost/Effectiveness</td>
<td>191</td>
</tr>
<tr>
<td>Initialization</td>
<td>191</td>
</tr>
<tr>
<td>Output Program</td>
<td>191</td>
</tr>
<tr>
<td>Permanent Variables</td>
<td>191</td>
</tr>
<tr>
<td>Sets</td>
<td>193</td>
</tr>
</tbody>
</table>
CONTENTS (Continued)

IX. Base Shops Maintenance Capability .......................... 202
    Initialization .................................. 206
    Output Program .................................. 206
    Permanent Variables ............................ 208
    Temporary Variables ............................ 211
    Sets ........................................... 212

X. NRTS Program ........................................ 231
    Initialization .................................. 231
    Output Program .................................. 231
    Permanent Variables ............................ 233
    Temporary Variables ............................ 233
    Sets ........................................... 233

XI. Depot Transportation Capability .............................. 239
    Cargo ........................................... 239
    Initialization .................................. 239
    Output Program .................................. 239
    Permanent Variables ............................ 242
    Temporary Variables ............................ 243
    Sets ........................................... 243
    UTIL .......................................... 248
    Initialization .................................. 248
    Output Program .................................. 248
    Permanent Variables ............................ 250
    Temporary Variables ............................ 250
    Sets ........................................... 251

XII. Depot Maintenance Capability ............................... 257
     Initialization .................................. 261
     Output Program .................................. 261
     Permanent Variables ............................ 263
     Temporary Variables ............................ 266
     Sets ........................................... 267

REFERENCES ............................................. 291
Part 1

INTRODUCTION AND INITIALIZATION INSTRUCTIONS
I. INTRODUCTION

In the research phase of weapon system development, the system manager faces the problem of designing a weapon system to meet specified operational objectives at a minimum cost. If he considers only the operational environment, the resulting hardware may be very difficult and costly to support. It often becomes apparent later, during development, that if some particular factor had been considered earlier, a more effective system could have resulted for the same cost.

Cost/effectiveness analysis, if properly used, brings into focus the parameters that affect mission capability. The object is usually to minimize the cost at which a specified level of effectiveness can be maintained; this involves a comparison of alternative ways of designing and supporting a particular system for a given mission.

The operating procedure for PLANET consists of a two-phase operation: first, the Simulation phase (S phase); second, the Report phase (R phase). The simulators can be used singly or assembled in various configurations to represent a more detailed description of the logistics system to be examined. Regardless, the output from the S phase will be a tape listing of selected variables accumulated during the simulation. This tape(s) can be retained as a permanent record of the simulation. From this tape(s), the desired reports are generated by using the following library of programs. There are twelve programs to choose from.

NOR TIME SUMMARY

The NOR (not operationally ready) time summary is a listing of the ABC Simulator NOR time distributions and a count of the unscheduled demands for the simulated fleet.

WEAPON SYSTEM AVAILABILITY

The weapon system availability program is designed for use with missile simulations. From the ABC output tape, it displays the missile off-alert time by tail number as well as a chronological, time-oriented
listing of what happened while the missile was off alert. Details include the time a team was dispatched from the support base, the arrival time at the site, when the maintenance action was completed, and when the missile was returned to alert status.

LOGISTICS RESOURCE UTILIZATION

The ABC logistics information is presented in three parts: Spares, Personnel, and Equipment. The spare part data presented by this report contain information regarding stock levels, NORS (not operationally ready--supply) time, NORS count, and demand quantities. The personnel report contains the man-hour accounting information. This report displays the utilization factors for each personnel type and the man-hours consumed by various tasks. The equipment data presented contain information regarding the utilization of maintenance equipment and facilities. NORE (not operationally ready--equipment) time, NORE count, and demand quantities are included.

The logistics resource utilization report can be used for either aircraft or missile simulations.

AIRCRAFT RECOVERY PACKAGE

The following four programs are peculiar to aircraft simulations; they cannot be used with in-place missile simulations.

Aircraft Recovery Time Distributions

The aircraft recovery histogram presents a display of the entire aircraft (as opposed to system or subsystem) recovery. The display is divided into two halves: one half shows the touchdown time by workshift, the corresponding second half shows the type of sortie. Below this display are a number of statistical computations for facilitating analysis, e.g., the average recovery time (for unscheduled maintenance) and the operationally ready (OR) time lost in recovering aircraft from the effects of the sorties, etc.
System Recovery

The aircraft System Recovery program produces a summary of the actions to clear unscheduled maintenance demands. This display serves two purposes: it enables the monitoring of break and recovery rates, and it provides a set of job standards for unscheduled maintenance. Below the display are a number of statistical computations for facilitating analysis, e.g., the total OR time lost to this subsystem, the average number of men working on system recovery, etc.

Work Center Recovery

The Work Center Recovery output is a series of frequency distributions (one for each hour of the simulated day), showing how the flight-line demands were distributed throughout the 24-hour period. The purpose of this display is to aid in determining shift assignments for personnel.

The work center data the recovery program produces are: the touchdown time, the time the work center began the first job and ended the last job, the number of people at work in each 30-minute trial period, and the conventional AFM 66-1 data.

Failure List

The Failure List provides a graphical history of break-rate information. This display presents the flight-line demand data. The break-rates are inferred from the demands, both scheduled and unscheduled, against each individual unit. The probability data are computed only for 25 or more sorties; the mean sorties-to-failure data are computed if five or more fixes have occurred.

COST/EFFECTIVENESS PROGRAM

The Cost/Effectiveness program can be set to examine two cost factors: the total system cost and the logistics support costs. Logistics costs are the summation of the various resource and facility costs specified for the simulation. Total system costs are the logistics
costs just mentioned plus the cost of the items being simulated (sites). The measure of effectiveness is operationally ready (OR) time.

**BASE SHOPS MAINTENANCE CAPABILITY**

As the title implies, this output program is used to display the outputs from the Bench Repair Simulator. The output display consists of five parts: the input to each shop and its output and repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare-part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of the simulation.

**NRTS PROGRAM**

The NRTS (not reparable this station) data display shows the reparables shipped off base for repair. It displays the pipeline time for the reparables. This program is used primarily as an input to the Depot Transportation Simulator when the simulators are operated separately.

**TRANSPORTATION CAPABILITY**

This output program is the output display for the Depot Transportation Simulator. The outputs are presented in six parts: the tonnage delivered to the various bases for each type of priority cargo; the quantity of each cargo type delivered to the various delivery points; the mileage and in-transit time for each transportation mode; the utilization rates of the vehicles assigned to the transportation system; cargo processing time and quantities; and queueing factors associated with cargo transportation at each base.

**DEPOT CAPABILITY**

The Depot Capability output program is used with the Depot Repair and Overhaul Simulator. The output displays consist of six parts:
the input to the depot, and the depot's output and in-process time distribution; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare-part type; stock levels, spare-part repair time, stockouts, and demands for each spare part; detailed information for each activity about its performance during each simulated period; and queueing factors and downtime for any "special" type of activity within the repair or overhaul process.

As previously mentioned, the manager may select from this library of programs those best suited for analysis of his particular problem. Even though the output programs cover a wide spectrum of problem areas, it can be anticipated that additional outputs will be required for some analyses. Either the output programs can be modified to incorporate any additional data required, or new programs can be written with relative ease.

This Memorandum is divided into two parts. Part I contains the introduction, which includes a brief description of each of the reports. Section II presents the SIMSCRIPT instructions required to initialize any of the report programs.

Part 2 is the library of programs, including a description of each program, the initialization requirements, a program description oriented to the skilled SIMSCRIPT programmer who may wish to make a change, and a listing of the SIMSCRIPT SOURCE program.
II. INITIALIZATION

All of the Report programs are written in SIMSCRIPT as Non-simulation programs. Non-simulation programs, as with the PLANET simulation programs (Refs. 2, 3, 4, 5), are translated by SIMSCRIPT into FORTRAN source programs, which are then compiled by the FORTRAN Monitor into a FORTRAN object program. Once the program has been compiled, the composition of the execute deck is as shown in Fig. 1.

The execution of object programs is accomplished in the usual FORTRAN manner, and whatever control cards may be required by a particular installation appear first.

![Diagram of execute deck]

The object programs may appear in any order; they are:

- A Simulation Package
- An Events Routine or a Main Routine
- All Entity, Attribute, and Set Packages
- All Exogenous Event Routines
- All Report Routines
- All Subroutines (SIMSCRIPT or FORTRAN)
- All Functions (SIMSCRIPT or FORTRAN)

The object programs are constructed from the information contained in the appropriate Report program. We shall assume at this point that the analyst has an "Object Deck" available and is now ready to assemble a "Data Deck."
DATA DECK

The composition of the Data Deck is shown in Fig. 2. The various elements and the data requirements for each are discussed below.

System Specification Card

The first card in the Data Deck is the System Specification Card. In Col. 1 must be the number 1. In Cola. 7 through 12 is punched the maximum "Array Number" as in Fig. 3. Only these two entries are required. For a complete System Specification Card format, refer to the top of Fig. 4.

Initial Conditions Deck

The Initial Conditions Deck consists of all Initialization Cards and Data Cards. Before discussing each of the cards contained in the Initial Conditions Deck, let us first discuss the SIMSCRIPT Initialization Form, because the Initial Conditions Deck is created from the information contained in the Initialization Form.

The specification of initial conditions for the Report Generators is a very simple process. Only a few arrays need be initialized in all cases. The appropriate arrays and the required values (data) are specified for each report.
### SIMSCRIPT INITIATION FORM

#### SYSTEM SPECIFICATION CARD

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use in</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### INITIALIZATION CARDS

<table>
<thead>
<tr>
<th>Array Numbers</th>
<th>Numbers/Columns</th>
<th>List Numbers</th>
<th>Table Row Numbers</th>
<th>Table Column Numbers</th>
<th>Initial Value</th>
<th>Comment</th>
<th>Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4 -- Initialization Form
The Initialization Deck consists of Data Cards plus Initialization Cards punched from the Initialization Form shown in Fig. 4. Every Array Number from "1" up to the largest must be considered in sequential order in the Initialization Cards. The complete sequence of Array Numbers must be accounted for.

Procedures for preparing the Initial Conditions Deck are discussed under the following headings:

Unsubscripted Permanent Attributes
Single-subscripted Permanent Attributes
Double-subscripted Permanent Attributes

Unsubscripted Permanent Attributes. Each unsubscripted Permanent Attribute (System Attribute) must have its initial value read in or set equal to zero.

Initial values of unsubscripted Permanent Attributes may be separately specified by means of individual Initialization Cards. They may also be handled in groups by means of a single Initialization Card followed by Data Cards. To be initialized as a group, the System Attributes in the group must have consecutive Array Numbers. Their values must also be read in by using the same FORMAT statement Field Description.

Figure 5 shows the entries required to read in the initial value of a single System Attribute. The initial value can be set to zero by

<table>
<thead>
<tr>
<th>ARRAY NUMBER</th>
<th>LIST AND TABLE DIMENSIONS</th>
<th>FORMATTED</th>
<th>INITIAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM TO</td>
<td>LIST</td>
<td>TABLE DIMENSIONS</td>
<td>FORMAT FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DESCRIPTION</td>
</tr>
</tbody>
</table>

1. Enter the Array Number inCols. 1 through 4. The unit's position of the Array Number must be in Col. 4.
2. Enter a zero in Col. 10.
3. Enter an "R" in Col. 12.
4. Enter the Initial Value as an integer or decimal number anywhere inCols. 50 through 66. Formats other than integer or decimal (e.g., hours or alphanumeric) must be read from the Data Cards.

Fig. 5 -- Initialization Card Entries for a Single Unsubscripted Permanent Attribute
inserting a zero in Cols. 50 through 66, or by leaving Col. 12 blank and inserting a "Z" in Col. 13.

**Single-subscripted Permanent Attributes.** If the initial values are to be read in, a separate Initialization Card followed by the Data Cards is required for each list of single-subscripted Permanent Attributes. If the initial values are to be set equal to zero, one or more lists of single-subscripted Permanent Attributes may be handled by a single Initialization Card, providing the lists are of the same length and have consecutive Array Numbers.

To read in the initial values of a list of single-subscripted Permanent Attributes, the Initialization Card entries shown in Fig. 6 are required.

![Initialization Card Entries](image)

1. Enter the Array Number in Cols. 1 through 4.
2. Enter a "1" in Col. 10.
3. Enter an "R" in Col. 12.
4. In Cols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 5 below.
5. In Cols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is preset for each table. It is preprinted in Table 7.
6. In Cols. 50 through 66, enter a single FORMAT statement Field Description enclosed in parentheses and preceded by an optional constant, if desired. This Field Description tells how the initial values of the list are to appear in the subsequent Data Cards. Each Data Card will be read starting in Col. 1. If desired, successive values may appear across the Data Card.

**Fig. 6 -- Initialization Card Entries for Reading In a Single-subscripted Permanent Attribute List**
One or more lists of single-subscripted Permanent Attributes describing the same Entity and having consecutive Array Numbers can be initially set equal to zero by the Initialization Card entries shown in Fig. 7. Inserting the letter "Z" in Col. 13 causes zeros to be stored in the entire word.

1. Enter the lowest Array Number inCols. 1 through 4.
2. Enter the highest Array Number inCols. 5 through 8.
3. Enter a "1" in Col. 10.
4. Enter a "2" in Col. 13.
5. InCols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 6 below.
6. InCols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is present for each table. It is preprinted in Table 7.

Fig. 7 -- Initialization Card Entries for Setting Single-subscripted Permanent Attribute Lists to Zero

Double-subscripted Permanent Attributes. If non-zero initial values are to be read in for a table of double-subscripted Permanent Attributes, each table requires a separate Initialization Card followed by Data Cards containing the values. However, a single Initialization Card may serve to zero out one or more Attribute tables, providing they all describe the same pair of Permanent Entities and have consecutive Array Numbers. The procedure for setting Ragged Tables equal to zero is described below.

Figure 8 shows the Initialization Card entries required for reading in the initial values of a table of double-subscripted Permanent Attributes.
1. Enter the Array Number inCols. 1 through 4.
2. Enter a "2" inCol. 10.
3. Enter an "R" inCol. 12.
4. InCols. 15 through 18, enter the largest value that the (row) subscript may take on. This will be the same value to which the Entity that describes the (row) units/coordinates of the table has been initialized. Refer to Item 5 below.
5. InCols. 19 through 22, enter the Array Number of the Entity that describes the (row) units/coordinates of the table. This Array Number is a function of the program and is present for each table. It is preprinted in Table 7.
6. InCols. 23 through 26, indicate the largest column subscript.
7. InCols. 27 through 30, enter the Array Number of the System Variable, the value of which is equal to the value of the largest column subscript.
8. Indicate the order in which the Attribute values are to be read from the Data Cards by entering an "B" inCol. 36 if the values are to be read across rows, or entering a "C" inCol. 37 if they are to be read down columns.
9. If the beginning of each new row or column is to start on a new Data Card, enter an "N" inCol. 38. If, instead of starting on a new card, the first entry in a new row or column immediately follows the last entry in the preceding row or column, put an "F" in Col. 39.
10. InCols. 50 through 66, enter a FORMAT statement Field Description enclosed in parentheses indicating how the table entries are to appear in subsequent Data Cards.

Fig. 8 -- Initialization Card Entries for Reading in a Double-subscripted Permanent Attribute Table

Initial Conditions. SIMSCRIPT requires that all permanent system variables be given initial values in ascending order of their Numbers (1-N). Data Deck Card 2 to the end (the last array number "N") is the initial conditions deck.

With each output program is a Variable Description and Initialization Table to specify the initial value(s) assigned to each permanent
system variable. The Formats used to initialize the different types of variables (e.g., unsubscripted, single-subscripted, double-subscripted) have been previously described. There are no exogenous events used in any of the report generators.
Part 2

PROGRAM LIBRARY

Program 1

NOR TIME SUMMARY
I. NOR TIME SUMMARY

The NOR (Not Operationally Ready) time summary (Fig. 9) is a listing of the downtime distributions for any set of fail levels and a count of the demands for the simulated fleet. The program may be initialized to specify any time period desired as well as any portion of the simulated fleet, i.e., a single base or all bases. This output is shown in the following figure along with an explanation of each of the columns of the output listing.

INITIALIZATION

Table 1 lists the Initialization requirements. Only six arrays require inputs. Array 23 is the number of bases to be analyzed. Array 24 lists the base (quantity specified in Array 23) numbers. Array 26 is the quantity of different failure levels to be counted in the NOR time. Array 27 lists the failure level numbers. Array 33 is the time that the reports are to end. Array 53 is the Report Interval, which specifies the time period at which the data are to be accumulated and printed (the example data are initialized for 1-day reports). All of the other arrays are set to zero.

For the initialization formats, the user may use the example data contained with the program listing or refer to Section 2, Initialization instructions for unsubscripted and single-subscripted system variables.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-FR.)

When a label record is READ from the input tape, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may run at one time.
### Hourly Repair Data Summary

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Start Time</th>
<th>End Time</th>
<th>Work Days</th>
<th>Time Spent</th>
<th>Maintenance Requests</th>
<th>Repair Time</th>
<th>AVG Repair Time</th>
<th>STD Dev</th>
<th>MIN Repair Time</th>
<th>MAX Repair Time</th>
<th>AVG STD Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:00</td>
<td>01:00</td>
<td></td>
<td>0</td>
<td>0.26%</td>
<td>0.66%</td>
<td>0.74%</td>
<td>0.49%</td>
<td>0.67%</td>
<td>0.00%</td>
<td>0.964%</td>
<td>0.749%</td>
</tr>
<tr>
<td>01:00</td>
<td>02:00</td>
<td></td>
<td>0</td>
<td>0.26%</td>
<td>0.66%</td>
<td>0.74%</td>
<td>0.49%</td>
<td>0.67%</td>
<td>0.00%</td>
<td>0.964%</td>
<td>0.749%</td>
</tr>
<tr>
<td>02:00</td>
<td>03:00</td>
<td></td>
<td>0</td>
<td>0.26%</td>
<td>0.66%</td>
<td>0.74%</td>
<td>0.49%</td>
<td>0.67%</td>
<td>0.00%</td>
<td>0.964%</td>
<td>0.749%</td>
</tr>
<tr>
<td>03:00</td>
<td>04:00</td>
<td></td>
<td>0</td>
<td>0.26%</td>
<td>0.66%</td>
<td>0.74%</td>
<td>0.49%</td>
<td>0.67%</td>
<td>0.00%</td>
<td>0.964%</td>
<td>0.749%</td>
</tr>
</tbody>
</table>

---

**Fig. 9 -- HOR Time Summary**
<table>
<thead>
<tr>
<th>Array Number</th>
<th>Number of Elements</th>
<th>Mode</th>
<th>Initialization to Initialize Variable in Table Col.</th>
<th>Array Number of Elements to Be Passed to Table 5 Col.</th>
<th>List Packing</th>
<th>Description of Variable to Be Initialized</th>
<th>Commencement Syntax</th>
<th>Variable</th>
<th>Entity</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Number of zones to be analyzed.</td>
<td>ENDR</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Specify each zone code to be analyzed.</td>
<td>Bilk</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Number of failure levels to be analyzed.</td>
<td>PL2F</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Specify each failure level code to be analyzed.</td>
<td>Bilk</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Then to end report.</td>
<td>FEND</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td>Reporting interval.</td>
<td>FEND</td>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30-12</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36-12</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-15</td>
<td>0</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When a label record is read, the value of the failure level is compared with the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

If ETIME is greater than ETIME, REPORT is called and the output is displayed. If ETIME is greater than TMEND, REPORT is called and the output is displayed. The run is then terminated.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-FR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/Wk
SXDW = S/Wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
THSM = ID Addresses
MORE = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-FR).

DTLV1 = Integer variable 1
DTLV2 = Integer variable 2
DTLV3 = Integer variable 3
DTLV4 = Integer variable 4
DTLV5 = Integer variable 5
DTLV6 = Integer variable 6
DTLV7 = Integer variable 7
DTLV8 = Integer variable 8
DTLV9 = Float variable 1
DTLV0 = Float variable 2
Base table.

BASES = Number of base codes to be processed.
BASE = Base codes to be processed.
BFLAG = Controls flow of events as a result of EBAS vs. BASES.

Failure level table.

FLVLS = Number of failure codes to be processed.
FLVL = Failure level codes to be processed.
TEND = Time initialized to end this run prematurely.

Variables used to output display. (Time in decimal days).

D1 = Reporting time
D2 = NOR time this period
D3 = NOR time to date
D4 = Number of NOR IN this period
D5 = Number of NOR OUT this period
D6 = Number of NOR IN to date
D7 = Number of NOR OUT to date
D8 = Min. NOR time this period
D9 = Max. NOR time this period
D10 = Avg. NOR time this period
D11 = Std. dev. for this period
D12 = Min. NOR time to date
D13 = Max. NOR time to date
D14 = Avg. NOR time to date
D15 = Std. dev. to date

Calculation variables.

TIMEN = NOR time for each period.
SUMP2 = Std. dev. this period.
SUMD2 = Std. dev. to date.
RTIME = Reporting time.
VTIME = Reporting time interval.
LINE = Counter used in report phase.
SETS

Name = NQUE used for NOR events. No subscripts. Ranked N1.
Owner = SIMSCRIPT system.
Member = NOR
    N1 = Event time - start of this period.
    N3 = System code.
    N4 = Failure level code.
    N8 = Team ID.
    N9 = Request ID.
    N10 = Site ID.
    N11 = IDSUB.
    N12 = Event time - start to date.

Name = PM used for preventive maintenance events. No subscripts. Ranked on P1.
Owner = SIMSCRIPT system.
Member = PM
    P1 = Event time - start PM.
    P2 = System code.
    P4 = Request ID.
    P5 = Site ID.

Name = NQUE used for overhaul events. No subscripts. Ranked on H1.
Owner = SIMSCRIPT system.
Member = OH
    H1 = Event time - start of overhaul.
    H2 = System code.
    H3 = Team ID.
    H4 = Request ID.
    H5 = Site ID.
110SB4 0 1
210SB4 0 1
3SHFT 0 1
4DAYW 0 1
55XDW 0 1
6EEAS 0 1
7VA 0 1
8VR 0 1
9VC 0 1
1CTRSP 0 1
11MORL 0 1
12ETIME 0 F
130TLV1 0 1
140TLV2 0 1
150TLV3 0 1
160TLV4 0 1
170TLV5 0 1
180TLV6 0 1
190TLV7 0 1
200TLV8 0 1
210TLV9 0 F
220TLV0 0 F
23BASIS 0 E 1
24BASE 0 1
25FLAG 0 1
26FLVLS 0 1
27FLVL 0 1
28FLQDF 0 1
29FLQDF 0 1
30STOPES 0 1
31LQDF 0 1
32LQDF 0 1
33LQDF 0 1
34LQDF 0 1
35LQDF 0 1
36LQDF 0 1
37LQDF 0 1
38LQDF 0 1
39LQDF 0 1
40LQDF 0 1
41LQDF 0 F
42LQDF 0 F
43LQDF 0 F
44LQDF 0 F
45LQDF 0 F
46LQDF 0 F
47LQDF 0 F
48LQDF 0 F
49LQDF 0 F
50LQDF 0 F
51LQDF 0 F
52LQDF 0 F
53LQDF 0 F
*IBF0C MAIN
   MAIN ROUTINE
C
C       PLANET - NOR TIME SUMMARY.....
C
C       PURPOSE - TO REPORT NOR TIME.
C
C       INPUT - TAPE FROM ABC MODEL.
C
C       OUTPUT - PRINTER (SIMSCRIPT RPG).
C
REWIND 9
C
LET RTIME = RTIME + VTIME
LET OR  = 99999.99999
LET DI7 = 99999.99999
C
1 CALL RLAL
   CALL SELECT
C
   IF (IDSUB) EQ ( 3), GO TO 3
   IF (BFLAG) NE ( 0), GO TO 9999
   IF (IDSUB) EQ (110), GO TO 110
   IF (IDSUB) EQ (200), GO TO 200
   IF (IDSUB) EQ (500), GO TO 500
   IF (IDSUB) EQ (600), GO TO 600
   IF (IDSUB) EQ (1900), GO TO 1900
   IF (IDSUB) EQ (2000), GO TO 2000
   IF (IDSUB) EQ (2100), GO TO 2100
   IF (IDSUB) EQ (2150), GO TO 2150
   IF (IDSUB) EQ (2300), GO TO 2300
   IF (IDSUB) EQ (2400), GO TO 2400
   IF (IDSUB) EQ (2900), GO TO 2500
C
   GO TO 9999
C
3 CALL R3
   CALL EXIT
C
110 CALL R110
   GO TO 9999
C
200 CALL R200
   GO TO 9999
C
500 CALL R500
   GO TO 9999
C 600 CALL R600
   GO TO 9999
C 1900 CALL R1900
   GO TO 9999
C 2000 CALL R2000
   GO TO 9999
C 2100 CALL R2100
   GO TO 9999
C 2150 CALL R2150
   GO TO 9999
C 2300 CALL R2300
   GO TO 9999
C 2400 CALL R2400
   GO TO 9999
C 2500 CALL R2500
   GO TO 9999
C 9999 IF (MORF) EQ (0). GO TO 1
   CALL RDTL
   GO TO 9999
C FND
*IRFTC LABEL

SUBROUTINE IRFTC

C

$KADS S-PHASE TAPE(9) (BIN MODE)....LABEL RECORDS.

C

LET AFLAG = 0

C

READ (9) 11,12,13,14,15,16,17,18,19,110,111,112

C

STORE 11 IN IDSOR
STORE 12 IN IDSUOR
STORE 13 IN SHFT
STORE 14 IN DAYW
STORE 15 IN SXOW
STORE 16 IN ERAS
STORE 17 IN VA
STORE 18 IN VB
STORE 19 IN VC
STORE 110 IN TRSM
STORE 111 IN MORE
STORE 112 IN FTIME

C

IF (TIME) GR (TMDN), GO TO 3

C

1 IF (TIME) GR (HTIME), GO TO 2
RETURN

C

2 CALL REPORT
GO TO 1

C

3 LET FTIME = TMEND
CALL R3
CALL EXIT

C

END
*IMHTC SELECT
  SUBROUTINE SELECT
  C
  C.....PURPOSE - TO SELECT EVENTS BY CASE.
  C
  DO 11 I = 1, FOR EACH CASES 1
     IF (ERAS) EQ (BASE(I)), GO TO 2
  1 LOOP
     LET HFLAG = 1
     GO TO 9999
  C
  2 LET HFLAG = 0
     GO TO 9999
  C
  9999 RETURN
  END

*IMHTC RUTL
  SUBROUTINE RUTL
  C
  C.........READS S-PHAS SF TAPE(9) (BIN MODE).....DETAIL RECORDS.
  C
  X     READ (9) 11,12,13,14,15,16,17,18,19,110
  C
  STORE 11 IN DTLV1
  STORE 12 IN DTLV2
  STORE 13 IN DTLV3
  STORE 14 IN DTLV4
  STORE 15 IN DTLV5
  STORE 16 IN DTLV6
  STORE 17 IN DTLV7
  STORE 18 IN DTLV8
  STORE 19 IN DTLV9
  STORE 110 IN DTLVO
  C
  LET MORE = MORE - 1
  C
  RETURN
  END
*IPFCTC R3

SUBROUTINE R3

C
C......PURPOSE - TO CLOSE-OUT AND END R-PHASE.
C
C......IOUSRK = 3.
C
C      LET LINF = 0
      LET RTIMF = CTIME
C
10 DO TO 3, FOR EACH M IN NQUE
C      REMOVE M FROM NQUE
C      LET TIMFN = ETIME - N1(M)
      LET TIMEM = ETIME - N2(M)
C      LET D5 = D5 + 1
      LET D7 = D7 + 1
      LET D2 = D2 + TIMEN
      LET D3 = D3 + TIMEN
C      LET SUMP2 = SUMP2 + TIMEN**2
      LET SUMD2 = SUMD2 + TIMFN**2
C      IF (TIMFN) GE (DA1), GO TO 1
      LET DA = TIMEN
C      1 IF (TIMEN) LE (D9), GO TO 2
      LET D9 = TIMEN
C      2 IF (TIMEN) LE (D13), GO TO 3
      LET D13 = TIMFN
C      3 REPEAT 10
C      CALL REPORT
C      REWIND)
C      RETURN
END
*IPFLC N110

ROUTINE R110

C
C
C*****PURPOSE - START WR FOR EXIG. FAILURE, EXIG. PM, EXIG. OVERHAUL.
C
C
C*****INSUM = 110.
C
C
IF (VC) EQ (2), GO TO 2
IF (VC) EQ (4), GO TO 4
IF (VC) EQ (6), GO TO 6
GO TO 9999
C
C**********EXIG. FAILURE.
C
2  DO TO 20, FOR EACH FLVLS 1
   IF (VR) EQ (FLVLS(1)), GO TO 21
20  LOOP
   GO TO 9999
C
21 CREATE WR CALLED N
   STORE FTIME IN N1(N)
   STORE FTIME IN N12(N)
   STORE VA IN N3(N)
   STORE VP IN N4(N)
   STORE TRSM IN N10(N)
   STORE LUSUR IN N11(N)
   FILE N IN 'QURF
   LET N4 = N4 + 1
   LET N6 = N6 + 1
   GO TO 9999
C
C**********EXIG. PM.
C
4 CREATE PM CALLED P
   STORE FTIME IN P1(P)
   STORE VA IN P2(P)
   STORE TRSM IN P5(P)
   FILE P IN PQUE
   GO TO 9999
C
C**********EXIG. OVERHAUL.
C
6 CREATE UN CALLED H
   STORE FTIME IN H1(H)
   STORE VA IN H2(H)
   STORE TRSM IN H5(H)
   FILE H IN HQUE
   GO TO 9999
C
9999 RETURN
   END
*IPFTC R200
SUMROUTINE R200
C
C
C......PURPOSE - TO END FORK OR MAINTENANCE COMPLETED.
C
C
C......IDSUM = 200.
C
C
IF (MORE) EQ (0), GO TO 9999
CALL KD1L
C
FINF FIRST, FOR EACH M IN NOUF, WITH (N8(M)) EQ (FRLS),
A WHERE IN, IF NONE, (0) TO 9999
C
LET TIMEN = ETIMEN = T4L(I4)
C
LET D7 = D7 + 1
LET D2 = D2 + TIMEN
LET D3 = D3 + TIMEN
C
LET SUMP2 = SUMP2 + TIMEN**2
LET SUMO2 = SUMO2 + TIMEN**2
C
IF (TIMEN) GT (D9), GO TO 1
LET D8 = TIMEN
C
1 IF (TIMEN) LT (D9), GO TO 2
LET D9 = TIMEN
C
2 IF (TIMEN) GE (D12), GO TO 3
LET D12 = TIMEN
C
3 IF (TIMEN) LT (D13), GO TO 4
LET D13 = TIMEN
C
4 REMOVE IN FROM NOUE
DESTROY FOR CALLED IN
C
4999 RETURN
END
*IBFTC RS00
SUBROUTINE RS00

C
C PURPOSE - TEAR DISPATCH BY BASE CONTROL.
C
C
C TSLUP = 500.
C
C
IF (MORF) EQ 10, GO TO 9999
CALL Rcil
C
IF (DLTV4) EQ (MSIL), GO TO 9999
C
FIND FIRST, FOR EACH M IN NNP, WITH (M(N)) EQ (VC)
X WHERE IN, IF NOUN, GO TO 1
C
STORE TASK IN NNP(N)
GO TO 9999
C
1 FIND FIRST, FOR EACH M IN NNP, WITH (M(N)) EQ (VC)
X WHERE IN, IF NOUN, GO TO 9999
C
STORE TASK IN NNP(N)
GO TO 9999
C
9999 RETURN
END
*IMTE K800

    SUBROUTINE K800

    C
    C
    C......PURPOSE - TELE ARRIVAL AT SITE.
    C
    C
    C......IDSUM = 600.
    C
    C
    IF (VB) IF (0), GO TO 9999
    C
    FIND FIRST, FOR EACH M IN HQUE, WITH (H3(M)) = (TSM),
    C
    WHERE IM, IF NONE, GO TO 9999
    C
    CREATE YOUR CALLED IN
    STORE ETIME IN 41(N)
    STORE ETIME IN 412(N)
    STORE F2(IN) IN N3P(N)
    STORE H3(IN) IN 48(N)
    STORE H4(IN) IN 47(N)
    STORE H5(IN) IN 410(N)
    STORE IDSUM IN 411(N)
    C
    FILL N IN HQUE
    C
    LET 04 = 04 + 1
    LET 06 = 06 + 1
    C
    REMOVE IM FROM HQUE
    PREVIOUS IN CALLED IN
    C
    9999 RETURN
    C

*IMTE R1900

    SUBROUTINE R1900

    C
    C
    C......PURPOSE - GENERATE PROP IN WSITE FOR RETURNING TRANS. (R190)
    C
    C
    C......IDSUM = 1900.
    C
    C
    IF (M(N)) < (0), GO TO 9999
    CALL RDTL
    C
    IF (VC) GO (MSITE), LET WSITE = VC + 1
    C
    9999 RETURN
    C

C
*IBFTC R2000
SUBROUTINE R2000
C
C......PURPOSE - REQUEST FOR PP.
C
C......IDSUP = 2000.
C
C CREATE PP CALLED P
C
STORE ETIME IN P1(P)
STORE VAR IN P2(P)
STORE TASM IN P5(P)
C
FILE P IN POUT
C
9999 RETURN
END

*IBFTC R2100
SUBROUTINE R2100
C
C......PURPOSE - TO START NUR FOR PP.
C
C......IDSUP = 2100.
C
C FIND FIRST, FOR EACH M IN POUT, WITH (P1(M)) EQ (VAR),
AND (P2(M)) EQ (V8), WHERE IP, IF NURF, GO TO 9999
C
C CREATE NLR CALLED N
C
STORE ETIME IN N1(N)
STORE ETIME IN N2(N)
STORE P2(IP) IN N3(N)
STORE VC IN N4(N)
STORE P4(IP) IN N5(N)
STORE P5(IP) IN N9(N)
STORE IDSUP IN N11(N)
C
FILE N IN NOUT
C
LET OA = OA + 1
LET OB = OB + 1
C
REMOVE IP FROM POUT
DESTROY PN CALLED IP
C
9999 RETURN
END
*IRFTC R7150

SUBROUTINE R7150

C

C....PURPOSE - TO START NOR FOR FAILURE CAUSED BY PM.

C

C....INSUM = 2150.

C

DO TO 1, FOR EACH FLVL, I
IF (VA) EQ (FLVL(I)), GO TO 2
1 LOOP
   GO TO 9999

C

2 CREATE NOR CALLED N

C
STORE ETIME IN N1(N)
STORE ETIME IN N12(N)
STORE VG IN N3(I)
STORE VA IN N4(N)
STORE TRSM IN N10(I)
STORE TDUM IN N11(N)

C
FILE N IN NQUE

C
LET D4 = D4 + 1
LET D6 = D6 + 1

C
9999 RETURN
END
*IRFTC R2300

SUPROUTINE R2300

C
C

*****PURPOSE - TO START NOR FOR ALERT-CONTINUOUS MONITOR.
C
C

*****IDSUB = 2300.
C
C

(1) TO L FOR EACH FLVL_S I
IF (VA) EQ (FLVL_I), GO TO 2
1 LOOP
GO TO 9999
C

2 CREATE NOR CALLED A
C

STORE FTIME IN N1(N)
STORE FTIME IN N12(N)
STORE VH IN N3(N)
STORE VA IN N4(N)
STORE TRSM IN N10(N)
STORE IDSUB IN N11(N)
C

FILE N IN NOQUE
C

LET D4 = D4 + 1
LET D6 = D6 + 1
C

9999 RETURN
FIND
*IBFTC R2400

SUBROUTINE R2400

C

C.......PURPOSE - RESOURCE REQUEST.

C

C.......IDSRH = 2400.

C

IF (MOCR) EQ (0), GO TO 9999
CALL ROTL

IF (DLV1) EQ (1), GO TO 1
IF (DLV1) EQ (2), GO TO 2
IF (DLV1) EQ (3), GO TO 3
IF (DLV1) EQ (4), GO TO 4
IF (DLV1) EQ (5), GO TO 5
IF (DLV1) EQ (6), GO TO 6
GO TO 9999

1 FIND FIRST, FOR EACH M IN NCUF, WITH (N1(M)) EQ (2300),
XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
GO TO 9999

STORE VC IN 4014
GO TO 9999

2 FIND FIRST, FOR EACH M IN NCUF, WITH (N1(M)) EQ (110),
XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
GO TO 9999

STORE VC IN 4014
GO TO 9999

3 FIND FIRST, FOR EACH M IN NCUF, WITH (N1(M)) EQ (2500),
XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
GO TO 9999

STORE VC IN 4911
GO TO 9999

4 FIND FIRST, FOR EACH M IN NCUF, WITH (P5(M)) EQ (TRSM),
XAND (P2(M)) EQ (VB), WHERE IP, IF NONE, GO TO 9999

STORE VC IN 4416
GO TO 9999

5 FIND FIRST, FOR EACH M IN NCUF, WITH (N1(M)) EQ (2150),
XAND (N10(M)) EQ (TRSM), AND (N3(M)) EQ (VB), WHERE IN, IF NONE,
GO TO 9999

STORE VC IN 4911
GO TO 9999
C
6 FIND FIRST, FOR EACH M IN HQUI, IF (HS(M)) EQ (THSM),
   XAND (H2(M)) EQ (VH), WHERE IN, IT NONE, GO TO 9999
C
STORE VC IN H411H
GO TO 9999
C
9999 RETURN
END

*HEPC R2500
SUBROUTINE R250C
C
C.....PURPOSE - TO START NOR FOR UNDETERMINED FAILURE.
C
C.....IDSUR: = 2500.
C
C   LO TO 1, FOR EACH FLVLS 1
C
1 IF (VA) EQ (FLVL(1)), GO TO 2
C   1 LOOP
C   GO TO 9999
C
C   2 CREATE NOR CALLED N
C
   STORE ETFM IN N11(N)
   STORE ETFM IN N12(N)
   STORE VH IN N31(N)
   STORE YA IN N41(N)
   STORE THSM IN N410(N)
   STORE IDSUR IN N11(N)
C
   FILE N IN NQUI
C
   LET D4 = D4 + 1
   LET D6 = D6 + 1
C
   9999 RETURN
   END
*INITL REPORT

SUBROUTINE REPORT

C

C PURPOSE - TO REPORT WAR TIME AS OF A GIVEN TIME.

C CALLED BY RLIT OR R3.

C

LET IP = 05
LET ID = 07

200 DO D1 TO 299, FOR EACH M IN QUEUE

C

LET TIMEN = RTIME - RI(M)
LET TIMEM = RTIME - N12(M)
LET N1(M) = RTIME

C

LET D2 = D2 + TIMEN
LET D3 = D3 + TIMEN

C

LET SUMP2 = SUMP2 + TIMEN**2
LET SUMD2 = SUMD2 + TIMEN**2

C

LET IP = IP + 1
LET ID = ID + 1

C

IF (TIMEN) GT (DB), GO TO 201
LET DB = TIMEN

201 IF (TIMEN) LE (D9), GO TO 202
LET D9 = TIMEN

202 IF (TIMEN) LE (D13), GO TO 299
LET D13 = TIMEM

299 REPEAT 200

C

IF (LINE) EQ (01), GO TO 2
IF (LINE) EQ (50), GO TO 2

C

1 LET D1 = RTIME

IF ((SUMP2 - (FLOAT(IP) * (D2 / FLOAT(IP)**2))) / (FLOAT(IP)))

X10,10,11

10 LET D11 = 0.0

GO TO 12

11 LET D11 = SQR((SUMP2 - (FLOAT(IP) * (D2 / FLOAT(IP)**2)))

X / (FLOAT(IP)))

12 IF ((SUMD2 - (FLOAT(ID) * (D3 / FLOAT(ID)**2))) / (FLOAT(ID)))

X13,13,14

13 LET D15 = 0.0

GO TO 15
14 LET D15 = SQRT((SUM+ - 'FLOAT(1D) * (D3 / FLOAT(1D)**2)))
   X / (FLOAT(1D))
15 LET D10 = D2 / FLOAT(1F)
   LET D14 = D3 / FLOAT(1D)
C
 IF (D8) GR (99999.0), LET U9 = 0.0
 LET S12 = D12
 IF (D12) GR (99999.0), LET S12 = 0.0
C
 CALL DETAIL
C
 LET LINE = LINE + 1
C
 LET HTIME = HTIME + VTIME + .00001
C
 LET h7 = 0.0
 LET C4 = 0
 LET D5 = 0
 LET NA = 99999, 9999
 LET OS = 0.0
 LET D17 = S12
 LET SUMP: = 0.0
C
 GO TO 9999
C
 2 CALL HDING
 LET LINE = 0
 GO TO 1
C
 9999 RETURN
**INITIALIZE DECK**

<table>
<thead>
<tr>
<th>ENTRY</th>
<th>MAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>1 K</td>
</tr>
<tr>
<td>00</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
</tr>
</tbody>
</table>
Program 2

WEAPON SYSTEM AVAILABILITY
II. WEAPON SYSTEM AVAILABILITY

The Weapon System Availability program can be used with either aircraft or missiles. It displays the off-alert time by tail number as well as a chronological, time-oriented listing of what happened while the site was off alert. This output is shown in Fig. 10. An explanation of each of the column headings follows.

**Event Time:** The simulated time at which the various events take place.

**Tail No.:** The tail number of the site.

**Site Type:** The site type number specified in the ABC simulator Exogenous Event Genera data card, Col. 13.

**SQUAD:** Squadron or Base number that was specified in the ABC simulator Exogenous Event Genera data card, Cols. 19-20.

**Init State:** Indicates the initial status that removed the site from the available status. Codes are as follows:

- AA = Detected failure
- AB = Latent failure
- ES = Exogenous failure
- OH = Unit removed for overhaul or time replacement
- 01-12 = Periodic Maintenance number

**Total Off-Alert:** Indicates the actual amount of time (days, hours, minutes) that the site is (not) available.

**Event Stat:** Indicates the events that occurred while the site was not available. Codes are as follows:

- AA = Detected failure
- AB = Latent failure
- ES = Exogenous failure
- OH = Unit removed for overhaul or time replacement
- 01-12 = Periodic Maintenance number
- AD = Demand
- AT = Team Dispatched by Base Control
- AS = Team Arrives at Site
- XA = Maintenance Successfully completed.
FD = Personnel failure during maintenance
FE = Equipment failure during maintenance
FS = Spare part failure during maintenance

**SYST FAIL:** Indicates the Unit responsible for the demand.

**FAIL LEVEL:** Failure level.

**LAC TIME:** The time the maintenance team arrives at the site minus the time the team is dispatched (AS - AT) in days, hours, minutes.

**DUR TIME:** The time that the site is not available. The time that the maintenance is successfully completed minus the time that the site went off alert. (XA-AA) or (XA-AB) or (XA-ZZ) or (XA-OH) or (XA-01 to 12).

**ALERT Dex. Time:** The time that maintenance was successfully completed minus the time of the failure. In the event of multiple failures, the subsequent degradation time will be from fix time to fix time.

**SEQ No:** Self-explanatory.

**TID:** Team Identification number used only for program checkout.

**RID:** Request Identification number used for program checkout.

**SID:** Site Identification number used for program checkout.

**EVNT:** Label Record number printed by the simulation phase.

The last page of the report will display the status of the sites that remain not available at the end of simulation.
-46-

~~~~"sl e220:0
o000

00000

0000

40

00000

000Man

00

00000

000000

00

W%00000

cc

c0

0to0Oc

00000

00000

000

00000

00000

00000

000000

60@0*000

mooos

£0

-0000

Ncocoa

-0

000

voo

Kb.

000

00000

0000

40.

000

00000

-0000

0000000

000000

cc

000000

0000o0

c0

00co00

0-00000

000000

00

000000C

COOoCO

000000

00

C-00000

000000

000000

~o

00000
P.00000
0c0000

00000

00000

00.0000

00

000000

00008

00000

000000

00

000000

NCC

o4~'C

4000oa

00

000000

00

000000

000000

c0

000000

000000

LU

000000

.0
000NOOO
000
W&

000
000
00

#00000000

c000
00000
0

P0000 -0000
000000000

.a.444NNNNNt

NNP.N

#6104~

000000o

20000

Wa%$eet

---

n"a

00

000000o
.00000

00

000000

0o

-----

00444 --

N

ftNNNNV

.NNN

NNN"N

P.....

.....

000000

5
000000

001%
00

NNNNNP
4440

000000
0%000
00c000

NNNNN

-

A."

44

4044O

A"

00044W

Oot

*f*000NOA.fs20-

0%0
000

00000t

00000t

00000

222

000000

ev009000

222
2::: 2222 :t
20422

000000

!112

:2!.Oe"

0

000
%

9922

-


<table>
<thead>
<tr>
<th>EVENT</th>
<th>TIME</th>
<th>TAIL NO</th>
<th>SITE TYPE</th>
<th>SQUAD</th>
<th>INIT STAT</th>
<th>TOTAL OFF-ALERT</th>
<th>EVRT STAT</th>
<th>SYST FAIL</th>
<th>FAIL LEVEL</th>
<th>LAG TIME</th>
<th>DUR TIME</th>
<th>ALERT DEG TIME</th>
<th>SEQ NO</th>
<th>TID</th>
<th>RID</th>
<th>SID EVNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>12</td>
<td>19</td>
<td>34</td>
<td>24</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>02</td>
<td>51</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>07</td>
<td>61</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>03</td>
<td>31</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>06</td>
<td>41</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 10 -- Continued
<table>
<thead>
<tr>
<th>EVENT TIME</th>
<th>TAIL NO.</th>
<th>SITE TYPE</th>
<th>SQUAD</th>
<th>INIT STAT</th>
<th>TOTAL EVNT</th>
<th>SYST FAIL</th>
<th>FAIL LEVEL</th>
<th>LAG TIME</th>
<th>DUR TIME</th>
<th>ALERT TIME</th>
<th>SEQ. TIME</th>
<th>NO.</th>
<th>TID</th>
<th>RID</th>
<th>SID</th>
<th>EVNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 7 47</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>*********</td>
<td>0 0 0</td>
<td>3 4</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>1 0</td>
<td>0</td>
<td>29445</td>
<td>2500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 1 12</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>*********</td>
<td>0 0 0</td>
<td>4 4</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>1 0</td>
<td>0</td>
<td>29447</td>
<td>2500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 13 26</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0 2 2 16</td>
<td>0 0 0</td>
<td>4 4</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>1 0</td>
<td>0</td>
<td>29295</td>
<td>2500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 11 42</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>0 0 0 0</td>
<td>0 0 0</td>
<td>1 6</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>2 0</td>
<td>0</td>
<td>29295</td>
<td>2500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 4 3</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>*********</td>
<td>0 0 0</td>
<td>5 1</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>0 0 0</td>
<td>1 0</td>
<td>0</td>
<td>29271</td>
<td>2500</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 10 -- Continued
INITIALIZATION

Table 2 lists the Initialization requirements. Only seven Arrays require initialization. Array 23 is the total number of bases to be examined. Array 24 lists the base numbers (quantity specified in Array 25). Array 26 is the quantity of different failure levels to be counted in the report. Array 27 lists the failure level numbers. Array 28 is the quantity of status codes to be included in the report. Array 29 lists the status codes. Array 33 specifies the time for the end of the report. All of the other arrays are set to zero.

The initialization instructions for unsubscripted and subscripted system variables are contained in Part 1, Sec. II of this Memorandum.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may be run at one time.

When a label record is read, the value of the failure level is compared to the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

The output display is given whenever a SITE is returned to on-alert status.

If ETIME is greater than TMEND, REPORT is called and the output is displayed. The run is then terminated.

A list of status codes for the STATS table is given on page 44.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.
### Table 2

**VARIABLE DESCRIPTION AND INITIALIZATION:**

**WEAPON SYSTEM AVAILABILITY**

<table>
<thead>
<tr>
<th>Array Number</th>
<th>Number of Subscripts</th>
<th>Mode</th>
<th>Floating Point</th>
<th>Zone</th>
<th>Value</th>
<th>Table Column 19-22</th>
<th>Table Column 27-30</th>
<th>List Ranking</th>
<th>Description of Variable to Be Initialized</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-32</td>
<td>0</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of times to be analyzed.</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of failures to be analyzed.</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number of states to be analyzed.</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time to end report.</td>
</tr>
</tbody>
</table>

Permanent System Variable Name | Entity | Attribute
---|---|---
BASE | E | A
FIVL | E | A
STAT | E | A
THIRED | E | A
Label records (see page 108 of RM-4659-PR).

IDSOR = Idr
IDSUB = Idd
SHFT = Shift
DAYW = Dy/wk
SXDW = S/wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRSM = ID Addresses
MORE = Dri
ETIME = Event time

Detail records (see page 109 of RM-4659-PR).

DTLV1 = Integer variable 1
DTLV2 = Integer variable 2
DTLV3 = Integer variable 3
DTLV4 = Integer variable 4
DTLV5 = Integer variable 5
DTLV6 = Integer variable 6
DTLV7 = Integer variable 7
DTLV8 = Integer variable 8
DTLV9 = Float variable 1
DTLV0 = Float variable 2

Base table.

BASES = Number of base codes to be processed.
BASE = Base codes to be processed.
BFLAG = Controls flow of events as a result of EBAS vs BASES.

Failure level table.

FLVLS = Number of failure codes to be processed.
FLVL = Failure level codes to be processed.

Status code table (See Fig. 10).
STATS = Number of status codes in table.
STAT  = Status codes (Alpha).
TMEND = Time initialized to end this run prematurely.

SETS

Name  = SQUE used for site events. No subscripts. Ranked on SERNO.
Owner = SIMSCRIPT system.
Member = SITES
    SID  = Site ID number.
    SMODE = Mode of site.
    SERNO = Tail number of site.
    EBASE = Base number of site.
    SINO = Number (1-n) assigned to site by the program.
    NFAIL = Number of failures occurring at the site.

Name  = EQUE used for site events. One subscript. Ranked on E1.
Owner = SITES.
Member = EVENT (Used for reporting display). (Time in hours, days, minutes).
    E1  = Event time.
    E2  = Event status.
    E3  = System failed.
    E4  = Failure level.
    E5  = Lag time (Traveling time).
    E6  = Duration time (Failure time).
    E7  = Alert degradation time.
    E8  = Team ID.
    E9  = Request ID.
    E10 = Site ID.
    E11 = IDSUB.
    E12 = Initial status.
    E13 = Total off-alert time.
    E14 = Sequence number of event.

Name  = PQUE used for preventive maintenance events. One subscript. Ranked on Pl.
Owner = SITES.
Member = PM
   P1 = Preventive maintenance request time.
P2 = System requesting PM.
P3 = Regular or exogenous PM flag.
P4 = Request ID.
P5 = Site ID.
P6 = Status code.

Name = HQE used for overhaul events. One subscript. Ranked on H1.
Owner = SITES.
Member = OH
   H1 = Overhaul request time.
H2 = System requesting overhaul.
H3 = Team ID.
H4 = Request ID.
H5 = Site ID.
H6 = Status code.

Status code
STAT = 1-100 are alpha for blank/00, 01-99.
   101 is alpha for AA code for a continuous monitor failure.
   102 is alpha for AB code for an undetermined failure.
   103 is alpha for AD code for a demand at base control.
   104 is alpha for AT code for a team dispatch by base.
   105 is alpha for AS code for a team arrival at site.
   106 is alpha for LO code for a team lost en route.
   107 is alpha for FE code for an equipment failure at site.
   108 is alpha for XA code for maintenance completed.
   109 is alpha for ZZ code for exogenous failure.
   110 is alpha for OH code for exogenous overhaul.
   111 is alpha for FP code for a personnel failure at site.
   112 is alpha for FS code for a parts failure at site.
   113-200 are not in use.
*IRFTC MAIN

MAIN ROUTINE

.....PLANET - WEAPON SYSTEM AVAILABILITY.....

.....PURPOSE - TO REPORT OFF-ALERT STATUS.

.....INPUT - TAPE FROM ABC MODEL.

.....OUTPUT - PRINTER (SIMSCRIPT RPG).

CALL HDING

REWIND 9
1 CALL RLAL
  CALL SELECT

IF (IDSUB) EQ (  3), GO TO 3
IF (EFLAG) NE (  0), GO TO 9999
IF (IDSUB) EQ ( 100), GO TO 100
IF (IDSUB) EQ ( 110), GO TO 110
IF (IDSUB) EQ ( 200), GO TO 200
IF (IDSUB) EQ ( 500), GO TO 500
IF (IDSUB) EQ ( 600), GO TO 600
IF (IDSUB) EQ ( 800), GO TO 800
IF (IDSUB) EQ ( 900), GO TO 900
IF (IDSUB) EQ ( 925), GO TO 925
IF (IDSUB) EQ ( 950), GO TO 950
IF (IDSUB) EQ (1900), GO TO 1900
IF (IDSUB) EQ (2000), GO TO 2000
IF (IDSUB) EQ (2100), GO TO 2100
IF (IDSUB) EQ (2150), GO TO 2150
IF (IDSUB) EQ (2300), GO TO 2300
IF (IDSUB) EQ (2400), GO TO 2400
IF (IDSUB) EQ (2500), GO TO 2500

GO TO 9999

3 CALL R3
  CALL EXIT

100 CALL R100
  GO TO 9999

110 CALL R110
  GO TO 9999
200 CALL R200
    GO TO 9999
C
500 CALL R500
    GO TO 9999
C
600 CALL R600
    GO TO 9999
C
800 CALL R800
    GO TO 9999
C
900 CALL R900
    GO TO 9999
C
925 CALL R925
    GO TO 9999
C
950 CALL R950
    GO TO 9999
C
1900 CALL R1900
    GO TO 9999
C
2000 CALL R2000
    GO TO 9999
C
2100 CALL R2100
    GO TO 9999
C
2150 CALL R2150
    GO TO 9999
C
2300 CALL R2300
    GO TO 9999
C
2400 CALL R2400
    GO TO 9999
C
2500 CALL R2500
    GO TO 9999
C
9999 IF (MORE) EQ (0), GO TO 1
    CALL ROTL
    GO TO 9999
C
END
SUBROUTINE RLSL

C..........READS S-PHASE TAPJ (9) (IN MIND)....LABEL RECORDS.
C
LET NFLAG = 0
C
READ (9) II,12,13,14,15,16,17,18,19,110,111,112
C
STORE II IN IDSUR
STORE 12 IN IDSUM
STORE 13 IN SHFT
STORE 14 IN DAYW
STORE 15 IN SNSW
STORE 16 IN EBAS
STORE 17 IN VA
STORE 18 IN VN
STORE 19 IN VC
STORE 110 IN THSM
STORE 111 IN MORE
STORE 112 IN ETIMF
C
IF (TIME) OR (TMEND), GO TO 1
C
RETURN
C
1 CALL R3
CALL EXIT
C
1ND
*IRFTC RUTL

SUBROUTINE RUTL

C
C unwittingly reads 5+PHASE Tape(9) (BIN mode).....DETAIL RECORDS.
C
X READ (9) 11,12,13,14,15,16,17,18,19,110
C
STORE 11 IN DTLV1
STORE 12 IN DTLV2
STORE 13 IN DTLV3
STORE 14 IN DTLV4
STORE 15 IN DTLV5
STORE 16 IN DTLV6
STORE 17 IN DTLV7
STORE 18 IN DTLV8
STORE 19 IN DTLV9
STORE 110 IN DTLV0
C
LET MORE = MORE - 1
C
RETURN
END

*IRFTC SELECT

SUBROUTINE SELECT

C
C
C unwittingly selects EVENTS BY BASE.
C
C
DO TO 1, FOR EACH BASES I
IF (EBAS) EQ (BASE(I)), GO TO 2
1 LOOP
LET HFLAG = i
GO TO 9999
C
2 LET RFLAG = 0
GO TO 9999
C
9999 RETURN
END
*IBFTC R3

SUBROUTINE R3

..PURPOSE - TO CLOSE-OUT AND END R-PHASE.

..IDSUR = 3

CALL HDING
LET I LINES = 0
1 DO TO 2, FOR EACH IS IN SQUE
   CALL REPORT(IS, I LINES)
2 REPEAT 1

CALL POMD

3 DO TO 6, FOR EACH IS IN SQUE
4 DO TO 5, FOR EACH IP IN POUT(IS)
   CALL PQRPG(IS, IP)
5 REPEAT 4
6 REPEAT 3

CALL HOMD

7 DO TO 10, FOR EACH IS IN SQUE
8 DO TO 9, FOR EACH IH IN HQUE(IS)
   CALL HURPG(IS, IH)
9 REPEAT 8
10 REPEAT 7

REWIND 9

RETURN
END
SUBROUTINE RI00

C

C....PURPOSE - ALERT DEMAND ARRIVAL AT BASE CONTROL.
C

C....IOSUB = 100.
C

C

IF (MORF) EQ (0), GO TO 9999
CALL RDIL
C

FIND FIRST, FOR EACH M IN SWUF, WITH (SID(M)) EQ (DILV1),
WHERE IS, IF NONE, GO TO 9999
C

FIND FIRST, FOR EACH N IN EQUE(IS), WITH (EID(N)) EQ (TRSM),
WHERE IF, IF NONE, GO TO 9999
C

CREATE EVENT CALLED E
C

STORE ETIME IN E1(E)
STORE E3(E) IN E3(E)
STORE E4(E) IN E4(E)
STORE IOSUB IN E11(E)
STORE STAT(103) IN E2(E)
C

FILE E IN EQUE(IS)
C

9999 RETURN
END
*IBFTC R110

SUBROUTINE R110

C
C
C......PURPOSE - EXOG. FAILURE, EXOG. PM, EXOG. OVERHAUL.
C
C......IDSUB = 110.
C
C
C       FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
C          XWHERE IS, IF NONE, GO TO 9999
C
C       IF (VC) EQ (2), GO TO 2
C 1: IF (VC) EQ (4), GO TO 4
C 2: IF (VC) EQ (6), GO TO 6
C       GO TO 9999
C
C
C...........EXOG. FAILURE.
C
C
C       2  DO TO 20, FOR EACH FLVLs 1
C 3: IF (VBI) EQ (FLVL(1)), GO TO 21
C 4: 20 LOOP
C       GO TO 9999
C
C 21 LET NFAIL(IS) = NFAIL(IS) + 1
C       CREATE EVENT CALLED F
C       STORE ETIME IN E1(E)
C       STORE VA IN E3(E)
C       STORE VR IN E4(E)
C       STORE TRSM IN E10(E)
C       STORE IDSUB IN E11(E)
C       FILF E IN FWUF(IS)
C       GO TO 9999
C
C
C...........EXOG. PM.
C
C
C       4 CREATE PM CALLED P
C       STORE ETIME IN P1(P)
C       STORE VA IN P2(P)
C       STORE IDSUB IN P3(P)
C       STORE TRSM IN P5(P)
C       FILE P IN PQUE(IS)
C       GO TO 9999
C
C
C...........EXOG. OVERHAUL.
C
C
C       6 CREATE ON CALLED M
C       STORE ETIME IN M1(H)
C       STORE VA IN M2(H)
C       STORE TRSM IN M5(H)
C       FILE M IN MWUE(IS)
C       GO TO 9999
C
C 9999 RETURN

FND
*IBFTC R200

SUBROUTINE R200

C
C......PURPOSE - MAINTENANCE COMPLETED.
C
C......IDSUB = 200.
C
C IF (MORF) EQ (0), GO TO 9999
CALL RDRL
C
FIND FIRST, FOR EACH M IN SQUF, WITH (SID(M)) EQ (DULV1),
XWHERE IS, IF NONF, GO TO 9999
C
FIND FIRST, FOR EACH N IN EQUE(IS), WITH (EB(N)) EQ (TRSM),
XWHERE IE, IF NONE, GO TO 9999
C
CREATE EVENT CALLED E
C
STORE ETIME IN E1(E)
STORE STAT(IN) IN E2(E)
STORE E3(IC) IN E3(E)
STORE E4(IE) IN E4(E)
STORE IDSUB IN E11(E)
C
LET E8(IE) = 0
LET E9(IE) = 0
LET F10(IF) = 0
C
LET F6(IF) = ETIME - E11(E)
C
FILE E IN EQUE(IS)
C
LET NFAL(IS) = NFAL(IS) - 1
C
IF (NFAL(IS)) EQ (0), GO TO 99
GO TO 9999
C
99 CALL REPORT(IS, ILINES)
C
9999 RETURN
END
SUBROUTINE R500

*IPFTC R500

C

C....PURPOSE - TEAM DISPATCH I/Y BASE CONTROL.

C

C....IDSUB = 500.

C

C IF (MORE) EQ (0), GO TO 9999
CALL RDTL
C
C IF (DTLV4) EQ (MSITE), GO TO 9999
C
C FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (DTLV1),
WHERE IS, IF NONE, GO TO 9999
C
C FIND FIRST, FOR EACH N IN FCUF(IS), WITH (E9(N)) EQ (VC),
WHERE IE, IF NONE, GO TO 1
C
C CREATE EVENT CALLED F
C
STORF DTLV9 IN E1(F)
STORF E3(IF) IN E3(E)
STORF E4(IE) IN E4(E)
STORF TRSM IN E8(F)
STORF IDSUB IN F11(E)
STORF STAT(104) IN F21(E)
C
FILE E IN EQUE(IS)
C
STORF TRSM IN E11(F)
C
GO TO 9999
C
C IF (MORE) EQ (0), GO TO 9999
C
C FIND FIRST, FOR EACH M IN HQUE(IS), WITH (H4(M)) EQ (VC),
WHERE IH, IF NONE, GO TO 9999
C
C STORE TRSM IN H3(1H)
C
GO TO 9999
C
9999 RETURN
END
*IAFRC R600
SUBROUTINE R600

**....PURPOSE** - TEAM ARRIVAL AT SITE.

**.....IDSUB** = 600.

FIND FIRST, FOR EACH $M$ IN SQUF, WITH $(S1U(M))$ EQ (VA),
WHEREF IS; IF none, GO TO 9999

IF (VA) EQ (6), GO TO 6

FIND FIRST, FOR EACH $N$ IN EQUE(IS), WITH $(E8(N))$ EQ (TRSM),
WHEREF IE; IF none, GO TO 9999

LET TLAG = ETIME - E11(IF)

LET $E8(IE) = 0$
LET $E9(IE) = 0$
LET $E10(IE) = 0$

FIND FIRST, FOR EACH $N$ IN EQUE(IS), WITH $(E8(N))$ EQ (TRSM),
WHEREF IE; IF none, GO TO 9999

LET $F5(IE) = E5(IE) + TLAG$

CREATE EVENT CALLED $E$

STORE ETIME IN E11(IF)
STORE $E3(IE)$ IN F3(IE)
STORE $E4(IE)$ IN E4(F)
STORE IDSUB IN F11(IE)
STORE STAT(105) IN F2(F)

FILE $E$ IN EQUE(IS)

GO TO 9999

6 FIND FIRST, FOR EACH $M$ IN HOUE(IS), WITH $(H3(M))$ EQ (TRSM),
WHEREH IM; IF none, GO TO 9999

LET $NFALI(IS) = NFALI(IS) + 1$

CREATE EVENT CALLED $F$

STORE ETIME IN F11(E)
STORE $H6(IM)$ IN E2(F)
STORE $H2(IM)$ IN E3(E)
STORE TRSM IN E8(F)
STORE $H4(IM)$ IN E9(E)
SUBROUTINE R900

... PURPOSE - EQUIP. FAILURE AT SITE.
... IODSUB = 900.

FIND FIRST, FOR EACH M IN SCUE, WITH (SID(M)) EQ (VA),
XWHERE IS, IF NONE, GO TO 9999

FIND FIRST, FOR EACH N IN EQUE(IS), WITH (EB(N)) EQ (TRSM),
XWHERE IE, IF NONE, GO TO 9999

CREATE EVENT CALLED E

STORE ETIME IN E1(E)
STORE E3(IIE) IN E3(E)
STORE E4(IIE) IN E4(E)
STORE IODSUB IN E11(E)
STORE STAT(107) IN E2(E)

FILE E IN EQUE(IS)

9999 RETURN
END

SUBROUTINE R925

... PURPOSE - PERS. FAILURE AT SITE.
... IODSUB = 925.

FIND FIRST, FOR EACH M IN SCUE, WITH (SID(M)) EQ (VA),
XWHERE IS, IF NONE, GO TO 9999

FIND FIRST, FOR EACH N IN EQUE(IS), WITH (EB(N)) EQ (TRSM),
XWHERE IE, IF NONE, GO TO 9999

CREATE EVENT CALLED E

STORE ETIME IN E1(E)
STORE E3(IIE) IN E3(E)
STORE E4(IIE) IN E4(E)
STORE IODSUB IN E11(E)
STORE STAT(111) IN E2(E)

FILE E IN EQUE(IS)

9999 RETURN
END
*IRFTC R950

SUBROUTINE R950

C

C.....PURPOSE - PARTS FAILURE AT SITE.

C

C.....IDSUK = 950.

C

C     FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (VA),
C     WHERE IS, IF NONE, GO TO 9999

C

C     FIND FIRST, FOR EACH N IN EQU(IS), WITH (E8(N)) EQ (THSM),
C     WHERE IE, IF NONE, GO TO 9999

C

C     CREATE EVENT CALLED E

C

C     STORE FTIME IN E1(E)
C     STORE E3(E) IN E3(E)
C     STORE E4(E) IN E4(E)
C     STORE IDSUB IN E11(E)
C     STORE STAT(112) IN E2(E)

C

C     FILE E IN FQUF(IS)

C

9999 RETURN

END
*IRFTC R1900

SUBROUTINE R1900

C
C
C.-----PURPOSE - GENERATE SITES AT BASE.
C
C
C.-----IDSUP = 1900.
C
C
IF (MORF) EQ 10), GO TO 9999
CALL R01L
C
CREATE SITES CALLED S
C
LET I = 1 + 1
C
STORE TKSM IN SID(S)
STORE VC IN SMODE(S)
STORE DTLV1 IN SERNO(S)
STORE FHAS IN TBASE(S)
STORE I IN SIND(S)
C
IF (VC) GE IMSITE), LET MSITE = VC + 1
C
FILE S IN SQUE
C
9999 RETURN
END

*IBFTC R2000

SUBROUTINE R2000

C
C
C.-----PURPOSE - REQUEST FOR PM.
C
C
C.-----IDSUP = 2000.
C
C
FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TKSM),
WHERE IS, IF NONE, GO TO 9999
C
CREATE PM CALLED P
C
STORE FTIME IN PL(P)
STORE VH IN P2(P)
STORE TKSM IN P5(P)
C
FILE P IN PWUFI(S)
C
9999 RETURN
END
*IRFTC R2100
SUBROUTINE R2100
C
C......PURPOSE - BEGIN PM.
C
C......IDSUB = 2100.
C
C
FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
XWHERE IS, IF NONE, GO TO 9999
C
FIND FIRST, FOR EACH N IN PQUL(IS), WITH (P5(N)) EQ (TRSM),
XAND (IP2(N)) EQ (VB), WHERE IP, IF NONE, GO TO 9999
C
LET NFAIL(IS) = NFAIL(IS) + 1
C
CREATE EVENT CALLED E
C
STORE ETIME IN E1(E)
STORE P6(IP) IN F2(E)
STORE P2(IP) IN F3(E)
STORE VC IN ERA(E)
STORE P4(IP) IN E9(E)
STORE P5(IP) IN E10(E)
STORE IDSUB IN F11(E)
C
IF (P3(IP)) EQ (110), GO TO 110
GO TO 9998
C
110 STORE P3(IP) IN E11(E)
GO TO 9998
C
9998 FILE E IN EQUE(IS)
C
REMOVE IP FROM PQUL(IS)
DESTROY PM CALLED IP
C
9999 RETURN
END
*IMFTC K2150  
SUBROUTINE K2150  
C  
C.....PURPOSE - FAILURE CAUSED BY PM.  
C  
C.....idosub = 2150.  
C  
DO TO 1, FOR EACH FLVL 1  
IF (VA) EQ (FLVL(1)), GO TO 2  
1 LOOP  
GO TO 9999  
C  
2 FIND FIRST, FOR EACH M IN SWUF, WITH (SID(M)) EQ (TRSM),  
WHERE IS, IF NONE, GO TO 9999  
C  
LET NFAIL(IS) = NFAIL(IS) + 1  
C  
CREATE EVENT CALLED F  
C  
STORE (TIME IN OUT)  
STORE VM IN F3(F)  
STORE VA IN T4(E)  
STORE TRSM IN E10(F)  
STORE IUSDU IN L11(F)  
C  
FILE E IN EUUF(IIS)  
C  
9999 RETURN  
+ND
*IPFIC R2300

SUBROUTINE R2300

C.....PURPOSE - ALERT - CONTINUOUS MONITOR.
C
C.....IUSTR = 2300.
C
DO TO 1, FOR EACH FLVLS 1
   IF (VA) EQ (FLV(L)), GO TO 2
1 LOOP
   GO TO 9999
C
2 FIND FIRST, FOR EACH M IN SOUR, WITH (SID(M)) EQ (TRSM),
   WHERE IS, IF NONE, GO TO 9999
C
   LET NFAIL(IS) = NFAIL(IS) + 1
C
   CREATE EVENT CALLED E
C
   STORE ETIME IN E1(E)
   STORE VA IN E3(E)
   STORE VA IN E4(E)
   STORE TRSM IN E10(E)
   STORE IUSUR IN E11(E)
C
   FILE E IN FOUE(IS)
C
9999 RETURN
   FND
*IRFTC R2400

SUBROUTINE R2400

C

C......PURPOSE - RESOURCE REQUEST.
C
C
C......IDSUR = 2400.
C
C IF (MORE) EQ (0), GO TO 9999
CALL RDTL
LET DTLV2 = DTLV2 + 1
C
FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C
IF (DTLV1) EQ (1), GO TO 1
IF (DTLV1) EQ (2), GO TO 2
IF (DTLV1) EQ (3), GO TO 3
IF (DTLV1) EQ (4), GO TO 4
IF (DTLV1) EQ (5), GO TO 5
IF (DTLV1) EQ (6), GO TO 6
GO TO 9999
C
1 FIND FIRST, FOR EACH N IN EQUI(S), WITH (E11(N)) EQ (2300),
XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VH), WHERE IE, IF NONE,
XGO TO 9999
C
STORE STAT(101) IN E2(IE)
STORE VC IN E9(IE)
GO TO 9999
C
2 FIND FIRST, FOR EACH N IN EQUI(S), WITH (E11(N)) EQ (110),
XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
XGO TO 9999
C
STORE STAT(109) IN E2(IF)
STORE VC IN F9(IF)
GO TO 9999
C
3 FIND FIRST, FOR EACH N IN EQUI(S), WITH (E11(N)) EQ (2500),
XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE IE, IF NONE,
XGO TO 9999
C
STORE STAT(102) IN E2(IE)
STORE VC IN E9(IE)
C
CREATE EVENT CALLED E
C
STORE ETIME IN E1(E)
STORE STAT(DTLV2) IN E2(EIF)
STORE VA IN F3(EIF)
STORE VA IN F4(F)
STORE IOSUB IN F11(F)
C FILE E IN EQUE(I5)
C GO TO 9999
C
4 FIND FIRST, FOR EACH N IN POUE(I5), WITH (P5(N)) EQ (TRSM),
XAND (P2(N)) EQ (VB), WHERE TP, IF NONE, GO TO 9999
C STORE VC IN P4(IP)
LET VR = VB + 1
STORE STAT(VB) IN P6(IP)
GO TO 9999
C
5 FIND FIRST, FOR EACH N IN EQUE(I5), WITH (E11(N)) EQ (2150),
XAND (E10(N)) EQ (TRSM), AND (E3(N)) EQ (VB), WHERE I'T, IF NONE,
XGO TO 9999
C STORE STAT(DTLV2) IN F2(IF)
STORE VC IN F9(IE)
GO TO 9999
C
6 FIND FIRST, FOR EACH N IN HQUE(I5), WITH (H5(N)) EQ (TRSM),
XAND (H2(N)) EQ (VB), WHERE IH, IF NONE, GO TO 9999
C STORE VC IN H4(1H)
STORE STAT(110) IN H6(1H)
GO TO 9999
C
9999 RETURN
END
SUBROUTINE R2500

*IBFTC R2500

C

C.......PURPOSE - UNDETERMINED FAILURE.
C

C.......(IDSUB = 2500).
C

C

GO TO 1, FOR EACH FLVLS I
IF (VA) EQ (FLVL(I)), GO TO 2
1 LOOP
GO TO 9999

C

2 FIND FIRST, FOR EACH M IN SQUE, WITH (S1)(M)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C

LET NFAIL(IS) = NFAIL(IS) + 1
C

CREATE EVENT CALLED F

STORE ETIME IN FI(E)
STORE VM IN E3(E)
STORE VA IN E4(F)
STORE TRSM IN E10(F)
STORE IDSUB IN F1111)
C

FILE F IN EQUF(IS)
C

9999 RETURN
END
*IRITC REPORT
SUBROUTINE REPORT(IS,ILINES)

C
C
C..PURPOSE - TO REPORT THE EVENTS OF A SITE FOR OFF-ALERT STATUS.
C
C
C..CALLED BY R200 OR R3.
C
C
IF EOUF(IS) IS EMPTY, GO TO 9999
C
STORE E?IFOQUE(IS)) IN E12(IFOQUE(IS))
LET E13(IFOQUE(IS)) = E1(IFOQUE(IS)) - F(I(IFOQUE(IS))

1 DD TO 2, FOR EACH IF IN FQUE(IS)
LET ISEQ = ISEQ + 1
LET F14(IF) = ISEQ
IF (F12(IF)) EQ (0), LET F12(IF) = STAT(1)
2 REPEAT 1
LET ISEQ = 0
C
CALL ALDGE(IS)
C
10 IF EOUF(IS) IS EMPTY, GO TO 9999
LET IE = FQUE(IS)
C
LET CTIME = F1(IF)
CALL CNVRT(CTIME,ID,IH,IM)
LET ID1 = ID
LET IH1 = IH
LET IM1 = IM
C
LET CTIME = E13(IE)
CALL CNVRT(CTIME,ID,IH,IM)
LET ID2 = ID
LET IH2 = IH
LET IM2 = IM
C
LET CTIME = F5(IF)
CALL CNVRT(CTIME,ID,IH,IM)
LET ID3 = ID
LET IH3 = IH
LET IM3 = IM
C
LET CTIME = F6(IF)
CALL CNVRT(CTIME,ID,IH,IM)
LET ID4 = ID
LET IH4 = IH
LET IM4 = IM
C
LET CTIME = F7(IF)
CALL CNVRT(CTIME,ID,IH,IM)
LET ID5 = ID
LET IM6 = IM
LET IMS = IM
C
LET ILLINES = ILLINES + 1
IF ILLINES) LS (55), GO TO 20
CALL H0ING
LET ILLINES = C
C
20 CALL FORM(IF, IS, ID1, IM1, IM2, IM2, ID3, IM3, IMS, IMS, IMS, IMS, IMS)
C
REMOVE IE FROM FOUR(1S)
DESTROY EVENT CALL FOR IE
GO TO 10
C
9948 CALL BLANK
LET ILLINES = ILLINES + 1
C
9999 RETURN
END
*18FTC ALDEG

SURROUTINE ALDFG(IS)

C
C
C.....PURPOSE - TO CALCULATE ALERT DEGRADATION TIME FOR A SITE.
C
C
C.....CALLED BY REPORT.

C

LET LFAIL = 0
LET LFIX = 0
LET PTIME = 0.0
LET IF14 = 0

C 1 (II) TO 2, FOR EACH IE IN EQUE(IS), WITH (E14(IE)) NE (0)
  IF (E11(IE)) EQ (2500), GO TO 3
  IF (E11(IE)) EQ (2100), GO TO 3
  IF (E11(IE)) EQ (2300), GO TO 3
  IF (E11(IF)) EQ (2150), GO TO 3
  IF (E11(IF)) EQ (110), GO TO 3
  LET E14(IE) = 0
  2 REPEAT 1
C

GO TO 999

C 3 LET IFail = IF
  FIND FIRST, FOR EACH M IN EQUE(IS), WITH (E11(M)) EQ (200), AND
  X(E3(M)) EQ (F3(IFAIL1)), AND (E4(M)) EQ (F4(IFAIL1)), AND (E14(M))
  XGE (LFIX), WHERE 1200, IF MINE, GO TO 40
C
  IF (LFAIL) EQ (0), GO TO 20
  LET F7(IFAIL) = E11(IF00) - PTIME
  GO TO 30
C
  20 LET F7(IFAIL) = E11(IF00) - E11(IFAIL)
C
  30 LET LFIX = E11(IF00)
  LET PTIME = E11(IF00)
C
  40 LET IFAIL = E14(IFAIL)
  LET E14(IF) = 0
  GO TO 2
C
  999 (U TO 9999, FOR EACH IE IN EQUE(IS))
  LET IE14 = IE14 + 1
  LET E14(IF) = IE14
  9999 REPEAT 999
C

RETURN
END
*IBFTC CNVRT
SUBROUTINE CNVRT(CTIMF, ID, IH, IM)
C
C......PURPOSE - TO CONVERT DECIMAL DAYS TO DAYS(ID), HOURS(IH), MIN(SIM)
C
C......CALLED BY REPORT.
C
C
LET ID = MPART(CTIMF)
LET IH = MPART(CTIMF)
LET IM = MPART(CTIMF)
IF (IM) NE (60), GO TO 1
LET IH = IH + 1
LET IM = 0
1 IF (IH) LS (24), GO TO 9999
LET IH = IH - 24
LET ID = ID + 1
C
9999 RETURN
END

*IBFTC HDING
REPORT HDING,
F EVENT FAIL SITE INIT TOTAL EVENT UNIT FAIL
F TIME NO TYPE SQUAD STAT OFF-ALERT STAT FAIL LEV
F END

END

*IBFTC FORM
REPORT FORM(1,15,1D1,1H1,1M1,1D2,1H2,1M2,1D3,1H3,1M3,1H4,1M4,
1D1,1H1,1M1,1H2,1M2,1D3,1H3,1M3,1H4,1M4,

1D1,1H1,1M1,1H2,1M2,1D3,1H3,1M3,1H4,1M4,
1D1,1H1,1M1,1H2,1M2,1D3,1H3,1M3,1H4,1M4,
1D1,1H1,1M1,1H2,1M2,1D3,1H3,1M3,1H4,1M4,
1D1,1H1,1M1,1H2,1M2,1D3,1H3,1M3,1H4,1M4,

END

END
FND
FND

REPORT PQHD
PM EVENT TAIL EVNT UNIT
TIME NO STAT FAIL R1D S1D

END

REPORT PORPG(I5,IP)
PM EVENT TAIL EVNT UNIT
TIME NO STAT FAIL R1D S1D

END

REPORT HQHD

END

REPORT HQRG(I5,IH)

END
<table>
<thead>
<tr>
<th>ENTRY</th>
<th>MAIN</th>
<th>94</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>26</td>
<td>O</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>R</td>
<td>10</td>
</tr>
<tr>
<td>28</td>
<td>O</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>R</td>
<td>200</td>
</tr>
</tbody>
</table>
Program 3

LOGISTICS RESOURCE UTILIZATION
III. LOGISTICS RESOURCE UTILIZATION

The logistics information is presented in three parts: Spares, Personnel, and Equipment. The spare parts data presented by this report contain information regarding stock levels, NORS (not operationally ready--supply) time, NORS count, and demand quantities. The personnel report contains the man-hour accounting information. This report displays the utilization factors for each personnel type and the man-hours consumed by various tasks. The equipment data presented contain information regarding the utilization of maintenance equipment and facilities. NORE (not operationally ready--equipment) time, NORE count, and demand quantities are included.

The Logistics Resource Utilization report can be used for either aircraft or missile simulations.

Fig. 11 is a display of some example data showing the spare parts usage for the period* ending day 21.00 of the simulation. Column 1 lists each part type specified in the input data of the simulator. Column 2 lists the authorized quantity, also specified in the inputs of the simulator. Column 3 lists, for each spare type, the quantity of serviceable items remaining in stock at report time. Column 4 lists the number of demands for each spare part during the reporting period. Column 5 is the total demands for each spare part. Column 6 is the NORS (not operationally ready--supply) time during the reporting period. NORS time in this report is treated independently of the other NOR conditions, i.e., this is the maximum amount of time that could be attributed to Supply. Column 7 is the accumulation of all NORS time to date. Column 8 lists, for each spare part, the number of stockouts that occurred during the period. Column 9 lists the total number of stockouts accumulated to date.

Figure 12 is a display of some example data showing the man-hour accounting information for the period* ending day 21.00 of the simulation. Column 1 lists each personnel type specified in the input data of the simulator. Column 2 is the man-shifts (number of personnel x number of

*The period is an initialized variable.
### SPARE PARTS USAGE FOR PERIOD ENDING 21.00

<table>
<thead>
<tr>
<th>TYPE</th>
<th>AUTH. QTY</th>
<th>TOTAL QTY AVAILABLE</th>
<th>SITE DEMANDS TO DATE</th>
<th>SITE DEMANDS THIS PERIOD</th>
<th>MORES TIME TO DATE</th>
<th>MORES TIME THIS PERIOD</th>
<th>NO. MORES THIS PER</th>
<th>NO. MORES TO DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>18</td>
<td>10</td>
<td>40</td>
<td>0.09</td>
<td>0.09</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>1</td>
<td>8</td>
<td>21</td>
<td>95.21</td>
<td>95.21</td>
<td>92</td>
<td>93</td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>16</td>
<td>40</td>
<td>130</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>34</td>
<td>9</td>
<td>29</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 11 -- Spare Parts Usage

### MANHOUR ACCOUNTING FOR PERIOD ENDING 21.00

<table>
<thead>
<tr>
<th>TYPE</th>
<th>MAN SHFT</th>
<th>TOTAL HRS AVAILABLE</th>
<th>TOTAL HRS TO DATE</th>
<th>FLY LINE MAINT HRS</th>
<th>INTRAN HOURS</th>
<th>OVERTIME HOURS</th>
<th>UTIL FACTOR THIS PERIOD</th>
<th>UTIL FACTOR TO DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1052</td>
<td>0416.00</td>
<td>2533.00</td>
<td>8.06</td>
<td>2.73</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>427</td>
<td>3376.00</td>
<td>10117.00</td>
<td>1.37</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>338</td>
<td>2704.00</td>
<td>8048.00</td>
<td>7.56</td>
<td>0.31</td>
<td>1.01</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>842</td>
<td>6796.00</td>
<td>20192.00</td>
<td>15.55</td>
<td>3.09</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>212</td>
<td>1696.00</td>
<td>5072.00</td>
<td>14.12</td>
<td>0.47</td>
<td>0.81</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>420</td>
<td>3960.00</td>
<td>10060.00</td>
<td>9.98</td>
<td>0.42</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Fig. 12 -- Man-hour Accounting

### MAINT. EQUIPMENT USAGE FOR PERIOD ENDING 21.00

<table>
<thead>
<tr>
<th>TYPE</th>
<th>AUTH. QTY</th>
<th>TOTAL QTY AVAILABLE</th>
<th>SITE DEMANDS TO DATE</th>
<th>SITE DEMANDS THIS PERIOD</th>
<th>MORES TIME TO DATE</th>
<th>MORES TIME THIS PERIOD</th>
<th>NO. MORES THIS PER</th>
<th>NO. MORES TO DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>50</td>
<td>92</td>
<td>218</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>20</td>
<td>18</td>
<td>61</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>16</td>
<td>8</td>
<td>21</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>40</td>
<td>1</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 13 -- Maintenance Equipment Usage
shifts that the personnel are assigned) available. Column 3 lists, for each personnel type, the total man-hours (man-shifts x shift duration) available. Column 4 lists the accumulation of the total man-hours available to date. Column 5 lists the maintenance man-hours (decimal hours) for each personnel type. Column 6 lists, for each personnel type, the man-hours spent in transit to and from the site. Column 7 lists the overtime required. Column 8 displays the utilization of each personnel type (Σ Col. 5, 6, 7 ÷ Col. 3). Column 9 is the accumulation of the utilization factor for the simulation to date (Σ all Col. 5, 6, and 7's ÷ Col. 4).

Figure 13 is a display of some example data showing the maintenance equipment usage for the period ending day 21.00. Column 1 lists each equipment type specified in the input data of the simulator. Column 2 lists the authorized quantity, also specified in the inputs of the simulator. Column 3 lists, for each equipment type, the quantity available for use at report time. Column 4 lists the number of demands for each equipment type during the reporting period. Column 5 is the total demands for each equipment type. Column 6 is the MORE (not operationally ready--equipment) time during the reporting period. MORE time is treated independently of other MORE conditions, i.e., this is the maximum amount of time that could be attributed to Equipment if there were no other MORE conditions. Column 7 is the accumulation of all MORE time to date. Column 8 lists, for each equipment type, the number of demands that could not be satisfied immediately because of a shortage of equipment. Column 9 lists the total number of MORE conditions to date.

**INITIALIZATION**

Table 3 lists the initialization requirements. Only six arrays require inputs. Array 23 is the report interval for the man-hour accounting report. This report interval must be initialized either daily or weekly. Arrays 24 and 25 are the reporting interval for Spares and Equipment and can be initialized to any floating point (decimal)

*The period is an initialized variable.*
Table 3

VARIABLE DESCRIPTION AND INITIALIZATION:
LOGISTICS RESOURCE UTILIZATION

<table>
<thead>
<tr>
<th>Array Number</th>
<th>Number of Attributes</th>
<th>Mode</th>
<th>Initialization Method</th>
<th>Initialization Value in Fig. 5 Col.</th>
<th>Description of Variable to Be Initialized</th>
<th>Permanent Location</th>
<th>Variable Name</th>
<th>Entity, Subentity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Reporting interval for general accounting (Must be 1 0000 to 7 0000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Reporting interval for maintenance equipment usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Reporting interval for spare parts usage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13-20</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Hours per shift (25 6 00000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-30</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Hours of house to be analyzed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Specific time house must be analyzed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41-50</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td>10</td>
<td>Specific time house must be analyzed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
number. Array 29 specifies the duration of the work shift (floating point) for use with the man-hour accounting report. Arrays 33 and 34 specify the number of bases to be contained in the report and the bases(s) number(s). The remaining arrays are zeroed.

OUTPUT PROGRAM

The input to this program is the tape generated by the ABC Simulator.

The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of one or more bases may be run at one time.

When a label record is read, ETMD is compared to RT10, RT20, and RT30. If ETMD is greater than or equal to RT10, RT20, or RT30, then RPT10, RPT20, and/or RPT30 is called and the reports are displayed.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-PR).

IDSO = Ids
LDUB = Idd
SHFT = Shift
DAYW = Dw/wk
SKDW = S/wk
EBAS = Base no.
VA = Variable-1
VB = Variable-2
VC = Variable-3
TRMN = ID Addresses
MORE = Dri
ETMD = Event time
Detail records (see page 109 of RM-4659-PR).

- \[\text{DTLV1} = \text{Integer variable } 1\]
- \[\text{DTLV2} = \text{Integer variable } 2\]
- \[\text{DTLV3} = \text{Integer variable } 3\]
- \[\text{DTLV4} = \text{Integer variable } 4\]
- \[\text{DTLV5} = \text{Integer variable } 5\]
- \[\text{DTLV6} = \text{Integer variable } 6\]
- \[\text{DTLV7} = \text{Integer variable } 7\]
- \[\text{DTLV8} = \text{Integer variable } 8\]
- \[\text{DTLV9} = \text{Float variable } 1\]
- \[\text{DTLV0} = \text{Float variable } 2\]

- \[\text{T10} = \text{RT10 increments for Personnel display.}\]
  \[\text{Must be 1.00000 or 7.00000.}\]
- \[\text{T20} = \text{RT20 increments for equipment display.}\]
  \[\text{No restrictions.}\]
- \[\text{T30} = \text{RT30 increments for spare parts display.}\]
  \[\text{No restrictions.}\]
- \[\text{SHIFT} = \text{Hours per shift.}\]
- \[\text{BASES} = \text{Number of base codes to be processed.}\]
- \[\text{BASE} = \text{Base codes to be processed.}\]
- \[\text{BFLAG} = \text{Controls flow of events as a result of \text{EBAS} vs. \text{BASES}.}\]

**SETS**

- \[\text{Name} = \text{LQ10 used for display of Personnel. No subscripts.}\]
  \[\text{Ranked on L11.}\]
- \[\text{Owner} = \text{SIMSCRIPT system.}\]
- \[\text{Member} = \text{L10} \quad \text{(Time is in decimal hours).}\]
  \[\text{L11 = Type.}\]
  \[\text{L12 = Man-shift available.}\]
  \[\text{L13 = Total man-hours this period.}\]
  \[\text{L14 = Total man-hours to date.}\]
  \[\text{L15 = Flight-line maintenance hours this period.}\]
  \[\text{L15A = Flight-line maintenance hours to date.}\]
  \[\text{L16 = Not used.}\]
  \[\text{L17 = Intransit hours for this period.}\]
Li7A = Intransit hours to date.
Li8 = Overtime hours for this period.
Li8A = Overtime hours to date.
Li9 = Utilization factor for this period.
Li10 = Utilization factor to date.

Name = LQ20 used for display of equipment. No subscripts. Ranked on L21.
Owner = SIMSCRIPT system.
Member = L20 (Time is in decimal days).
   L21 = Type.
   L22 = Authorized quantity.
   L23 = Total quantity available.
   L24 = Site demands this period.
   L25 = Site demands to date.
   L26 = NORE time this period.
   L27 = NORE time to date.
   L28 = Number of NORE this period.
   L29 = Number of NORE to date.

Name = LQ30 used for display of spares. No subscripts. Ranked on L31.
Owner = SIMSCRIPT system.
Member = L30 (Time is in decimal days).
   L31 = Type.
   L32 = Authorized quantity.
   L33 = Total quantity available.
   L34 = Site demands this period.
   L35 = Site demands to date.
   L36 = NORS time this period.
   L37 = NORS time to date.
   L38 = Number of NORS this period.
   L39 = Number of NORS to date.

Name = PQUE used to queue personnel. No subscripts. Ranked on TP10.
Owner = SIMSCRIPT system.
Member = P10.
   TP10 = Type.
   SP10 = Shift.
   QP10 = Quantity.
   DP10 = Day.
   PFLG = Control flag to compute day.

Name = PQ12 used to queue exogenous personnel. No subscripts.
       Ranked on TP12.
Owner = SIMSCRIPT system.

Member = P12.
   TP12 = Type.
   SP12 = Shift
   QP12 = Quantity.
   DP12 = Day.

Name = BQUE used for team resources. No subscripts. Ranked
       on RTID. RQUE used for team resources. One subscript.
       Ranked on TTID.
Owner = BQUE is SIMSCRIPT system.
       RQUE is Team.

Member = RESEC.
   RTID = Team ID.
   RVID = Request ID.
   RQDF = Type.
   RQTY = Quantity.
   RSUB = Resource.
   RQFT = Shift.

Name = TQUR used to queue teams. No subscripts. Ranked on T500.
Owner = SIMSCRIPT system.

Member = TEAM.
   TTID = Team ID.
   TRID = Request ID.
   T500 = Time team arrived at site.
   T504 = Time team left site.
   T600 = Time team left base/site.
T700 = Time team arrived at base.
T200 = Time maintenance was completed.
T1450 = Time team was returned to base pool.
T800 = Time team lost (if applicable).
S500 = Time team left Base.

Name = NQUE used for NORS and NCE. No subscripts. Ranked on TNOR.
Owner = SIMSCRIPT system.
Member = NOR.
NTID = Team ID.
NRID = Request ID.
NTYP = Type.
NQTY = Quantity.
NSUB = NORS or NCRE.
TNOR = Time of NORS or NORE.
| LL12 | 7 |
| LL13 | 4 |
| LL14 | 5 |
| LL15 | 6 |
| LL16 | 8 |
| LL17 | 31 |
| LL17A| 32 |
| LL18 | 33 |
| LL19 | 34 |
| LL1A | 35 |
| LL1B | 36 |
| PLQ1037| |
| SLQ1037| |

**TL20 B 8**

| LL21 | 1 |
| LL22 | 2 |
| LL23 | 3 |
| LL24 | 5 |
| LL25 | 6 |
| LL26 | 7 |
| LL27 | 8 |
| LL28 | 31 |
| LL29 | 32 |
| PLQ2033| |
| SLQ2034| |

**TL30 B 8**

| LL31 | 1 |
| LL32 | 2 |
| LL33 | 4 |
| LL34 | 5 |
| LL35 | 6 |
| LL36 | 7 |
| LL37 | 8 |
| LL38 | 31 |
| LL39 | 32 |
| PLQ3033| |
| SLQ3034| |

**TP10 B**

| TP10 | 1 |
| SP10 | 2 |
| QQ10 | 3 |
| DP10 | 4 |
| PQ10 | 5 |
| SQ10 | 6 |
| PLG  | 7 |

**TP12 B**

| TP12 | 1 |
| SP12 | 2 |
*IRFTC MAIN

MAIN ROUTINE

C

C ....PLANET - LOGISTICS RESOURCE UTILIZATION....
C

C.....PURPOSE - TO REPORT (1) MANHOUR ACCOUNTING,
C (2) MAINT. EQUIPMENT USAGE,
C (3) SPARE PARTS USAGE.
C
C
C.....INPUT - TAPI FROM ABC MODEL.
C
C
C.....OUTPUT - PRINTER (SIMSCRIPT RPG).
C
C

RENEW 9

C

LET RT10 = T10
LET RT20 = T20
LET RT30 = T30

C

1 CALL RLPL
CALL SELECT

C

IF (IDSUB) EQ ( 3), GO TO 3
IF (BFLAG) NE ( 0), GO TO 9999
IF (IDSUB) EQ ( 10), GO TO 10
IF (IDSUB) EQ ( 20), GO TO 20
IF (IDSUB) EQ ( 30), GO TO 30
IF (IDSUB) EQ ( 200), GO TO 200
IF (IDSUB) EQ ( 500), GO TO 500
IF (IDSUB) EQ ( 600), GO TO 600
IF (IDSUB) EQ ( 700), GO TO 700
IF (IDSUB) EQ ( 800), GO TO 800
IF (IDSUB) EQ (1002), GO TO 1002
IF (IDSUB) EQ (1010), GO TO 1010
IF (IDSUB) EQ (1012), GO TO 1012
IF (IDSUB) EQ (1020), GO TO 1020
IF (IDSUB) EQ (1022), GO TO 1022
IF (IDSUB) EQ (1100), GO TO 1100
IF (IDSUB) EQ (1200), GO TO 1200
IF (IDSUB) EQ (1210), GO TO 1210
IF (IDSUB) EQ (1220), GO TO 1220
IF (IDSUB) EQ (1400), GO TO 1400
IF (IDSUB) EQ (1401), GO TO 1401
IF (IDSUB) EQ (1405), GO TO 1405
IF (IDSUB) EQ (1470), GO TO 1470
IF (IDSUB) EQ (1900), GO TO 1900

C

GO TO 9999
C 3 CALL #3
   CALL EXIT
C 10 CALL #10
   GO TO 9999
C 20 CALL #20
   GO TO 9999
C 30 CALL #30
   GO TO 9999
C 400 CALL R400
   GO TO 9999
C 500 CALL R500
   GO TO 9999
C 600 CALL R600
   GO TO 9999
C 700 CALL R700
   GO TO 9999
C 800 CALL R800
   GO TO 9999
C 1002 CALL R1002
   GO TO 9999
C 1010 CALL R1010
   GO TO 9999
C 1012 CALL R1012
   GO TO 9999
C 1020 CALL R1020
   GO TO 9999
C 1022 CALL R1022
   GO TO 9999
C 1100 CALL R1100
   GO TO 9999
C 1200 CALL R1200
   GO TO 9999
C 1210 CALL R1210
   GO TO 9999
C 1220 CALL R1220
SUBROUTINE RLRL

C......READS 5-PHASE TAPE(9) (PIN M113F)......LABEL RECORDS.
C
C    LET RFLAG = 0
C
C    READ (9) 11,12,13,14,15,16,17,18,19,110,111,112
C
C    STORE 11 IN IDSUR
C    STORE 12 IN IDSUR
C    STORE 13 IN SMFI
C    STORE 14 IN DAYW
C    STORE 15 IN SXOW
C    STORE 16 IN EHAS
C    STORE 17 IN VA
C    STORE 18 IN VR
C    STORE 19 IN VC
C    STORE 110 IN TRSM
C    STORE 111 IN MREF
C    STORE 112 IN ETME
C
C    1 IF (ETIME) GE (RT10), Go TO 10
C    2 IF (ETIME) LE (RT20), Go TO 20
C    3 IF (ETIME) LE (RT30), Go TO 30
C
C    RETURN
C
C    10 CALL RPT10
C            Go TO 1
C
C    20 CALL RPT20
C            Go TO 2
C
C    30 CALL RPT30
C            Go TO 3
C
C    END
*IRFTC SELECT
   SUBROUTINE SELECT
   C
   C
   C......PURPOSE - TO SELECT EVENTS BY BASE.
   C
   DO TO 1, FOR EACH BASE I
   IF (FRAS) EQ (BASE(I)), GO TO 2
   1 LOOP
      LET IFLAG = 1
      GO TO 9999
   C
   2 LET IFLAG = 0
      GO TO 9999
   C
   9999 RETURN
   END

*IRFTC RDIL
   SUBROUTINE RDIL
   C
   C........READS 5-PHASE TAPE(9) (MIN M0DF).....DETAIL RECORDS.
   C
   READ (9) 11,12,13,14,15,16,17,18,19,20
   C
   STORE 11 IN DLV1
   STORE 12 IN DLV2
   STORE 13 IN DLV3
   STORE 14 IN DLV4
   STORE 15 IN DLV5
   STORE 16 IN DLV6
   STORE 17 IN DLV7
   STORE 18 IN DLV8
   STORE 19 IN DLV9
   STORE 20 IN DLV10
   C
   LET M0NE = M0RE - 1
   C
   RETURN
   END
*IBFIE 85
  SUBROUTINE 83

C
C
C......PURPOSE - TO END RUN.
C
C
C......INSUP = 3.
C
   CALL KP110
   CALL KP120
   CALL KP130
C
   CALL KP140
C
9999 RETURN
   END

*IBFIE 810
  SUBROUTINE 810

C
C
C......PURPOSE - TO INITIALIZE PERSONNEL WIFS.
C
C
C......INSUP = 10.
C
   FIND FIRST, FOR EACH M IN LO10, WITH (L11(L10)) EQ (VA),
   XWHERE 11, IF NONE, GO TO 10
C
   GO TO 20
C
   10 CREATE L10
   STORF VA IN L11(L10)
   FILE L10 IN LCT10
   GO TO 20
C
   20 FIND FIRST, FOR EACH M IN PQUE, WITH (TP10(M)) EQ (VA),
   XAND (SP10(M)) EQ (VR), WHERE IP, IF NONE, GO TO 21
C
   LET OP10(IP) = OP10(IP) + VC
   GO TO 9999
C
   21 CREATE P10
   LET TP10(P10) = VA
   LET SP10(P10) = VR
   LET OP10(P10) = VC
C
   IF (VR) GR (MAX10), LET MAX10 = VR
C
   FILE P10 IN PQUE
   GO TO 9999
C
9999 RETURN
   END
*INTEL RIOP

SUBROUTINE RIOP

C

C......PURPOSE - TO CALCULATE MAN SHIFT AVAIL,
C       TOTAL MANKHS THIS PERIOD,
C       TOTAL MANKHS TO DATE.
C
C......CALLED BY RPT10.
C
C
IF (IFLAG) NE (0), GO TO 30
C
10 LET ISHFT = MAX(0) / 7
   LET ICAY = 1
   LET IFLAG = 1
C
20 IF TO TO 200, FOR EACH M IN POUF, WITH (PFLG(M)) EQ (0)
   IF (SPL0(M)) GR (ISHFT), GO TO 200
   LET OPL0(M) = IDAY
   LET PFLG(M) = 1
200 REPEAT 20
C
   LET ISHFT = ISHFT + (MAX0 / 7)
   IF (ISHFT) GR (MAX0), GO TO 30
   LET IDAY = IDAY + 1
   GO TO 20
C
30 IF (T10) EQ (7.0), G0 TO 37
   IF (T10) EQ (1.0), GO TO 31
   GO TO 9999
C
31 LET LOOP = LOOP + 1
   IF (LOOP) EQ (8), GO TO 310
   GO TO 311
C
310 LET LOOP = 1
   LET ILOOP = ILOOP + 7
C
311 LET JDAY = DPARK(MINT) - ILOOP
C
312 DO TO 319, FOR EACH J IN POUE, WITH (QPL0(J)) EQ (JDAY)
   FIND FIRST, FOR EACH K IN L110, WITH (L11(K)) EQ (TP10(J)),
   WHERE IL, IF NONE, GO TO 9999
   LET L12(IL) = L12(IL) + QPL0(J)
   LET V3 = QP10(J)
   LET L13(IL) = L13(IL) + V3 * SHIFT
   LET L14(IL) = L14(IL) + V3 * SHIFT
319 REPEAT 312
   GO TO 9999
C
37 DO TO 370, FOR EACH N IN POUE
FIND FIRST, FOR EACH K IN LQ10, WITH (L11(K)) EQ (TP10(N)),
XWHERF IL, IF NONE, GO TO 9999
LET L12(IL) = L12(IL) + QP10(N)
LET V3 = QP10(N)
LET L13(IL) = L13(IL) + V3 * SHIFT
LET L14(IL) = L14(IL) + V3 * SHIFT
370 REPEAT 37
C 371 GO TO 374, FOR EACH M IN PQ12
C 372 GO TO 373, FOR EACH I IN PQ12, WITH (TP10(I)) EQ (TP12(M)),
XAND (SP10(I)) EQ (SP12(M))
C LET OP10(I) = WP10(I) + WP12(M)
C 373 REPEAT 372
C REMOVE M FROM PQ12
DESTRUCT PQ12 CALLED M
C 374 REPEAT 371
C GO TO 9999
C 9999 RETURN
END

*IRFTC R20
SUBROUTINE R20
C
C
C.......PURPOSE - TO INITIALIZE EQUIPMENT OUES.
C
C
C.......IDSUR = 20.
C
C FIND FIRST, FOR EACH M IN LQ20, WITH (L21(M)) EQ (VA),
XWHERF IL, IF NONE, GO TO 10
C
GO TO 20
C 10 CREATE L20
STORE VA IN L21(L20)
LET L22(L20) = L22(L20) + VC
LET L23(L20) = L23(L20) + VC
FILE L20 IN LQ20
GO TO 9999
C 20 LET L22(IL) = L22(IL) + VC
LET L23(IL) = L23(IL) + VC
GO TO 9999
C 9999 RETURN
*IRFTC R30

SUBROUTINE R30

C

C......PURPOSE - TO INITIALIZE SPARE PARTS QUES.

C

C......INSUB = 30.

C

C FIND FIRST, FOR EACH M IN LG30, WITH (L31(M)) EQ (VA),
WHERE IL, IF NONE, GO TO 10

C

GO TO 20

C

10 CREATE L30
STORF VA in L31(L30)
LET L32(L30) = L32(L30) + VC
LET L33(L30) = L33(L30) + VC
FILE L30 IN LG30
GO TO 9999

C

20 LET L32(IL) = L32(IL) + VC
LET L33(IL) = L33(IL) + VC
GO TO 9999

C

9999 RETURN
END
*IPFTE R200

SUBROUTINE R200

C
C
C......PURPOSE - MAINTENANCE COMPLETE.
C
C
C......10SUM = 200.
C
C
IF (MUR1) EQ 0, GO TO 999
CALL R010
C
100 TO 4, FOR EACH M IN TOUER, WITH (TT16(M)) EQ (TRSM)
C
LET T200(M) = ETIME
C
200 TO 3, FOR EACH N IN KODE(M), WITH (R1D(N)) EQ (TRSM),
XAND (RSUM(N)) EQ (1002)
C
FIND FIRST, FOR EACH L IN L410, WITH (L11(L)) EQ (RTYP(N)),
XWHERE IL, IF NONZ, GO TO 3
C
LET F01Y = RTYP(N)
LET L15(LL) = L15(LL) + (T200(M) - 1600(M)) * F01Y
C
3 REPEAT ?
C
4 REPEAT 1
C
9999 RETURN
END
*RAFTC R500

SUBROUTINE R500

C

C......PURPOSE - TEAM DISPATCHED BY BASE CONTROL.

C

C......IDSUR = 500.

C

IF (MORF) EQ (C), GO TO 499
CALL KOLR
IF (DLV4) EQ (PSITE), GO TO 300

C

C............SEARCH FOR NUR (NORS/MARK).

C

100 DD TO 109, FOR EACH M IN 1001, WITH (NEM1(M)) EQ (VC)
C
IF (NUR(M)) EQ (1019), GO TO 101
IF (NUR(M)) EQ (1020), GO TO 102
GO TO 198

C

101 FIND FIRST, FOR EACH L IN L20, WITH (L31(L)) EQ (TYPM(M)),
WHERE IL, IF NONE, GO TO 198
C
LET L36(IL) = L36(IL) + DLV4 + NUR(M)
C
GO TO 198

C

102 FIND FIRST, FOR EACH L IN L20, WITH (L21(L)) EQ (TYPM(M)),
WHERE IL, IF NONE, GO TO 198
C
LET L26(IL) = L26(IL) + DLV9 - IDPR(M)
C
GO TO 198

C

198 REMOVE M FROM NODE
DESTROY NUR CALLED M
C

C

199 REPEAT 100
C

C........CREATE TEAMS ALONG WITH ITS RESOURCES.

C

FIND FIRST, FOR EACH M IN T, WITH (TID(M)) EQ (TRSM),
WHERE IT, IF NONE, GO TO 200
C
LET TIDT(IIT) = TRSM
LET TKIDT(IIT) = VC
LET T501T(IIT) = DLV9
C
GO TO 201

C

200 CREATE TEAM CALLED T
C
LET TTID(T) = TRSM
LET TRID(T) = VC
LET T500(T) = DTLV9
LET S500(T) = DTLV9
C
FILE T IN IQUE
C
GO TO 201
C
201 DO TO 202, FOR EACH M IN IQUE, WITH (RTID(M)) EQ (TRSM)
C
FILE M IN RUUF(T)
REMOVE M FROM RUUF
C
202 REPEAT 201
C
GO TO 9999
C
C .............STORE DEPARTURE TIME FROM SITF(T504).
C
300 DO TO 301, FOR EACH M IN IQUE, WITH (TTID(M)) EQ (TRSM)
C
LET T504(M) = ETIME
C
301 REPEAT 300
C
GO TO 9999
C
9999 RETURN
END
*IDF DC R600  
SUBROUTINE R600

C
C....PURPOSE - TEAM ARRIVAL AT SITE.
C
C....RUN = 600.
C
C
1 DO TO 4, FOR EACH M IN TQUF, WITH (TTID(M)) EQ (TRSM)
C
   LET T600(M) = ETIMF
C
2 DO TO 3, FOR EACH N IN KQUF(M), WITH (RTID(N)) PO (TRSM),
   XAND (RSUM(N)) EQ 1002
C
   FIND FIRST, FOR EACH L IN QLQU, WITH (LLLL(L)) PY (R1P1N1),
   XWHERE LM, IF NON1, GO TO 3
C
   LET F41Y = R41Y(N)
   LET L17(L1) = L17(L1) + ((T600(M) - T500(M)) * F41Y)
C
3 REPEAT 2
C
4 REPEAT 1
C
9999 RETURN
END
*IPFGC R700

SUBROUTINE R700

C......PURPOSE - TEAM ARRIVAL AT BASE.
C
C......TDSUP = 700.
C
1 DO TO 4, FOR EACH M IN TQUE, WITH (TTID(M)) EQ (TRSM)
C
   LET T700(M) = FTIME
C
2 DO TO 3, FOR EACH N IN ROUL(M), WITH (RTID(N)) EQ (TRSM),
   XAND (RSUB(N)) EQ (LOU2)
C
   FIND FIRST, FOR EACH L IN LO10, WITH (LL1(L)) EQ (HTYP(N)),
   XWHERE: IL, IF NUNE, GO TO 3
C
   LET FQTY = QTYP(N)
   LET IL1(L) = IL1(L) + (T700(M) - T504(M)) * FQTY
C
3 REPEAT ?
L
4 REPTA L
L
1999 RETURN
END
*RFTL RAOU
SUBROUTINE RAOU
C
C
C*****PURPOSE - TEAM LOST IN ROUTE TO SITE/LARGE
C
C
C*****IUSUP = 800
C
C
1 10 TO 9, FOR EACH IT IN TOUE, WITH (TTIM(IT)) EO (TSM)
C
LET TMOO: IT = 1 TIME
C
2 10 TO 8, FOR EACH IR IN PUQF(IT), WITH (PTLF(IR)) EO (TSM),
XAND (RSUB(IR)) EO (1007)
C
FIND FIRST, FOR EACH ML IN LQ16, WITH (LML(IR)) EO (RTYP(IR)),
XWHERE IT, IF NUNF, GO TO 7
C
IF (T500(IT)) EO (0.0), GO TO 5
IF (T504(IT)) EO (0.0), GO TO 3
GO TO 4
C
3 LET FQFY = RTYF(IR)
LET LIT(IIR) = LIT(IIR) + ((T800(IT) - T900(IT)) * FQFY)
GO TO 5
C
4 LET FQFY = RTFY(IR)
LET LIT(IIR) = LIT(IIR) + ((T900(IT) - T504(IT)) * FQFY)
GO TO 5
C
5 10 TO 6, FOR EACH IP IN PUQ, WITH (TP01(IP)) EO (RTYP(IR)),
XAND (SP10(IP)) EO (RSFT(IR))
C
LET VIPO(IP) = VIPO(IP) - RTYF(IR)
C
6 REPEAT 5
C
CALL OTIME(IT, IR, IL)
C
1 REMOVE IR FROM PUQF(IT)
DESTROY RESRC CALLED IR
C
8 REPEAT 7
C
IF PUQF(IT) IS NOT EMPTY, GO TO 4
C
REMOVE IT FROM TOUE
DESTROY TEAM CALLED IT
C
9 REPEAT 1
C
9994 RETURN
END
*INFTC R1002
   SUBROUTINE R1002
   
   C.....PURPOSE - PERSONNEL RESOURCES ASSIGNED TO TEAM.
   
   C.....IDSUR = 1002.
   
   CREATE RESRC CALLED R
   LET R1ID(R) = VA
   LET R2ID(R) = TRSM
   LET R1YP(R) = VH
   LET R2YR(R) = VC
   LET R3FT(R) = SXW
   LET R4SU(R) = IDSUR.
   
   FILL 4 IN BQUT.
   RETURN.
   END.

*IBFTC R1010
   SUBROUTINE R1010
   
   C.....PURPOSE - PARTS STOCKOUT (NORS).
   
   C.....IDSUR = 1010.
   
   1 DO TO 2, FOR EACH M IN L30, WITH (L31(M)) EQ (VR)
   LET L39(M) = L38(M) + 1
   LET L39(M) = L39(M) + 1
   2 :REPEAT 1
   
   CREATE NORS CALLED S
   LET N1ID(S) = VA
   LET N2ID(S) = TRSM
   LET N1YP(S) = VH
   LET N2YR(S) = VC
   LET N5UR(S) = IDSUR
   LET N6NRS(S) = ETIME
   
   FILL S IN MQUT.
   1999 RETURN.
   END.
*IHFTC R1G12
SUBROUTINE R1G12
C
C
C.....PURPOSE - PARTS ASSIGNED TO TEAM.
C
C
C.....10SUB = 1012.
C
C
1 DO TO 2, FOR EACH M IN LU30, WITH (L31(M)) EQ (VM)
    LET L33(M) = L33(M) - VC
    LET L34(M) = L34(M) + 1
    LET L35(M) = L35(M) + 1
    2 REPEAT 1
C
9999 RETURN
END

*IHFTC R1020
SUBROUTINE R1020
C
C
C.....PURPOSE - EQUIPMENT SHELLED IN STORE.
C
C
C.....10SUB = 1020.
C
C
1 DO TO 2, FOR EACH M IN LU20, WITH (L21(M)) EQ (VM)
    LET L28(M) = L28(M) + 1
    LET L29(M) = L29(M) + 1
    2 REPEAT 1
C
CREATE NUR CALLED E
C
LET NTRY(E) = VA
LET NROD(E) = TRSM
LET NTRY(E) = VB
LET NTRY(E) = VC
LET MSUR(E) = 10SUB
LET TMDRF = ETIM
C
FILE E IN NQUE
C
9999 RETURN
END
*IPFTC RIO72
SUBROUTINE R1072
C
C
C......PURPOSE - EQUIPMENT ASSIGNED TO TEAM.
C
C
C......IDSUN = 1022.
C
C
1 DO T = 2, FOR EACH M IN LQ2O, WITH (L21(M)) EQ (VR)
   LET L23(M) = L23(M) - VC
   LET L24(M) = L24(M) + 1
   LET L25(M) = L25(M) + 1
   2 REPEAT 1
C
9999 RETURN
END

*IPFTC R1100
SUBROUTINE R1100
C
C
C......PURPOSE - EXTRA PARTS ASSIGNED TO TEAM.
C
C
C......IDSUN = 1100.
C
C
1 DO T = 2, FOR EACH M IN LQ30, WITH (L31(M)) EQ (VP)
   LET L33(M) = L33(M) - VC
   2 REPEAT 1
C
9999 RETURN
END
*INTEC R1200
SUBROUTINE R1200

C
C
C.....PURPOSE - TXNG. PERSONAL ARRIVAL TO POOL.
C
C
C.....I0505 = 1200.
C
C
IF (T10) EQ (1.0), GO TO 1
IF (T10) EQ (7.0), GO TO 7
GO TO 9999
C
1 FIND FIRST, FOR EACH I IN PLU5, WITH (TP10(I)) EQ (VA),
XAND (SP10(I)) EQ (VAL), WHERE IP, IF NINE, GO TO 9999
C
LET CP10(IP) = CP10(IP) + VC
GO TO 2222

C
7 LET ISHFT = MAX10 / 7
LET I0AY = 1
71 IF (VA) GR (ISHFT), GO TO 72
LET J0AY = I0AY
GO TO 73

C
72 LET ISHFT = ISHFT + (MAX10 / 7)
IF (ISHFT) GR (MAXI0), GO TO 73
LET I0AY = I0AY + 1
GO TO 71

C
73 LET I0AY = DPART(TIME)
IF (I0AY) LE (J0AY), GO TO 1
C
CREATE P12 CALLED P
C
STORE VA IN TP12(P)
STORE VA IN SP12(P)
STORE VC IN WP12(P)
STORE JDAY IN DP12(P)
C
FILE P IN P012
C
GO TO 9999
C
9999 RETURN
END
*IBFTC R1210

SUBROUTINE R1210

C

C....PURPOSE - EXNG. PARTS ARRIVAL TO POOL.
C
C
C....IDSUR = 1210.
C
C
1 DO TO 2, FOR EACH M IN L030, WITH (L31(M)) EQ (VB)
LET L32(M) = L32(M) + VC
LET L33(M) = L33(M) + VC
2 REPEAT 1
C
9999 RETURN
END

*IBFTC R1220

SUBROUTINE R1220

C

C....PURPOSE - EXNG. EQUIPMENT ARRIVAL AT POOL.
C
C
C....IDSUB = 1220.
C
C
1 DO TO 2, FOR EACH M IN L020, WITH (L21(M)) EQ (VR)
LET L22(M) = L22(M) + VC
LET L23(M) = L23(M) + VC
2 REPEAT 1
C
9999 RETURN
END

*IBFTC R1400

SUBROUTINE R1400

C

C....PURPOSE - PARTS RETURNED TO POOL.
C
C
C....IDSUR = 1400.
C
C
1 DO TO 2, FOR EACH M IN L030, WITH (L31(M)) EQ (VB)
LET L33(M) = L33(M) + VC
2 REPEAT 1
C
9999 RETURN
END
*IAFTC R1401

SUBROUTINE R1401

C
C
C........PURPOSE - PARTS RETURNED TO POOL (REPAIRED).
C
C
C........IDSUM = 1401.
C
C
1 DO TO 2, FOR EACH M IN LQ30, WITH (L31(M)) EQ (VR)
   LET L33(M) = L33(M) + VC
2 REPEAT 1
C
9999 RETURN
END

*IBFTC R1450

SUBROUTINE R1450

C
C
C........PURPOSE - PERSONNEL RETURNED TO BASE POOL.
C
C
C........IDSUM = 1450.
C
C
1 DO TO 5, FOR EACH IT IN TOUE, WITH (TITD(IT)) EQ (TRSM)
C
   LET T1450(IT) = ETIME
C
2 DO TO 4, FOR EACH IR IN KQUE(IT), WITH (RTID(IR)) EQ (TRSM),
   XAND (RSUB(IR)) EQ (1002)
C
   FIND FIRST, FOR EACH ML IN LQ10, WITH (LL1(ML)) EQ (RTYP(IR)),
   XWHERE IL, IF NONE, GO TO 3
C
   CALL OVTIME(IT, IR, IL)
C
3 REMOVE IR FROM KQUE(IT)
   DESTROY RESRC CALLED IR
C
4 REPEAT 2
C
   IF KQUE(IT) IS NOT EMPTY, GO TO 2
C
   REMOVE IT FROM TOUE
   DESTROY TEAM CALLED IT
C
5 REPEAT 1
C
9999 RETURN
END
*IBMFC R1470
   SUBROUTINE R1470

   C
   C******PURPOSE - EQUIPMENT RETURNED TO POOL.
   C
   C******IOSUR = 1470.
   C
   1 DO TO 2, FOR EACH M IN L:70, WITH (L21(M)) EQ (VB)
   LET L23(M) = L23(M) + VC
   2 REPEAT 1
   C
   9999 RETURN
   END

*IBMFC R1900
   SUBROUTINE R1900

   C
   C******PURPOSE - TO SET-UP MSITF VARIABLE TO DETERMINE WHICH
   DIRECTION TEAM IS TRAVELING (IOSUR-500).
   C
   C******IOSUR = 1900.
   C
   C
   IF (VC) GE (MSITF), LET MSITF = VC + 1
   C
   RETURN
   END
*IHFTC OVTIME

SUBROUTINE OVTIME(IT, IR, IL)

C
C PURPOSE - TO COMPUTE OVERTIME HOURS.
C
C IDSUn = 1450/800.
C
C IF (T1450(IT)) NE (0.0), GO TO 1
IF (T1800(IT)) NF (0.0), GO TO 2
C
GO TO 9999
C
1 LET OVRHRS = T1450(IT)
GO TO 3
C
2 LET OVRHRS = T1800(IT)
GO TO 3
C
3 LET ICAY = DPART(S500(IT))
LET FDAY = ICAY
LET HSFT = S500(IT) - FDAY
C
IF (HSFT) LS (0.33333), GO TO 10
IF (HSFT) LS (0.66666), GO TO 20
IF (HSFT) LE (1.00000), GO TO 30
C
GO TO 9999
C
10 LET HSFT = 0.33333 + FDAY
GO TO 100
C
20 LET HSFT = 0.66666 + FDAY
GO TO 100
C
30 LET HSFT = 1.00000 + FDAY
GO TO 100
C
100 IF (HSFT) GE (OVRHRS), GO TO 9999
C
LET FQTY = RO TY(IR)
LET L18(IL) = L18(IL) + ((OVRHRS - HSFT) * FQTY)
C
9999 RETURN
END
SUBROUTINE RPT10

PURPOSE - TO REPORT MANHOUR ACCOUNTING.

CALL RLBL/R3.

CALL RTEVT
CALL RTFLM
CALL RTINT
CALL HWNDO
CALL RTUP

1 DO TO 2, FOR EACH L IN LO10
LET L15(L) = L15(L) * 24.0
LET L17(L) = L17(L) * 24.0
LET L18(L) = L18(L) * 24.0
LET L15A(L) = L15A(L) + L15(L)
LET L17A(L) = L17A(L) + L17(L)
LET L18A(L) = L18A(L) + L18(L)
LET L19(L) = (L15(L) + L17(L) + L18(L)) / L13(L)
LET L110(L) = (L15A(L) + L17A(L) + L19A(L)) / L14(L)

CALL RPGIO(L)

LET L12(L) = 0
LET L13(L) = 0.0
LET L15(L) = 0.0
LET L17(L) = 0.0
LET L18(L) = 0.0

2 REPEAT 1

LET RT10 = RT10 + T10

9999 RETURN
END
SUBROUTINE RTFLM

C

C...PURPOSE: TO COMPUTE FLT LINE MAINT HRS AS OF REPORTING TIME.

C

C...CALLED BY RPT10.

C

1 DO TO 5, FOR EACH IT IN TQUE, WITH (T600(IT)) NE (0.0),
X AND (T200(IT)) EQ (0.0)

2 DO TO 3, FOR EACH IR IN RQUE(IT), WITH (RSUR(IR)) EQ (1002)

FIND FIRST, FOR EACH L IN LQ10, WITH (L11(L)) EQ (RTYP(IR)),
WHERE IL, IF NONE, GO TO 4

LET FQTY = RTY(IR)

LET L15(IL) = L15(IL) + ((RT10 - T600(IT)) * FQTY)

3 REPEAT 2

4 LET T600(IT) = RT10

5 REPEAT 1

9999 RETURN
END
*IHFTC RTINT
  SUBROUTINE RTINT

C  C.....PURPOSE - TO COMPUTE INTRAN INPUTS AS OF REPORTING TIME.
C  C
C  C.....CALLED BY RPT10.
C  C
C  1 DO TO 5, FOR EACH IT IN TOUL, WITH (T500(IT)) NE (0.0),
C        AND (T600(IT)) EQ (0.0)
C  C
C  2 DO TO 3, FOR EACH IK IN RQIVIT(1T), WITH (RSUR(IN)) EQ (1002)
C        FIND FIRST, FOR EACH L IN L110, WITH (L11(L)) EQ (RTYP(IR)),
C        WHERE IL, IF NONE, GO TO 4
C  C        LET FQTY = RTY(IK)
C        LET L17(IL) = L17(IL) + ((R10 - T500(IT)) * FQTY)
C  C  3 REPEAT 2
C  C  4 LET T500(IT) = R10
C  C  5 REPEAT 1
C  C  11 DO TO 15, FOR EACH IT IN TOUL, WITH (T504(IT)) NE (3.0),
C        AND (T700(IT)) EQ (0.0)
C  C  12 DO TO 13, FOR EACH IK IN RQIVIT(IT), WITH (RSUR(IN)) LU (1002)
C        FIND FIRST, FOR EACH L IN L110, WITH (L11(L)) EQ (RTYP(IR)),
C        WHERE IL, IF NONE, GO TO 14
C  C        LET FQTY = RTY(IK)
C        LET L17(IL) = L17(IL) + ((R110 - T504(IT)) * FQTY)
C  C  13 REPEAT 17
C  C  14 LET T504(IT) = R110
C  C  15 REPEAT 11
C  C  9999 RETURN
END
*IRFC *RTIVT
SUBROUTINE RTIVT
C
C
C.....PURPOSE - TO COMPUTE OVERTIME HOURS AS OF REPORTING TIME.
C
C.....CALLED BY HRT10.
C
LET OVHRS = RT10
C
100 DO TO 2000, FOR EACH IT IN TOUR, WITH (SSU011IT) IF (0.0)
C
LET IDAY = UPART(SSU011IT)
LET FDAY = IDAY
LET HSFT = SSU011IT - FDAY
C
IF (HSFT) LS 0.33333, GO TO 10
IF (HSFT) LS 0.66666, GO TO 20
IF (HSFT) LE 1.00000, GO TO 10
C
10 LET HSFT = 0.33333 + FDAY
GO TO 100
C
20 LET HSFT = 0.66666 + FDAY
GO TO 100
C
30 LET HSFT = 1.00000 + FDAY
GO TO 130
C
100 IF (HSFT) GE 0.0000, GO TO 2000
C
200 DO TO 2000, FOR EACH IT IN TOUR, WITH (MSU011IT) TO (196)
C
FIND FIRST, FOR EACH LT IN LTQ, WITH (LT001) TO (196),
WHERE LT, IF NONE, GO TO 2000
C
LET FOTY = MSU011
LET LSTLIL = LT001 + ((OVHRS - HSFT) * FOTY)
C
1000 REPEAT 200
C
2000 REPEAT 1
C
9999 RETURN
END
*IBFTC RPT20
SUBROUTINE RPT20

C
C
C......PURPOSE - TO REPORT MAINT. EQUIPMENT USAGE.
C
C......CALLED BY RLBL/R3.
C
CALL RTNORE
CALL HUG20

1 DO TO 2, FOR EACH L IN LQ20
LET L27(L) = L27(L) + L26(L)
CALL RPG20(L)
LET L24(L) = 0
LET L26(L) = 0.0
LET L28(L) = 0
2 REPEAT

LET RT2U = RT20 + T20
9999 RETURN
END

*IBFTC RTNORE
SUBROUTINE RTNORE

C
C
C......PURPOSE - TO COMPUTE NDRE TIME AS OF REPORTING TIME.
C
C......CALLED BY RPT20.
C
1 DO TO 2, FOR EACH N IN NQUE, WITH (NSUB(N)) EQ (1020)
FIND FIRST, FOR EACH L IN LQ20, WITH (L21(L)) EQ (NTYP(N)),
WHERE IL, IF NONE, GO TO 2
LET L26(IL) = L26(IL) + RT
LET TNOR(N) = RT20
2 REPEAT

9999 RETURN
END
*IBFTC RPT30
  SUBROUTINE RPT30
  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  C  ....PURPOSE - TO REPORT SPARE PARTS USAGE.  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  C  ....CALLED BY RLBL/R3.  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  CALL RTNORS
  CALL HDG30
  
  1 DO 2, FOR EACH L IN LQ30
  LET L37(L) = L37(L) + L36(L)
  CALL RPG30(L)
  LET L34(L) = 0
  LET L35(L) = 0.0
  LET L38(L) = 0
  
  2 REPEAT
  LET RT30 = RT30 + T30
  
  9999 RETURN
  END

*IBFTC RTNORS
  SUBROUTINE RTNORS
  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  C  ....PURPOSE - TO COMPUTE NORS TIME AS OF REPORTING TIME.  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  C  ....CALLED BY RPT30.  
  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  
  1 DO 2, FOR EACH N IN NGUE, WITH 'NSUB(N)' EQ (1010)
  FIND FIRST; FOR EACH L IN LQ30, WITH 'L(L)' EQ (NTYP(N)),
  WHERE IL, IF NOE, GO TO 2
  LET L36(IL) = L36(IL) + RT30 - TNOR(N)
  LET TNOR(N) = RT30
  
  2 REPEAT
  
  9999 RETURN
  END
*IBFTC RPPTO
  SUBROUTINE RPPTO
  CALL HDGTO
  CALL HDGTO

  1 DO TO 7, FOR EACH I IN TQUS
    CALL RPPTO(I)
  2 REPEAT 1
  RETURN
  END

*IBFTC HDG10
  REPORT HDG10
  MANHOUR ACCOUNTING FOR
  MAN SHFT TOTAL MANHRS TOTAL MANHRS FLT LINE
  TYPE AVAIL THIS PERIOD TO DATE MAINT HRS
  END:

  PERIOD ENDING ****.**
  INTRAY OVERTIME UTIL FACTOR UTIL FACTOR
  HOURS HOURS THIS PERIOD TO DATE
  END

*IBFTC RPGL0
  REPORT RPGL0(L)
  L11(L),L12(L),L13(L),L14(L),L15(L),L17(L),L18(L),L19(L),L10(L)
  END
  END

-128-
### MAINT. EQUIPMENT USAGE FOR

<table>
<thead>
<tr>
<th>TYPE</th>
<th>QTY AVAILABLE</th>
<th>SITE DEMANDS THIS PERIOD</th>
<th>SITE DEMANDS TO DATE</th>
</tr>
</thead>
</table>

**PERIOD ENDING ****,**

<table>
<thead>
<tr>
<th>NORS</th>
<th>NORS TIME</th>
<th>NORS TIME</th>
<th>NO. NORS</th>
<th>NO. NORS</th>
<th>NORS</th>
<th>NORS</th>
<th>NORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SPARE PARTS USAGE FOR

### END
*IBFTC RPG30
REPORT RPG30(L)
*  *  *  *  *  *
* L31(L),L32(L),L33(L),L34(L),L35(L),L36(L),L37(L),L38(L),L39(L)
END
*  **  **  *  *  *  *
END

*IBFTC HDGTQ
REPORT HDGTQ
* MEMBERS OF TQUES
* TID RID T500 T600 T200 T504 T700
END
*  *
T800 T1450 2
END

*IBFTC RPGTQ
REPORT RPGTQ(I)
*  *  * ****** ****** ****** ****** ****** ****** ******
* TTID(I),TRID(I),T500(I),T600(I),T200(I),T504(I),T700(I),T800(I),T1
END
* ****** ****** ******
450(I)
END
*ENTRY  MAIN

1  22  2
23  R  7
24  R
25  R
26  28  Z
29  R
30  32  I
33  R
34  1  R  100  33

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

INITIALIZATION DECK

7.00000
7.00000
7.00000
8.00000
100

(12,
Programs 4 to 7

AIRCRAFT RECOVERY PACKAGE
The "recovery package" deals with the problem of maintenance quality. The displays that follow refer to the measures associated with the recovery of an aircraft from the effects of the previous sortie.

The best way to determine maintenance quality is to determine its impact on the stated mission of the weapon system. Good maintenance results in high mission capability, poor maintenance degrades mission capability. Most measures of maintenance quality contain this idea by implication, generally being stated as an index of capability to deliver clean aircraft quickly.

The sortie is also a major factor in aircraft maintenance. The sortie, more than any other factor, determines maintenance actions. Consequently, maintenance quality is more intimately associated with the aircraft's recovery.

Analyzing sortie data is a two-step process. The first step is to process the label record tape data by use of a "recovery program." This results in the generation of an "Aircraft Output Tape," (format described in Fig. 14), which is used to generate the selected Program(s) 4-6. This tape must be sorted on tail number and "time job started."

**INITIALIZATION**

The variables description and initialization table, which follows, contains the information required to initialize the program. An example initialization data deck listing follows the "OUTPUT PROGRAM" section. The program requires initialization of 37 variables; only 5 require values (see Table 4). Array 23 specifies the quantity of bases to be analyzed. Array 24 lists the base numbers. Array 26 specifies the total quantity of failure levels to be included in the report. Array 27 specifies the failure level numbers. Array 29 specifies the time that the report is to terminate. All of the other Arrays are set to zero.

**OUTPUT PROGRAM DESCRIPTION**

The input to this program is the tape generated by the ABC Simulator.
Table 4

VARIABLE DESCRIPTION AND INITIALIZATION:
AIRCRAFT RECOVERY SORT

<table>
<thead>
<tr>
<th>Group Number</th>
<th>Number of Elements</th>
<th>Mode</th>
<th>Variance</th>
<th>Initial Value</th>
<th>Array Number of Elements to Be Included in Fig. 3 Col.</th>
<th>Description of Values</th>
<th>Permanent System-Variable Names</th>
<th>Entry Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>Z00Z20</td>
<td>Number of moves to be analyzed</td>
<td>00000</td>
<td>E</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>Z00Z20</td>
<td>Specify each move code to be analyzed</td>
<td>00000</td>
<td>E</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>Z00Z20</td>
<td>Specify each failure level to be analyzed</td>
<td>00000</td>
<td>E</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>Z00Z20</td>
<td>Pluses and negatives other than 010-111</td>
<td>00000</td>
<td>E</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>Z00Z20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### MAINTENANCE DATA (CARD-0)

<table>
<thead>
<tr>
<th>TIME JOB</th>
<th>TIME JOB</th>
<th>TS</th>
<th>TAIL</th>
<th>UNIT</th>
<th>MAN</th>
<th>UNIT</th>
<th>JUL</th>
<th>ELP</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED</td>
<td>ENDED</td>
<td>EI</td>
<td>NO</td>
<td>FAIL</td>
<td>NRS</td>
<td>FAIL</td>
<td>DAY</td>
<td>THE</td>
<td>SUB</td>
</tr>
<tr>
<td>DAY</td>
<td>HR</td>
<td>ME</td>
<td>HR</td>
<td>HR</td>
<td>S</td>
<td>S</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

### SORTIE FLOW (CARD-2)

<table>
<thead>
<tr>
<th>TAKE OFF</th>
<th>LANDING</th>
<th>TAIL</th>
<th>B</th>
<th>JUL</th>
<th>ELP</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>NO</td>
<td>SITE</td>
<td>DAY</td>
<td>THE</td>
<td>STD</td>
</tr>
<tr>
<td>HR</td>
<td>ME</td>
<td>HR</td>
<td>HR</td>
<td>D</td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

### AIRCRAFT STATUS (CARD-1)

<table>
<thead>
<tr>
<th>STATUS</th>
<th>S</th>
<th>TAIL</th>
<th>JUL</th>
<th>ELP</th>
<th>ID</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTED</td>
<td>T</td>
<td>NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HR</td>
<td>ME</td>
<td>HR</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Status Codes:
- 0 Flying
- 1 Off-Alert
- 2 On-Alert

Fig. 14 -- Format for aircraft output tape
The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record. (See pages 108 and 109 of RM-4659-PR).

When a label record is read, the value of EBAS is compared with the table called BASES. If they are equal, the record is processed. Therefore, any combination of 1 or more bases may be run at one time.

When a label record is read, the value of the failure level is compared to the table called FLVLS. If they are equal, the record is processed. Therefore, any combination of 1 or more failure levels may be run at one time.

If ETIME is greater than TMEND, the program is then terminated.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set and/or last-of-set and predecessor and/or successor of set.

Label records (see page 108 of RM-4659-PR).

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDSOR</td>
<td>Idr</td>
</tr>
<tr>
<td>IDSUB</td>
<td>Idd</td>
</tr>
<tr>
<td>SHFT</td>
<td>Shift</td>
</tr>
<tr>
<td>DAYW</td>
<td>Dy/wk</td>
</tr>
<tr>
<td>SXDW</td>
<td>S/wk</td>
</tr>
<tr>
<td>EBAS</td>
<td>Base no.</td>
</tr>
<tr>
<td>VA</td>
<td>Variable-1</td>
</tr>
<tr>
<td>VB</td>
<td>Variable-2</td>
</tr>
<tr>
<td>VC</td>
<td>Variable-3</td>
</tr>
<tr>
<td>TRSM</td>
<td>ID Addresses</td>
</tr>
<tr>
<td>MORE</td>
<td>Dri</td>
</tr>
<tr>
<td>ETIME</td>
<td>Event time</td>
</tr>
</tbody>
</table>

Detail records (see page 109 of RM-4659-PR).

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLV1</td>
<td>Integer variable 1</td>
</tr>
<tr>
<td>DTLV2</td>
<td>Integer variable 2</td>
</tr>
<tr>
<td>DTLV3</td>
<td>Integer variable 3</td>
</tr>
<tr>
<td>DTLV4</td>
<td>Integer variable 4</td>
</tr>
<tr>
<td>DTLV5</td>
<td>Integer variable 5</td>
</tr>
<tr>
<td>DTLV6</td>
<td>Integer variable 6</td>
</tr>
<tr>
<td>DTLV7</td>
<td>Integer variable 7</td>
</tr>
<tr>
<td>DTLV8</td>
<td>Integer variable 8</td>
</tr>
<tr>
<td>DTLV9</td>
<td>Float variable 1</td>
</tr>
<tr>
<td>DTLV0</td>
<td>Float variable 2</td>
</tr>
</tbody>
</table>

Base table.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASES</td>
<td>Number of base codes to be processed.</td>
</tr>
</tbody>
</table>
BASE = Base codes to be processed.
BFLAG = Controls flow of events as a result of EKAS vs BASES.

Failure level table.
FLVLS = Number of failure codes to be processed.
FLVL = Failure level codes to be processed.
TMEND = Time initialized to end this run prematurely.

SETS

Name = MQUE used for maintenance events. No subscript. Ranked on ETIME.
Owner = SIMSCRIPT system.
Member = MAINT.
M1 = Start time.
M2 = End time.
M3 = Team size.
M4 = Tail no.
M5 = Unit that failed.
M6 = Team ID.
M7 = Request ID.
M8 = Site ID.
M9 = IDSUB.

Name = SQUE used for site events. No subscript. Ranked on SID.
Owner = SIMSCRIPT system.
Member = SITES.
SID = Site ID.
ANO = A Site No.
BNO = B Site No.
SMOD = Mode of site.
Serno = Tail No.
STATS = Status of site.
STIME = Start time of status.
NFAIL = Number of failures at site.

Name = SKTQ used for sorties. No subscript. Ranked on Sl.
Owner = SIMSCRIPT system.
Member = SORTE.
S1 = Start time.
S2 = End time.
S3 = B Site No.
S4 = Tail No.
S8 = Site ID.

Name = TQUE used for team events. No subscript. Ranked on TTID.
Owner = SIMSCRIPT system.
Member = TEAM.
TTID = Team ID.
TQTY = Team size.
<table>
<thead>
<tr>
<th>T MAIN0</th>
<th>T ML 1</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T M2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>T M3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>T M4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>T M5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>T M6</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>T M7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>T M8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>T M9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T SQUE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>T PMQ 34</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>T SQUE</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>T MLT 1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T SIT 58</th>
<th>T SID 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T MLT 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MQEO  *ML  L
SQUEO  *S10 L
MAIN ROUTINE

...PLANET - AIRCRAFT RECOVERY TAPE INPUT...

...PURPOSE - TO CREATE AIRCRAFT STATUS TAPE
FOR INPUT TO RECOVERY PACKAGE.

...INPUT - TAPE FROM ABC MODEL.

...OUTPUT - A/C STATUS TAPE.

REWIND 4
REWIND 4

1 CALL FLUSH
CALL SELECT

IF (IDSUR) EQ ( 3), GO TO 3
IF (IFLAG) NE ( 0), GO TO 9999
IF (IDSUR) EQ (110), GO TO 110
IF (IDSUR) EQ ( 200), GO TO 200
IF (IDSUR) EQ ( 500), GO TO 500
IF (IDSUP) EQ (600), GO TO 600
IF (IDSUP) EQ (400), GO TO 600
IF (IDSUP) EQ (1002), GO TO 1002
IF (IDSUP) EQ (1450), GO TO 1450
IF (IDSUP) EQ (1900), GO TO 1900
IF (IDSUP) EQ (2000), GO TO 2000
IF (IDSUP) EQ (2100), GO TO 2100
IF (IDSUP) EQ (1150), GO TO 2150
IF (IDSUP) EQ (2300), GO TO 2300
IF (IDSUP) EQ (2400), GO TO 2400
IF (IDSUP) EQ (2500), GO TO 2500
IF (IDSUP) EQ (3100), GO TO 3100
IF (IDSUP) EQ (3200), GO TO 3200

C 600 CALL 9999
C 110 CALL R110
CALL EXIT
C 200 CALL R200
GO TO 9999
C 500 CALL R500
GO TO 9999
C 600 CALL R600
GO TO 9999
C 800 CALL R800
GO TO 9999
C 1002 CALL R1002
GO TO 9999
C 1450 CALL R1450
GO TO 9999
C 1900 CALL R1900
GO TO 9999
C 2000 CALL R2000
GO TO 9999
C 2100 CALL R2100
GO TO 9999
C 2150 CALL R2150
GO TO 9999
2300 CALL R2300
   GO TO 9999
C
2400 CALL R2400
   GO TO 9999
C
2500 CALL R2500
   GO TO 9999
C
3100 CALL R3100
   GO TO 9999
C
3200 CALL R3200
   GO TO 9999
C
9999 IF (MORE) EQ (0), GO TO 1
   CALL R0TL
   GO TO 9999
C
   END
*IBFTC RLPL
   SUBROUTINE RLPL
C
66666..READS S-PHASE TAPE(N) (BIN MODE)....CALL RECORDS.
C
   LET AFLAG = 0
C
   READ (N) 11,17,11,14,15,16,17,11,14,110,111,112
C
   STORE 11 IN DOSH
   STORE 12 IN DOSH
   STORE 13 IN SHFT
   STORE 14 IN DAW
   STORE 15 IN SDW
   STORE 16 IN FAS
   STORE 17 IN VA
   STORE 19 IN VB
   STORE 19 IN VC
   STORE 110 IN TASM
   STORE 111 IN MORE
   STORE 112 IN ETIME
C
   IF (FTIME) GT (TENM), GO TO 1
   GO TO 9999
C
   1 CALL K3
   CALL EXIT
C
9999 RETURN
   END
*IBFTC SELECT
   SUBROUTINE SELECT
C
C........PURPOSE - TO SELECT EVENTS BY RASE.
C
C
UN TO 1, FOR EACH RASE
IF (CRAS) EQ (BASE(1)), GO TO 2
1 LOOP
LET INLAG = 1
GO TO 9999
C
2 LET INLAG = 0
GO TO 9999
C
9999 RETURN
END
*1BFRC R0TL
SUBROUTINE R0TL
C
C........READS 5-PHASE TAPE (9) (BIN MOD)....DETAIL RECORDS.
C
X
READ (9) 11,12,13,14,15,16,17,18,19,110
C
STORE 11 IN DLV1
STORE 12 IN DLV2
STORE 13 IN DLV3
STORE 14 IN DLV4
STORE 15 IN DLV5
STORE 16 IN DLV6
STORE 17 IN DLV7
STORE 18 IN DLV8
STORE 19 IN DLV9
STORE 110 IN DTV0
C
LET MURE = MORE - 1
C
RETURN
END
*1BFRC R3
SUBROUTINE R3
C
C........PURPOSE - TO CLOSE-OUT AND END R-PHASE.
C
C
C........IODSUM = 3.
C
C
REWIND TAPE 9
C
ENDFILE TAPE 9
REWIND TAPE N
CALL MODI;
1 GO TO 2, FOR EACH N IN MWUE
LET DTIME = M1(N)
CALL CLK1(DTIME,11,12,13)
LET IS1 = 11
LET IS2 = 12
LET IS3 = 13
LET DTIME = M2(N)
CALL CLK1(DTIME,11,12,13)
LET IE1 = 11
LET IE2 = 12
LET IE3 = 13
CALL MQRPG(M,IS1,IS2,IS3,IE1,IE2,IE3)
2 KEPFAT 1
9999 RETURN
END
SUBROUTINE R110
******PURPOSE - START MAINTENANCE FOR******
(1) EXOG. FAILURE.
(2) EXOG. PM.
(3) EXOG. OVERHAUL.
******IDSUB = 110.
FIND FIRST, FOR EACH N IN SWUE, WITH (SID(N)) EQ (TRSN).
WHERE IS; IF NOT, GO TO 9999
IF (VC) EQ (2), GO TO 2
IF (VC) EQ (4), GO TO 6
IF (VC) EQ (6), GO TO 6
GO TO 9999
******EXOG. FAILURE.
2 GO TO 20, FOR EACH FLVLS I
IF (VR) EQ (FLVL(I)), GO TO 21
20 LOOP
   GO TO 9999
C
21 IF (NFAIL(I)) NE (0), GO TO 22
   CALL CRD-3(I)
   LET STATS(I) = 1
   LET STIME(I) = ETIME
C
22 LET NFAIL(I) = NFAIL(I) + 1
C
CREATE MAINT CALLED E2
C
STORE SHNO(I) IN M4(E2)
STORE ETIME IN M1(E2)
STORE VA IN M5(E2)
STORE TRSM IN M6(E2)
STORE LISUH IN M7(E2)
C
FILE E2 IN MQUE
C
GO TO 9999
C
C.............EXDG. PM.
C
4 CREATE MAINT CALLED E4
C
STORE SHNO(I) IN M4(E4)
STORE ETIME IN M1(E4)
STORE VA IN M5(E4)
STORE TRSM IN M6(E4)
STORE LISUH IN M7(E4)
C
FILE E4 IN MQUE
C
GO TO 9999
C
C.............EXDG. OVERHAUL.
C
6 CREATE MAINT CALLED E6
C
STORE SHNO(I) IN M4(E6)
STORE ETIME IN M1(E6)
STORE VA IN M5(E6)
STORE TRSM IN M6(E6)
STORE LISUH IN M7(E6)
C
FILE E6 IN MQUE
C
GO TO 9999
C
9999 RETURN
END
*IBFTC R200

SUBROUTINE R200

C
C.....PURPOSE - END MAINTENANCE.
C
C.....IDSUM = 200.
C
C
IF (MORF) EQ (0), GO TO 9999
CALL KOTL
C
FIND FIRST, FOR EACH M IN MOUE, WITH (M6(M)) EQ (TRSM).
X
WHERE IM, IF NONE, GO TO 9999
C
FIND FIRST, FOR EACH N IN MOUE, WITH (SID(N)) EQ (DTLV1).
XWHERE IS, IF NONE, GO TO 9999
C
IF (NFAIL(IS)) NE (1), GO TO 1
C
CALL CRD3(IS)
LET STATS(IS) = 2
LET STIME(IS) = ETIME
C
1 STORE ETIME IN M2(IM)
C
2 DO TO 3, FOR EACH I IN MOUE, WITH (TTID(I)) EQ (TRSM)
LET M3(IM) = M3(IM) + TOTV(I)
3 REPEAT 2
C
CALL CRDO(IM)
C
LET MFAIL(IS) = NFAIL(IS) - 1
C
REMOVE IM FROM MOUE
DELETE MAINT CALLED IN
C
9999 RETURN
END

*IBFTC R500

SUBROUTINE R500

C
C.....PURPOSE - TEAM DISPATCH BY BASE CONTROL.
C
C.....IDSUM = 500.
C
C
C IF (MURE) EQ (0), GO TO 9999
   CALL RDSL
C IF (DLTV4) EQ (MSITE), GO TO 9999
C FIND FIRST, FOR EACH M IN MQUL, WITH (M7(M)) EQ (VC),
   AND (M8(M)) EQ (DLTV1),
   WHERE IM, IF NONE, GO TO 9999
C STORE TRSM IN M(M)
C 9999 RETURN
   END
*IFTC R800
   SUBROUTINE R800
C C PURPOSE - TEAM ARRIVAL AT SITE.
C C.....IDSUM = 600.
C C
C FIND FIRST, FOR EACH M IN MQUL, WITH (M6(M)) EQ (TRSM),
   AND (M8(M)) EQ (VA),
   WHERE IM, IF NONE, GO TO 9999
C STORE TIME IN M(M)
C 9999 RETURN
   END
*IFTC R800
   SUBROUTINE R800
C C C.....PURPOSE - TEAM LOST ENROUTE.
C C C.....IDSUM = 400.
C C C
C 1 DO TI 2, FOR EACH M IN TWUE, WITH (TTID(M)) EQ (TRSM)
   REMOVE FROM TWUE
   DESTROY TEAM CALLED M
   2 REPST I
C 7999 RETURN
   END
*IFTC R9992
SUBROUTINE R1002

C
C......PURPOSE - PERSONNEL ASSIGNED TO TEAMS (CREW SIZE).
C
C......IDSUM = 1002.
C
C
C    FIND FIRST, FOR EACH M IN TUE, WITH (TTEU(M)) EQ (VA), 
C    WHERE IT, IF NONE, GO TO I
C
C    LET TTY(M) = TTY(M) + VC
C    GO TO 9499
C
C    I CREATE TEAM CALLED I
C    STOR: VA IN TTIU(I)
C    STOR: VC IN TTY(I)
C    FILE I IN TOWH
C    GO TO 9499
C
C 9499 RETURN
CND
*IMFC R1450
SUBROUTINE R1450

C
C......PURPOSE - PERSONNEL RETURNED TO BASE POOL.
C
C
C......IDSUM = 1450.
C
C
C    I LO TO J, FOR EACH I IN T.U., WITH (TTI(M)) EQ (TASH)
C    REMOVE I FROM TUE
C    DESTRIP TEAM CALLED I
C    J REPEAT I
C
C 9499 RETURN
CND
*IMFC R1900
SUBROUTINE R1900

C
C......PURPOSE - GENERATE SITES AT KASI.
C
C
C......IDSUM = 1900.
C
C
IF (MOD(I) EQ 0), GO TO 9999
CALL R0TL
C
CREATI SITES CALLED S
C
STORE: TRSM IN SID(S)
STORE: VA IN AND(S)
STORE: VP IN PNO(S)
STORE: VG IN SMOD(S)
STORE: DTLV IN SERNO(S)
C
LET STATS(S) = ?
LET STIME(S) = FTIME
C
IF (VC) GE (MSITE), LET MSITE = VC + 1
C
FILE S IN SWUL
C
LET I999 = 999
LET 199999 = 99999
LET ITAILN = DTLV
LET ICONF = 2
C
WRITE ON TAPE 8, 1999, 199999, ITAILN, 1999, ICONF
FORMAT (IA, 13, 15, S18, I4, S2M, 13, S10, 11)
C
GO TO 9999
C
9999 RETURN
END
*IBFCT R2000
SUBROUTINE R2000
C
C.....PURPOSE - REQUEST FOR PM.
C
C.....IDSUH = 2000.
C
C
FIND FIRST, FOR EACH N IN SQNE, WITH (SID(N)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C
CREATE MAINT CALLED P
C
STORE SERNO(S) IN M4(P)
STORE: TRSM IN M8(P)
STORE: FTIME IN M1(P)
STORE: VP IN M5(P)
STORE: IDSUH IN M9(P)
FILE P IN MOUI

C
9999 RETURN
END

SUBROUTINE R2100

C
C......PURPOSE - START MAINT. ON FI.
C
C......IDSUB = 2100.
C
C
FINI FIRST, FOR EACH M IN SIX, WITH (MS(M)) EQ (VC),
XAND (M8(M)) EQ (VR), WHERE IM, IF NONE, GO TO 9999
C
FINI FIRST, FOR EACH N IN SQU, WITH (SID(N)) EQ (TRS(N)),
XWHERE IS, IF NONE, GO TO 9999
C
IF (NFAIL(IS)) NE (0), G1 TO 1
CALL CR0(IS)
LET STATS(IS) = 1
LET TIME(IS) = TIME
C
LET NFAIL(IS) = NFAIL(IS) + 1
C
9999 RETURN
END

SUBROUTINE R2150

C
C......PURPOSE - START MAINT. FOR FAILURES CAUSED BY PM.
C
C......IDSUB = 2150
C
C
GO TO 1, FOR EACH FLVLS 1
IF (VA) EQ (FLVL), GO TO 2
LOOP
GO TO 9999
C
2 FINI FIRST, FOR EACH N IN SQU, WITH (SID(N)) EQ (TRS(N)),
XWHERE IS, IF NONE, GO TO 9999
C
IF (NFAIL(IS)) NE (0), GO TO 3
CALL CR13(IS)
LET STATS(IS) = 1
LET STIME(IS) = ETIME

3 LET NFAIL(IS) = NFAIL(IS) + 1
CREATE MAINT CALLED M
STORE SIDM0(IS) IN M4(M)
STORE ETIME IN M1(M)
STORE VA IN M5(M)
STORE TRSM IN M8(M)
STORE I0SUB IN M9(M)

4994 RETURN

*INFL R2300
SUBROUTINE R2300

....PURPOSE - START MAINT. FOR FAILURE CAUSED BY CONTINUOUS MONITOR.

*IDSUM = 2300.

1 UD TO 1, FOR EACH FLVLS I
   IF (VA) EQ (FLVL(II)), GO TO 2
   LOOP
   GO TO 9999

2 FIND FIRST, FOR EACH N IN SQUE, WITH (SID(M)) EQ (TRSM),
   WHERE: IS, IF NONE, GO TO 9999
   IF (NFAIL(IS)) NE (0), GO TO 3
   CALL CRU3(IS)
   LET STATS(IS) = 1
   LET STIME(IS) = ETIME

3 LET NFAIL(IS) = NFAIL(IS) + 1
CREATE MAINT CALLED M
STORE SIDM0(IS) IN M4(M)
STORE ETIME IN M1(M)
STORE VA IN M5(M)
STORE TRSM IN M8(M)
STORE I0SUB IN M9(M)
FILE M IN MQUE
C
9994 RETURN
END
*IBITL R2400
SUBROUTINE R2400
C
C......PURPOSE - KSOURCE REQUEST FOR FAILURES.
C
C......IIDSUB = 2400.
C
C
IF (MORE) EQ (0), GO TO 9999
CALL RDL
C
IF (OTLV1) EQ (1), GO TO 1
IF (OTLV1) EQ (2), GO TO 246
IF (OTLV1) EQ (3), GO TO 3
IF (OTLV1) EQ (4), GO TO 246
IF (OTLV1) EQ (5), GO TO 2
IF (OTLV1) EQ (6), GO TO 246
GO TO 9999
C
1 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2300),
XAND (M8(M)) EQ (TRSM), AND (M5(M)) EQ (VR), WHERE IM, IF NONE,
XGO TO 9999
C
STORE VC IN M7(IM)
GO TO 9999
C
246 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (110),
XAND (M8(M)) EQ (TRSM),
XWHERE IM, IF NONE, GO TO 2460
C
STORE VC IN M7(IM)
GO TO 9999
C
2460 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2000),
XAND (M8(M)) EQ (TRSM),
XWHERE IM, IF NONE, GO TO 9999
C
STORE VC IN M7(IM)
GO TO 9999
C
3 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2500),
XAND (M8(M)) EQ (TRSM), AND (M5(M)) EQ (VR), WHERE IM, IF NONE,
XGO TO 9999
C
STORE VC IN M7(IM)
STORE ETIME IN M1(IM)
C
FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (MA(IM)),
WHERE IS, IF NONE, GO TO 9999
C
IF (NFAIL(IS)) 'IE (0), GO TO 30
CALL CF03(IS)
LET STATS(IS) = 1
LET STIME(IS) = ETIME
C
30 LET NFAIL(IS) = NFAIL(IS) + 1
C
GO TO 9999
C
5 FIND FIRST, FOR EACH M IN MQUE, WITH (M9(M)) EQ (2150),
XAND (M2(M)) EQ (TRSM), AND (M5(M)) EQ (VH), WHERE IM, IF NONE,
XGO TO 9999
C
STORE VC IN M7(IM)
GO TO 9999
C
9999 RETURN
END
*IBFTC R2500
SUBROUTINE R2500
C
C
.....PURPOSE - START MAINT. FOR FAILURE FOR UNDETERMINED FAILURE.
C
C
.....IDSUB = 2500.
C
C
UO TO 1, FOR EACH FLVLS 1
IF (VA) EQ (FLVL(1)), GO TO 2
1 LOOP
GO TO 9999
C
2 FIND FIRST, FOR EACH N IN SQUE, WITH (SID(N)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C
CREATE MAINT CALLED M
C
STORF SFRNO(IS) IN M4(M)
STORF ETIME IN M1(M)
STORF VP IN M5(M)
STORF TRSM IN M8(M)
STORF IDSUK IN M9(M)
FILE M IN MQUE
C
9999 RETURN
END

*IBFTC R3100
CALL R3100
C
C......PURPOSE - START FLIGHT.
C
C
C......IUSUB = 3100.
C
C
FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C
CALL CRD3(IS)
C
LET STAT$1(IS) = 0
LET STIME1(IS) = ETIME
C
CREATE SORTED CALLED S
C
STORE SERNO1(IS) IN S4(IS)
STORE ETIME1 IN S1(IS)
STORE BNO1(IS) IN S3(IS)
STORE TRSM IS IN S8(IS)
C
FILE S IN SRTQ
C
9999 RETURN
END

*IBFTC R3200
CALL R3200
C
C......PURPOSE - END FLIGHT.
C
C
C......IUSUB = 3200.
C
C
FIND FIRST, FOR EACH M IN SRTQ, WITH (SID(M)) EQ (TRSM),
WHERE ISRT, IF NONE, GO TO 9999
C
FIND FIRST, FOR EACH M IN SQUE, WITH (SID(M)) EQ (TRSM),
WHERE IS, IF NONE, GO TO 9999
C
CALL CRD3115)
LET STA.S1(IS) = 2
LET STIME(IS) = ETIME
STORE ETIME IN S2(ISHT)
CALL CR02(ISHT)
REMOVE ISRT FROM SKTW
DESTROY SORTED CALLED ISHT

9999 RETURN
END

*8FTC CRDL
SUPROUTINE CRDL(IIM)

C
C PURPOSE - OUTPUT MAINTENANCE DATA.
C
C CALLED BY R200.
C
C
LET DTIME = H1(IM)
CALL CLK1(DTIME,11,12,13)
LET III = 11
LET I52 = 12
LET I53 = 13

LET DTIME = H2(IM)
CALL CLK1(DTIME,11,12,13)
LET IIE1 = 11
LET IIE2 = 12
LET IIE3 = 13

IF (I51) EQ (IIE1), GO TO 100

IF (I51) EQ (IIE1), GO TO 10
LET IIE1 = I51
LET IIE2 = 23
LET IIE3 = 5

CALL CNVKT(IS1,IS2,IS3,IIF1,IIE2,IIE3,CNVTHK)

LET IHKELP = CNVTHK * 10 * 0 + .5
IF (IHKELP) EQ (0), LET IHKELP = 1
LET IHKMAN = IHKELP * M3<IM>

WRITE ON TAPE B, 151,152,153,IIE1,IIE2,IIE3,
M3<IM>,M4<IM>,M5<IM>,IHKMAN,M5<IM>,IS1,IHKELP,
M9<IM>,0
SUBROUTINE CRDZ(ISRT)

C .......PURPOSE - OUTPUT SORTIE DATA.
C
C .......CALLED BY R3200.
C
C
LET DTIME = SIG(SRT)
CALL CLK(DTIME,II,II,II)
LET IS1 = II
LET IS2 = II
LET IS3 = III
C
LET DTIME = SIG(SRT)
CALL CLK(DTIME,II,II,II)
LET IE1 = II
LET IE2 = II
LET IE3 = III
C
IF (IS1) EQ (IE1), GO TO 100
I IF (IS1) EQ (LFA1), GO TO 10
    LET 111 = IS1
    LET 112 = 23
    LET 113 = 59
C CALL CNVR(T(IS1,IS2,IS3,1E1,1E2,1E3,CNVTHK))
C LET IMHELP = CNVTHK * 10.0 + .5
    IF (IMHELP) FC (0), LET IMHELP = 1
C WRITE ON TAPE 8, IS1,IS2,IS3,1E1,1E2,1E3,
    S4(LISRT),S3(LISRT),IS1,IMHELP,SM(1S1T),2
    FORMAT (5A,14,12,1?,14,12,12,510,14,514,14,510,13,13,51,15,51,11)
C LET IS1 = IS1 + 1
    LET IS2 = 0
    LET IS3 = 0
C 10 TO 1
C 10 IF (IS2) NE (1E1), GO TO 100
    IF (IS3) NE (1E3), GO TO 100
    GO TO 9999
C 100 CALL CNVR(T(IS1,IS2,IS3,1E1,1E2,1E3,CNVTHK))
C LET IMHELP = CNVTHK * 10.0 + .5
    IF (IMHELP) FC (0), LET IMHELP = 1
C WRITE ON TAPE 8, IS1,IS2,IS3,1E1,1E2,1E3,
    S4(LISRT),S3(LISRT),IS1,IMHELP,SM(1S1T),2
    FORMAT (5A,14,12,1?,14,12,12,510,14,514,14,510,13,13,51,15,51,11)
C 9999 RETURN
    END

SUBROUTINE CR13(IS1)
C
C PURPOSE - OUTPUT STATUS DATA.
C
C CALLED BY - R110, R200, R2100, R2150, R2500, R3100, R3200.
C
C LET DTIME = STIME(IS1)
    CALL CLK(DTIME,11,12,13)
    LET IS1 = 11
LET J52 = 12
LET J53 = 13
C
LET GTIME = ETIME
CALL CLK1(DTIME, 11, 12, 13)
LET IE1 = 11
LET IE2 = 12
LET IE3 = 13
C
IF (IS1) NE (IE1), GO TO 1000
IF (IS2) NE (IE2), GO TO 1000
IF (IS3) NE (IE3), GO TO 1000
GO TO 9999
C
1000 IF (IS1) EQ (IE1), GO TO 100
C
10 IF (IS1) EQ (IE1), GO TO 10
LET IE1 = IS1
LET IE2 = 23
LET IE3 = 53
C
CALL CNVRT(IS1, IS2, IS3, IL1, IL2, IL3, CNVTMH)
C
LET INHELP = CNVTMH * 10.0 -.5
IF (INHELP) IC (10), LET INHELP = 1
C
WRITE ON TAPE *, (IS1, IS2, IS3, IL1, IL2, IL3, CNVTMH)
X
FORMAT (SP, 14, 14, 12, 14, 12, 14, 3, 2, 1, 14, 15, 15, 11)
C
LET IS1 = IS1 + 1
LET IS2 = 0
LET IS3 = 0
C
GO TO 1
C
10 IF (IS2) NE (IE2), GO TO 100
IF (IS3) NE (IE3), GO TO 100
GO TO 9999
C
100 CALL CNVRT(IS1, IS2, IS3, IL1, IL2, IL3, CNVTMH)
C
LET INHELP = CNVTMH * 10.0 -.5
IF (INHELP) IC (10), LET INHELP = 1
C
WRITE ON TAPE *, (IS1, IS2, IS3, IL1, IL2, IL3, CNVTMH)
X
FORMAT (SP, 14, 14, 12, 14, 12, 14, 3, 2, 1, 14, 15, 15, 11)
C
9999 RETURN
END
*CNVT CLNVT
SUBROUTINE CONVRUL (IS1, IS2, IS3, IE1, IE2, IE3, CVNTHR)
C
C... PURPOSE - TO CONVERT STOP - START TIME DAY HOUR MINUTES
C TO DECIMAL HOURS.
C
LET FS1 = IS1
LET FS2 = IS2
LET FS3 = IS3
LET FE1 = IE1
LET FE2 = IE2
LET FE3 = IE3
C
LET CVNTHR = (((FE1 * 24.0 * 60.0) + (FE2 * 60.0) + FE3)
X - (((FS1 * 24.0 * 60.0) + (FS2 * 60.0) + FS3))
/ 60.0
C
RETURN
END
*INFTC CLK1
SUBROUTINE CLK1(DTIME, 11, 12, 13)
C
C... PURPOSE - TO CONVERT DECIMAL DAYS TO DAYS, HOURS, MINUTES.
C
LET IDY = DPAR(DTIME)
LET IHR = MPAR(DTIME)
LET IMM = MPAR(DTIME)
IF (IMM NE (6U), GO TO 10
LET IHR = IHR + 1
LET IMM = 0
10 IF (IHR) LE (24), GO TO 20
LET IHR = IHR - 24
LET IDY = IDY + 1
20 LET II = IDY
LET 12 = IHR
LET 13 = IMM
C
RETURN
END
*INFTC RMOD
REPORT RMOD
* MAINT START
END CREW A/C SYS TID RIO S10 S10S10SUB
END
*INFTC RORPG
REPORT RORPG(M, IS1, IS2, IS3, IE1, IE2, IE3)
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *}
*   IS1, IS2, IS3, IE1, IE2, IE3, M3(M), M4(M), M5(M), M6(M), M7(M), M8(M), M9(M)
END
-161-

END
ENTRY
1 37
  1 22 02
  23 0 R
  24 1 R 100 23
-163-

97
98
99

25 c z
26 u k
27 1 k 10 z 6

0
1
2
3
4
5
6
7
8
9

28 0 z
29 0 r
30 37 0 z

19
(11)

997.9994
Program 4

AIRCRAFT RECOVERY TIME DISTRIBUTION
IV. AIRCRAFT RECOVERY TIME DISTRIBUTION

The information of most general interest is related to the recovery of the entire aircraft (as opposed to system and subsystem recovery). Figure 15 depicts one of the recovery curves that has been developed.

Note that the two halves of the histogram are identical: each cell entry represents a sortie. The left half shows the touchdown time; the corresponding right half shows the type of sortie, but is left blank because PLANET does not distinguish between sortie types. Provisions are available if the user should decide to add this feature later. Below this, the "HOURS" and "P =" lines show the percentage of aircraft recovered at the hour indicated (in this case, 93 percent are recovered 2 hours after touchdown).

The next line contains a number of statistical computations for facilitating analysis. Of these, two are particularly interesting. "AVG = 1.18" shows that average recovery time (for unscheduled maintenance) was just over 1 hour. "TOTAL = 84.72" shows that approximately 85 hours of Operationally Ready (OR) time were lost in recovering aircraft from the effects of the sorties.

A fighter group is somewhat more fortunate than a bomber wing in that many fighter touchdowns require no unscheduled maintenance. This is reflected in the next line, showing that although 227 sorties were flown, 155 required no unscheduled maintenance, for a breakrate of .317. The following line shows the conventional data, except the "SATURDAY INDEX," which is obtained by dividing total man-hours by total elapsed time. This yields the average number of men on the aircraft for unscheduled maintenance during recovery.

The final line is a precaution that all data may not be graphed.

PROGRAM DESCRIPTION

This program is written in standard FORTRAN IV language.

Input is from card and tape. The highest tail number to be read from tape is read from a card as a five-digit integer. The program
reads the AIRCRAFT RECOVERY TAPE and accumulates the Recovery Time Distributions to be printed.

An Array H contains the histogram of time versus frequency, and an array NO contains a count of frequency at each time. When the last tail number desired has been processed, the histogram is printed.
### ENTIRE AIR-CRAFT RECOVERY TIME DISTRIBUTIONS

| TOUCH-DOWN TIMES | 40  | 39  | 38  | 37  | 36  | 35  | 34  | 33  | 32  | 31  | 30  | 29  | 28  | 27  | 26  | 25  | 24  | 23  | 22  | 21  | 20  | 19  | 18  | 17  | 16  | 15  | 14  | 13  | 12  | 11  | 10  | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| HOURS            | 2   | 4   | 6   | 8   | 10  | 12  | 14  | 16  | 18  | 20  | 22  | 24  |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| P                | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

**TOTAL TIME** = 64.72  **AVERAGE** = 1.18  **MIN** = 0.40  **MAX** = 3.27  **STD. DEV.** = 0.26

**TOTAL SORTIES** = 227  **155 SORTIES REQUIRED NO FIX TIME**  **BREAK RATE** = 0.317

**TOTAL MIN HOURS** = 250.2  **SATURATION INDEX** = 3.0  **AVE T/A (ALL SORTIES)** = 0.4

0 ITEM(S) GREATER THAN 24 HOURS

---

**Fig. 15 -- Aircraft recovery distribution -- an example**
DIMENSION H(40,50), NO(50)
DATA BCST/IH*9/ 
DATA BLANK/IH / 
REIND 9 
READ (5,9000) MAXT 
DO 100 J=1,50 
NO(J)=0 
DO 200 I=1,40 
100 H(I,J)=BLANK 
NSORT=0 
NOFIX=0 
MHR=0 
NG1=0 
NSUM=0 
NSD=0 
MIN=-999999 
MAX=-999999 
READ (8,9100) IDAY, INN, JDAY, JHR, JMN, ISC, ITN, MHR, JD, IET, ID, NC 
IF (JD.NE.9999) GO TO 250 
200 IF (ITN.GE.MAXT) GO TO 500 
GO TO 150 
250 IF (NC.NE.2) GO TO 150 
NSORT=NSORT+1 
MAINT=0 
MAIN=0 
IST=999999 
READ (8,9100) IDAY, INN, JDAY, JHR, JMN, ISC, ITN, MHR, JD, IET, ID, NC 
IF (JD.EQ.9999) GO TO 375 
IF (NC.NE.0) GO TO 350 
MAIN=1 
IST=IINN+60*(JHR+24*IDAY) 
JNO1=JMN+60*(JHR+24*JDAY) 
IF (ID.EQ.2150) GO TO 325 
IF (ID.EQ.2500) GO TO 325 
IF (ID.EQ.2900) GO TO 325 
IST=JNO1 
GO TO 300 
325 IF (MAINT.EQ.0) JND=JNO1 
MAINT=1 
IF (IST1.LE.IST) IST=IST1 
IF (JND1.GE.JNO1) JND=JND1 
MHR=MHR+MHR 
GO TO 300 
350 IF (ISC.NE.2) GO TO 300 
IF (MAIN.EQ.0) GO TO 300 
375 IF (MAINT.NE.0) GO TO 400 
NOFIX=NOFIX+1 
GO TO 475 
400 IREC=JND-IST 
IF (MIN.GT.IREC) MIN=IREC 
IF (MAX.LT.IREC) MAX=IREC
NSUM=NSUM+IREC
NSD=NSD+IREC*IREC
IF (IREC.LE.14400) GO TO 450
NGRT=NGRT+1
GO TO 475
450 J=Z+IREC/30
I=MO(J)+1
IF (I.GT.40) I=40
MO(J)=1
H(J)=BCAST
GO TO 475
475 IF (ZD.EQ.999) GO TO 200
GO TO 190
500 CONTINUE
N=NSORT-NOFIX
XN=N
XSORT=NSORT
XBR=XN/XSORT
XSUM=NSUM
XSUM=XSUM/600.
XAVG=XSUM/XN
XSD=NSD
XSD/XSD/(3600.*XN)*XAVG*XAVG
XMIN=MIN
XMAX=MAX
XMIN=XMIN/600.
XMAX=XMAX/600.
XMHR=XMHR
XMHR=XMHR/10.
XS=XMHR/XSUM
XAT=XSUM/XSORT
NOS=NO(J)
GO 595 J=1,49
NO(J)=100*NO(J)/N
NOS=NOS+NO(J+1)
550 CONTINUE
WRITE (6,9500)
WRITE (6,9510)
GO 600 I=1,40
I=I+1
WRITE (6,9520) I,1*(H(I,J),J=1,50),11
600 CONTINUE
WRITE (6,9530) (1,1=2,24,2),1,1=2,24,2)
WRITE (6,9540) (NO(J),J=5,50,4)
WRITE (6,9550) NSUM,XAVG,XMIN,XMAX,XSD
WRITE (6,9560) NSORT,NOFIX,XBR
WRITE (6,9570) XMHR,XSI,XAT
WRITE (6,9580) NGRT
RELOAD 0
STOP
9000 FORMAT (15)
9100 FORMAT (6X,14,218,14,312,6X,14,15X,14,5X,213,1X,15,1X,11)
9900 FORMAT (14I,35X,44MENTIRE AIR-CRAFT RECOVERY TIME DISTRIBUTIONS)
9510 FORMAT (1HO,20X,14HTOUCH-DOWN TIMES,50X,11HSORTIE TYPE)
9520 FORMAT (3X,12,4X,50A1,6X,12)
9530 FORMAT (1HO,5MHOURS,3X,1H,1214,5X,5MHOURS,3X,1H,1214)
9540 FORMAT (1X,3HP = ,6X,1214)
9550 FORMAT (1HO,3HN = ,14,3X,12HTOTAL TIME = ,F10.2,3X,9H AVERAGE = ,F8.2,
  3X,9HMIN. = ,F8.2,3X,9HMAX. = ,F8.2,3X,11HSTD. DEV. = ,F8.2)
9560 FORMAT (1X,13HTOTAL SORTIES,15,3X,15,1X,
  20H SORTIES REQUIRED NO FIX TIME,3X,12HBREAK RATE = ,F5.3)
9570 FORMAT (1X,17HTOTAL MAN HOURS = ,F8.1,
  3X,16HSATURATION INDEX = ,F5.1,
  3X,23HAVE T/A (ALL SORTIES) = ,F5.1)
9580 FORMAT (1HO,14,30H ITEM(S) GREATER THAN 24 HOURS)

ENTRY
20
SISYS ENDJOB TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
Program 5

SYSTEM RECOVERY
V. SYSTEM RECOVERY

The retrieval of the aircraft system (two-digit) and subsystem (three-digit) data serves several purposes: it enables the monitoring of break and recovery rates; it provides a set of job standards for unscheduled maintenance; and it helps identify aircraft having systems seriously aberrant from the fleet norm.

The two- and three-digit recovery program generally makes a more satisfactory job standard for unscheduled maintenance than does the conventional five-digit method of determining standards. The reason is that to maintenance and control personnel, the "job" consists of the entire action of clearing a complaint. The two- and three-digit recovery program produces the summary of the actions to clear the complaints.

All entries in Fig. 16 show the aircraft serial numbers. The ordinate is a column entry count. The P - line shows the percentage of aircraft recovery by the corresponding time in the hours row. The average recovery follows MEAN = and this is followed by conventional statistical information. Total time shows total OR time lost to this system. SATURATION INDEX is the average number of men working on system recovery.

PROGRAM DESCRIPTION

This program is written in standard FORTRAN IV language. Note the necessary changes indicated in the program listing to allow the program to run on IBM System 360 computers.

Input is from card and tape. The highest tail number to be read from tape and the highest unit number to be printed are read from a card as five-digit integers. The program reads the AIRCRAFT RECOVERY TAPE as many times as necessary to present a chart for each unit number. The information for as many as twenty units is accumulated each time the tape is read through.

An array ITN contains the list of tail numbers versus time of recovery for as many as twenty units, and an array NO contains a count of number of aircraft at each time for each type of failure. When the last tail number desired has been processed, the charts for each type of failure that occurred in the current set of twenty units are printed and the procedure is resumed for the next set of twenty units.
### System Recovery

**Unit Number 1**

<table>
<thead>
<tr>
<th>Hours</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table:**

- **Hours:** 1 to 57
- **Total Time:** 19.30
- **Average:** 1.23
- **Min:** 0.65
- **Max:** 1.88
- **STD. Dev.:** 0.09
- **Total Sorties:** 327
- **Sorties Required:** 119
- **Fix Time Break Rate:** 0.31%
- **Total Run Hours:** 57.6
- **Saturation Index:** 3.5
- **Ave Y/A (All Maintenance):** 0.1

**Fig. 16 -- System recovery**
**IBM 704 SR**

DIMENSION ITB(20), NA(20), MHRT(20), MIN(20), MAX(20), NSUM(20), NSD(20)
DIMENSION ITN(50, 20), h0(20, 20)
DIMENSION NT1(10), NT2(10), NT3(10), NT4(10)
DATA MBLANK/1H /
DATA IZ1/4H 0 /
DATA IZ2/4H 00 /
DATA IZ3/4H 000 /
DATA NT1(1)/4H0000, NT1(2)/4H0001, NT1(3)/4H0002, NT1(4)/4H0003 /
DATA NT1(5)/4H0004, NT1(6)/4H0005, NT1(7)/4H0006, NT1(8)/4H0007 /
DATA NT1(9)/4H0008, NT1(10)/4H0009 /
DATA NT2(1)/4H0000, NT2(2)/4H0010, NT2(3)/4H0020, NT2(4)/4H0030 /
DATA NT2(5)/4H0040, NT2(6)/4H0050, NT2(7)/4H0060, NT2(8)/4H0070 /
DATA NT2(9)/4H0080, NT2(10)/4H0090 /
DATA NT3(1)/4H0000, NT3(2)/4H0100, NT3(3)/4H0200, NT3(4)/4H0300 /
DATA NT3(5)/4H0400, NT3(6)/4H0500, NT3(7)/4H0600, NT3(8)/4H0700 /
DATA NT3(9)/4H0800, NT3(10)/4H0900 /
DATA NT4(1)/4H0000, NT4(2)/4H1000, NT4(3)/4H2000, NT4(4)/4H3000 /
DATA NT4(5)/4H4000, NT4(6)/4H5000, NT4(7)/4H6000, NT4(8)/4H7000 /
DATA NT4(9)/4H8000, NT4(10)/4H9000 /
C DELETE EVERYTHING BETWEEN THE ASTERISKS IF THE PROGRAM IS TO BE RUN
C ON A MACHINE THAT ALLOWS FEWER THAN FIVE HOLLERITH CHARACTERS/WORD.
C THESE CARDS SHOULD BE REMOVED TO RUN ON ANY IBM SYSTEM 360 COMPUTER.

C*******************************************************************************
DATA IZ2/5H0000 /
DO 1 I=1,10
NT1(I)=NT1(I)-IZ2
NT2(I)=NT2(I)-IZ2
NT3(I)=NT3(I)-IZ2
NT4(I)=NT4(I)-IZ2
1 CONTINUE
C*******************************************************************************
NIT=20
READ (5, 9000) MAXT, MAXU
NT1=-NIT
90 REMIND 0
NIT=NIT+MHT
IF (NIT.GT.MAXU) STOP
DC 100 K=1, NIT
ITB(K)=0
MHRT(K)=0
NSUM(K)=0
NSD(K)=0
NA(K)=0
MIN(K)=999999
MAX(K)=999999
DO 100 K=1, 20
MO(J,K)=0
DO 100 1=1, 50
100 ITN(I,J,K)=MBLANK
NSORT=0
NOFIX=0
NGRT=0
150 READ (8,9100) IDAY, IHR, IMN, JDAY, JHR, JMN, ISC, IT4, IT3, IT2, IT1,
* MHR, NUF, JD, IET, ID, NC
  ITA=IT1+10*(IT2+10*(IT3+10*(IT4))
  MAINT=0
  NSOR=0
  IF (JD.NE.999) GO TO 250
200 IF (NC.GE.MAXT) GO TO 500
GO TO 150
250 IF (NC.NE.2) GO TO 150
  NSORT=NSORT+1
  NSOR=1
  MAIN=0
300 READ (8,9100) IDAY, IHR, IMN, JDAY, JHR, JMN, ISC, IT4, IT3, IT2, IT1,
* MHR, NUF, JD, IET, ID, NC
  IF (JD.EQ.999) GO TO 375
  IF (NC.NE.0) GO TO 350
310 CONTINUE
  MAIN=1
  IF (ID.EQ.2150) GO TO 325
  IF (ID.EQ.2500) GO TO 325
  IF (ID.EQ.2300) GO TO 325
  IF (ID.EQ.110) GO TO 325
GO TO 340
325 MAINT=1
  K=NUF-NIT
  IF (K.LE.0) GO TO 340
  IF (K.GT.MIT) GO TO 340
  ITNO=NT1(IT1+1)+NT2(IT2+1)+NT3(IT3+1)+NT4(IT4+1)
  IF (ITA.GE.10) GO TO 330
  ITNO=I21-ITNO
GO TO 335
330 IF (ITA.GE.100) GO TO 332
  ITNO=I22-ITNO
GO TO 335
332 IF (ITA.GE.1000) GO TO 335
  ITNO=I23-ITNO
335 CONTINUE
  NA(K)=NA(K)+1
  MHR(K)=MHR(K)+MHR
  IST1=IMN+60*IHR+24*IDAY
  JND1=JMN+60*JHR+24*JDAY
  IREC=JND1-IST1
  IF (MIN(K).GT.IREC) MIN(K)=IREC
  IF (MAX(K).LT.IREC) MAX(K)=IREC
  NSUM(K)=NSUM(K)+IREC
  MSD(K)=MSD(K)+IREC*IREC
  J=IET/30
336 JJ=1
  IF (J.GT.24) J=20
  I=NO(J,K)+1
IF (I .LE. 50) GO TO 339
IF (J .LT. 20) GO TO 338
339 CONTINUE
    NO(J,K) = 1
    ITM(I,J,K) = ITNO
    ITB(K) = ITB(K) + 1
340 IF (NSOR.EQ.0) GO TO 150
    GO TO 300
350 IF (ISC.NE.2) GO TO 300
    IF (MAIN.EQ.0) GO TO 300
375 IF (MAINT.EQ.0) NOFIX = NOFIX + 1
    IF (JO.EQ.999) GO TO 2CO
    GO TO 150
500 CONTINUE
    N = NSORT - NOFIX
    XN = N
    XSORT = NSORT
    XBR = XN/XSORT
    DO 700 K = 1, NIT
    NIT = NIT + K
    IF (ITB(K).EQ.0) GO TO 700
    N = NA(K)
    XN = N
    XSUM = NSUM(K)
    XSUM = XSUM/60.
    XAVG = XSUM/XN
    XSD = NSD(K)
    XSD = XSD/(3600.*XN) - XAVG*XAVG
    XMIN = MIN(K)
    NMAX = MAX(K)
    XMAX = NMAX
    IF (NMAX.GT.60) NMAX = 60
    NMAXI = NMAX/3
    XMIN = XMIN/60.
    XMAX = XMAX/60.
    XMHR = MHRT(K)
    XMHR = XMHR/10.
    XSI = XMHR/XSUM
    XAT = XSUM/XSORT
    NOS = NO(1,K)
    DO 550 J = 1, 20
    NO(J,K) = 100*NOS/N
    NOS = NOS + NO(J+1,K)
550 CONTINUE
    WRITE (6,9500) NUT
    DO 600 I = 1, 90
    I = SI - 1
    WRITE (6,9520) [L, (ITM(I,J,K), J = 1, 20)]
600 CONTINUE
    WRITE (6,9530) (I, I = 3, 57, 3)
    WRITE (6,9540) (NO(J,K), J = 1, 20)
    WRITE (6,9550) N, XSUM, XAVG, XMIN, XMAX, XSD
WRITE (6, 9560) NSORT, NOFIX, XBR
WRITE (6, 9570) XMHR, XSI, XAT
700 CONTINUE
GO TO 50
9000 FORMAT (215)
9100 FORMAT (8X, 14, 2(12, 14, 3, 12, 8X, 411, 19X, 14, 15, 213, 1X, 15, 1X, 11)
9500 FORMAT (1H1, 35X, 15H SYSTEM RECOVERY, //, 35X, 11H UNIT NUMBER, 15)
9520 FORMAT (3X, 12, 4X, 20(1X, A4))
9530 FORMAT (1H0, 10HOURS LT, 19(1X, I4), 8H GT 57)
9540 FORMAT (1X, 3HP =, 5X, 19(15, 3X, 15)
9550 FORMAT (1H0, 3HM =, 14, 3X, 12HTOTAL TIME =, F10.2, 3X, 9HAVG =, F8.2,
* 3X, 5HMIN =, F8.2, 3X, 5HMAX =, F8.2, 3X, 11HSTD. DEV. =, F8.2)
9560 FORMAT (1X, 13HTOTAL SORTIES, 15, 3X, 15, 1X,
* 20HSORTIES REQUIRED NO FIX TIME, 3X, 12HBREAK RATE =, F5.3)
9570 FORMAT (1X, 17HTOTAL MAN HOURS =, F8.1,
* 3X, 17HSATURATION INDEX =, F5.1,
* 3X, 27HAVE T/A (ALL MAINTENANCE) =, F5.1)
END
ENTRY
20 15
$IBSYS ENDJOB TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
Program 6

WORK CENTER RECOVERY
VI. WORK CENTER RECOVERY

THE RECOVERY PROGRAM

The work-center data the recovery program produces are: the touchdown time, the time the work center began the first job and ended the last job, the number of people at work in each ¼-hour trial period, as well as the conventional AFM 66-1 data. In plotting the data, touchdown time is set to zero, and the data are plotted from this point in time. The program computes not only the percentage of aircraft still not recovered each ¼-hour period, but also the percentage of aircraft being worked on during each period.

Figure 17 is a plot of one month's data. We see that 227 sorties were flown, resulting in 10 requests for Work Center number 1. The matrix shows how these 10 demands were met: one hour after touchdown there were 10 times when a 3-man team was required, etc. By following along the two lines PCB and PBF, we can compare the percentage of aircraft yet to be recovered (PCB) with those actually being worked on (PBF) at each point in time.

Among the facts revealed are that, although 80 percent of the aircraft are recovered within 1 hour after touchdown, by 1½ hours, 30 percent are still being worked on. Note, too, that the work center may not work on 100 percent of the aircraft at all times.

It is worthwhile to point out that a number of analyses become possible with work-center data of the type described. Because touchdown times are on each card, we can compare periods of light and exceedingly heavy loads, thus determining the impact of load on the work center. We can isolate special exercises for similar comparison. We can determine the effects of time of day, or day of week. Further, we can isolate specific sorties in which the aircraft was not touched for, say, more than six hours, and determine why by referring to the Aircraft Recovery time distributions (Program 4) and the System Recovery Program (Program 5). Thus we begin to get at the heart of those factors (manning, scheduling, and performance) that affect the operationally ready rates.
UNIT 1 FLIGHT-LINE DEMANDS FROM TIME OF TOUCH-DOWN (TD=0 HOURS)

|   | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12+
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MEAN 3.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

PDB 100 100 80 30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PSF 100 100 80 30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

227 SORTIES FLOWN
10 REQUESTS ON UNITS
THE PROBABILITY OF BEING REQUESTED IS 0.04
TOTAL MAIN-HOURS = 27.2
(PCB=PERCENT OF CRIPPLED BIRDS STILL SICK. PSF=PERCENT OF BIRDS BEING FIXED.)

Fig. 17 -- Flight-line demands from time of touch-down
FLIGHT PROGRAM DESCRIPTION

This program is written in standard FORTRAN IV language. Note the necessary changes indicated in the program listing to allow the program to run on IBM System 360 computers.

Input is from card and tape. The highest tail number to be read from tape and the highest unit number to be printed are read from a card as five-digit integers. The program reads the AIRCRAFT RECOVERY TAPE as many times as necessary to present a chart for each unit number. The information for as many as twenty units is accumulated each time the tape is read through.

An array ITN contains the counts of crew size versus response time for as many as twenty units, array ZMN contains the mean crew size versus time for each unit, and arrays NCB and NFX contain counts of disabled aircraft and aircraft being fixed at each time for each unit. When the last tail number desired has been processed, the charts for each type of failure that occurred in the current set of twenty units are printed and the procedure is resumed for the next set of twenty units.
DIMENSION ITM(15,25,20),NCO(25,20),NFX(25,20)
DIMENSION ZND(25,20),ZMN(25,20),NREQ(20),NHRT(20)
DIMENSION NT1(10),NT2(10),NT3(10),NT4(10)
DATA NLANK/1H/
DATA IZ1/4H 0/
DATA IZ2/4H 00/
DATA IZ3/4H 000/
DATA NT1(1)/4H0000/,NT1(2)/4H0001/,NT1(3)/4H0002/,NT1(4)/4H0003/
DATA NT1(5)/4H0004/,NT1(6)/4H0005/,NT1(7)/4H0006/,NT1(8)/4H0007/
DATA NT1(9)/4H0008/,NT1(10)/4H0009/
DATA NT2(1)/4H0000/,NT2(2)/4H0010/,NT2(3)/4H0020/,NT2(4)/4H0030/
DATA NT2(5)/4H0040/,NT2(6)/4H0050/,NT2(7)/4H0060/,NT2(8)/4H0070/
DATA NT2(9)/4H0080/,NT2(10)/4H0090/
DATA NT3(1)/4H0000/,NT3(2)/4H0100/,NT3(3)/4H0200/,NT3(4)/4H0300/
DATA NT3(5)/4H0400/,NT3(6)/4H0500/,NT3(7)/4H0600/,NT3(8)/4H0700/
DATA NT3(9)/4H0800/,NT3(10)/4H0900/
DATA NT4(1)/4H0000/,NT4(2)/4H1000/,NT4(3)/4H2000/,NT4(4)/4H3000/
DATA NT4(5)/4H4000/,NT4(6)/4H5000/,NT4(7)/4H6000/,NT4(8)/4H7000/
DATA NT4(9)/4H8000/,NT4(10)/4H9000/
C DELETE EVERYTHING BETWEEN THE ASTERISKS IF THE PROGRAM IS TO BE RUN
C ON A MACHINE THAT ALLOWS FEWER THAN FIVE HOLLERITH CHARACTERS/WORD.
C THESE CARDS SHOULD BE REMOVED TO RUN ON ANY IBM SYSTEM 360 COMPUTER.

DATA IZ0/5H0000/
DO 1 I=1,10
   NT1(I)=NT1(I)-IZ0
   NT2(I)=NT2(I)-IZ0
   NT3(I)=NT3(I)-IZ0
   NT4(I)=NT4(I)-IZ0
1 CONTINUE

C=================================================================================================

MIT=20
READ (5,9000) MAXT,MAXU
NSORT=MIT

50 REWIND 5
MIT=MIT+MIT
IF (MIT.GT.MAXU) STOP
DO 100 K=1,MIT
   NHRT(K)=0
   NREQ(K)=0
   DO 100 J=1,25
      ZNO(J,K)=0.0
      ZMN(J,K)=0.0
      NCO(J,K)=0
      NFX(J,K)=0
   100 ND1(J,K)=0
NSORT=0
150 READ (8,9100) IDAY,IHR,IMN,JDAY,JHR,JMN,ISC,ITA.
   */ MHR,MUF,JD,LET,IO,NC
IF (JD.NE.999) GO TO 250
200 IF (ITA.GE.MAXT) GO TO 500
    GO TO 150
250 IF (NC.ME.2) GO TO 150
    MNSRT=MNSRT+1
    MAIN=0
    ITD=JMN+60*(JHR+24*JDAY)
300 READ (8,9100) IDAY, IHR, JMN, JDAY, JHR, JMN, ISC, ITA,
      MHR, NUF, JD, IET, ID, NC
    IF (JD.EQ.999) GO TO 200
    IF (NC.ME.0) GO TO 350
310 CONTINUE
    MAIN=1
    IF (ID.EQ.2150) GO TO 325
    IF (ID.EQ.2500) GO TO 325
    IF (ID.EQ.2300) GO TO 325
    IF (ID.EQ.110) GO TO 325
    GO TO 300
325 K=NUF-NIT
    IF (K.LE.0) GO TO 300
    IF (K.GT.199) GO TO 325
    MREQ(K)=MREQ(K)+1
    MHR(K)=MHR(K)+MHR
    ISTD=JMN+60*(JHR+24*IDAY)
    NJSTD=JMN+60*(JHR+24*JDAY)
    *REC=ISTL-ITD
    J=REC/30+1
    J=J*1ET/9
    IF (J.JT.25) J=25
    IF (J1.GT.25) J1=25
    I=ISC
    IF (I.GT.15) I=15
    IN11=IN11+IN1(J,J,K)+1
    ZM(IS(J,K))=ZM(IS(J,K))+10
    ZI=1
    ZMN(J,K)=ZMN(J,K)+ZI
    DO 330 J2=1,J1
330 M=BJ2,K)-MGB(J2,K)+1
    DO 332 J2=J1+1
    MJ(J2,K)=NFX(J2,K)+1
332 NOC(J2,K)=NOC(J2,K)-1
    GO TO 300
350 IF (ISC.ME.2) GO TO 300
    IF (MAIN.EQ.0) GO TO 300
375 IF (JD.EQ.999) GO TO 200
    GO TO 150
500 CONTINUE
    DO 700 K=1,NIT
      MNU=MNU+K
      XNHR=XNHR(K)
      XMHR=XNHR/10.
      M=MREQ(K)
      IF (M.EQ.0) GO TO 700
ZM=N
Z5=NSORT
ZP=ZN/Z5
DO 510 J=1,25
ZMN(J,K)=ZMN(J,K)/ZNO(J,K)
NCB(J,K)=100*NCB(J,K)/N
NFX(J,K)=100*NFX(J,K)/N
DO 510 I=1,15
L=ITNI(I,J,K)
ITNI(I,J,K)=NLANK
IF (L.EQ.0) GO TO 510
L4=L/10
L3=L-10*L4
ITNI=NT3(L3+1)+NT4(L4+1)
IF (L.LT.10) ITNI=123-1ITNI
ITNI(I,J,K)=ITNI
510 CONTINUE
WRITE (6,9500) NUT
WRITE (6,9510) (NT3(I),I=2,10)
DO 600 I=1,15
II=16-I
WRITE (6,9520) II,(ITNI(I1,J,K),J=1,25)
600 CONTINUE
WRITE (6,9535) (ZMN(J,K),J=1,25,2)
WRITE (6,9550) (NCB(J,K),J=1,25)
WRITE (6,9560) (NFX(J,K),J=1,25)
WRITE (6,9570) NSORT
WRITE (6,9580) N
WRITE (6,9590) ZP
WRITE (6,9600) XMHR
WRITE (6,9610)
700 CONTINUE
GO TO 50
9000 FORMAT (215)
9100 FORMAT (8X,14,212,14,312,8X,14,19X,14,15,213,2X,15,1X,11)
9500 FORMAT (1H1,10X,4HUNIT,14,2X,2MPFLIGHT-LINE DEMANDS FROM TIME,
              1X,26HOF TOUCH-DOWN (TO=0 HOURS))
9510 FORMAT (1HO,9X,2HO0,946X,A2),6X,2HM10,6X,2HM11,6X,3H12+)
9520 FORMAT (1HO,2X,12,3X,25(2X,A2))
9530 FORMAT (1HO,7X,2514)
9535 FORMAT (1HO,4HMEAN,3X,13(F4,1,4X))
9550 FORMAT (1HO,4HPCE=,3X,25(1X,13))
9560 FORMAT (1X,4HPBF=,3X,25(1X,13))
9570 FORMAT (1HO,16,1X,17HSORTIES FLOWN)
9580 FORMAT (1X,16,1X,17HREQUESTS ON UNITS)
9590 FORMAT (4X,30HTHE PROBABILITY OF BEING REQUESTED IS,F4.2)
9600 FORMAT (4X,1HTOTAL MAN-HOURS =,F8.1)
9610 FORMAT (4X,4ZHPC=PERCENT OF Crippled Birds Still Sick.,
          2X,34HPBF=PERCENT OF BIRDS BEING FIXED.))
END
ENTRY
20 15 30

SUBSYS   ENDDJOB
VII. FAILURE LIST

The Failure List program provides a visual history of break-rate information. The break-rates are inferred from unscheduled maintenance performed (Fig. 18).

The unscheduled maintenance is produced by having the recovery program search each sortie for unscheduled maintenance (by two- or selected three-digit systems, or both). Each time a fix is encountered, it is recorded under the appropriate system number.

<table>
<thead>
<tr>
<th>DAY</th>
<th>UNIT</th>
<th>TNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.34</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>0.34</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>0.34</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0.38</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.42</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>0.42</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>0.43</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>0.55</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0.55</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>0.65</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>0.65</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.65</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>0.65</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>1.34</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1.40</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>1.43</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>1.57</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1.60</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2.43</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>2.65</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3.43</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3.58</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>3.65</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3.65</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>4.43</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>4.55</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>6.00</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>7.00</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>7.43</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>7.55</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>8.55</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>9.55</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>9.57</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>9.59</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>10.34</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>10.61</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>10.65</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>11.34</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>11.45</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>11.55</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>13.00</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>14.34</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>14.34</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>14.38</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>14.58</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>14.59</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>15.43</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>15.43</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>15.55</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>15.65</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 18
PROGRAM DESCRIPTION

This program is written in standard FORTRAN IV language.

Input is from card and tape. The times to start and stop printing, the number of cases, and the number of different tail numbers to be listed are read from a card. The times to start and stop contain four characters each, including decimal point; the number of tail numbers (less than 1000) and the number of bases (less than 10) are each read as four-digit integers. If the number of bases is zero or blank, all bases will be listed; otherwise, a card containing the specific base numbers to be listed is read, with each base number entered as a three-digit integer. If the number of specific tail numbers to be listed is zero or blank, all tail numbers will be listed; otherwise, cards each containing 24 three-digit integer numbers identifying the specific tail numbers are read. The program then reads the original PLANET ABC tape and prints the time, unit, and tail number for all failures concerning the specified bases and tail numbers.
PROGRAM SIBFTC

DIMENSION IBS(10), ITN(1000), IST(1000), ISTN(1000)
NT=0
NPR=0

100 REWIND 9
READ (5,9000) TEND,NBASE,NTAIL
IF (TEND.EQ.0.0) TEND=1.0E10
IF (NBASE.EQ.0) GO TO 200
READ (5,9010) (IBS(I),I=1,NBASE)

200 IF (NTAIL.EQ.0) GO TO 300
READ (5,9010) (ITN(I),I=1,NTAIL)

300 READ (9) L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,T
IF (L2.EQ.3) GO TO 100
IF (L2.EQ.1900) GO TO 400
IF (L2.EQ.2150) GO TO 600
IF (L2.EQ.2300) GO TO 600
IF (L2.EQ.2500) GO TO 600

350 IF (L11.LE.0) GO TO 300
DO 375 I=1,L11
375 READ (9) II
GO TO 300

400 IF (NBASE.EQ.0) GO TO 450
DO 425 I=1,NBASE
425 IF (IBS(I).EQ.L6) GO TO 450
GO TO 350

450 IF (L11.LE.0) GO TO 300
L11=L11-1
READ (9) II
IF (NTAIL.EQ.0) GO TO 500
DO 475 I=1,NTAIL
475 IF (ITN(I).EQ.I1) GO TO 500
GO TO 350

500 NT=NT+1
IST(NT)=L10
ISTN(NT)=I1
GO TO 350

600 IF (T.LT.T0) GO TO 350
IF (T.GT.TEND) GO TO 100
DO 650 I=1,NT
650 IF (IST(I).EQ.L10) GO TO 700
GO TO 350

700 II=ISTN(I1)
NPR=NPR-1
IF (NPR.GT.0) GO TO 800
NPR=50
WRITE (6,9500)

800 WRITE (6,9510) T,L8,II
GO TO 350

9000 FORMAT (2F4.0,2I4)
9010 FORMAT (2I3/)
9500 FORMAT (1H1,6X,12HFAILURE LIST,///,4X,15HDAY UNIT TND)
9510 FORMAT (3X,F6.2,2(2X,13))
END
ENTRY
0.0 0.0 0 40
1 2 3 4 5 6 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
$IBSYS ENDJOB TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
Program 8

COST/EFFECTIVENESS
VIII. COST/EFFECTIVENESS

The Cost/Effectiveness program reports the results of dividing the SITE and Logistics costs by the effectiveness. The measure of effectiveness is derived by subtracting the SITE(s) downtime from the total SITE(s) time for the simulated period; site time may be initialized to specify any time period, as well as base, or for the entire fleet of weapons.

INITIALIZATION

Table 5 lists the initialization requirements. Thirty-six arrays are used. Only twelve require values, however.

For initialization formats the user may use the example data (which are strictly hypothetical) contained with the program listing or refer to Part 1, Section II for the initialization instructions.

OUTPUT PROGRAM

The input to the program is the tape generated by the ABC simulator. The input tape consists of a 12-variable label record and is sometimes followed by a 10-variable detail record.

Any change to the personnel, equipment, or spare part quantities, after the initial assignment, causes a message to be printed. The message contains the quantity and type resource, the base, the cost, and the simulated time.

PERMANENT VARIABLES

EBASE - number of bases
SUM - site cost (by base)
PSUM - personnel cost (by base)
ESUM - equipment cost (by base)
SSUM - spare part cost (by base)
SITE - number of site types
COST - cost per site type
QTY - quantity of each site type
PTYPE - number of personnel types
### Table 5

**VARIABLE DESCRIPTION AND INITIALIZATION: COST/EFFECTIVENESS**

<table>
<thead>
<tr>
<th>Array Number</th>
<th>Number of Subscripts</th>
<th>Mode</th>
<th>Initialize to</th>
<th>Initialize Value in</th>
<th>Description of Variable to Be Initialized</th>
<th>Permanent System Variable Name</th>
<th>Entity Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>1</td>
<td>Z</td>
<td>1</td>
<td></td>
<td>Number of bases to be analyzed</td>
<td>ERASE E</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>V</td>
<td>8</td>
<td></td>
<td>Number of site types</td>
<td>SITE E</td>
<td></td>
</tr>
<tr>
<td>9-10</td>
<td>2</td>
<td>V</td>
<td>1 10-8</td>
<td></td>
<td>Cost of each site type</td>
<td>COST A</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>V</td>
<td>11</td>
<td></td>
<td>Quantity of each site type</td>
<td>QTY A</td>
<td></td>
</tr>
<tr>
<td>12-13</td>
<td>1</td>
<td>Z</td>
<td>1 11</td>
<td></td>
<td>Number of personnel types</td>
<td>PTYPE A</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>V</td>
<td>14</td>
<td></td>
<td>Cost of each personnel type</td>
<td>PCOST A</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>F</td>
<td>16</td>
<td></td>
<td>Number of equipment types</td>
<td>TYPE E</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>Z</td>
<td>1 14</td>
<td></td>
<td>Cost of each equipment type</td>
<td>ECOST A</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>V</td>
<td>17</td>
<td></td>
<td>Number of spare part types</td>
<td>STYPE E</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>F</td>
<td>17</td>
<td></td>
<td>Cost of each spare part type</td>
<td>SCOST A</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>Z</td>
<td>1 17</td>
<td></td>
<td>SUTY A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>F</td>
<td>V</td>
<td></td>
<td>ENDIN = end of report</td>
<td>FINISH E</td>
<td></td>
</tr>
<tr>
<td>21-27</td>
<td>0</td>
<td>Z</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>F</td>
<td>Z</td>
<td></td>
<td>Insert code &quot;0&quot; = Base statistics</td>
<td>FLEET E</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>F</td>
<td>1</td>
<td></td>
<td>&quot;1&quot; = Entire fleet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>F</td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The table format is read from left to right, top to bottom, with each cell representing a specific entry or code.
PCOST - cost per personnel type
PQTY - quantity of each personnel type
ETYPE - number of equipment types
ECOST - cost per equipment type
EQTY - quantity of each equipment type
STYPE - number of spare part types
SCOST - cost per spare part type
SQTY - quantity of each spare part type
FINSH - report time
SITID - first variable of detail record
CODE - second variable of label record
BASE - sixth variable of label record
V1 - seventh variable of label record
V2 - eighth variable of label record
V3 - ninth variable of label record
ETIME - twelfth variable of label record
TST - total site time by base
TTST - total site time for fleet
NOR - total site downtime by base
TNOR - total site downtime for fleet
CEFT - cost/effectiveness by base
TCEFT - total cost/effectiveness for fleet
TSUM - total logistics cost for fleet
FLEET - flag to determine if statistics are by base or for fleet
FOLLOW - flag to determine if detail record follows label record

SETS
NAME SET USED FOR PHYSICAL ENTITIES
ONE SUBSCRIPT RANKED BY SITE ID NUMBER
OWNER SIMSCRIPT SYSTEM
MEMBER WORK: ID = site ID number
START = downtime of site
STOP = uptime of site
CT100 = count of 100 records
CT200 = count of 200 records
C  RECORD DOWN TIME OF SITE
+T WORK 8              T ID 1  I  1EBASE E  I  SET 1  *ID  L
         T START 2  I  2FSET 1  I  
         T STOP 3  I  3LSET 1  I  
         T C100 4  I  4SUM 1  I  
         T C120 5  I  5PSUM 1  I  
         T PSET 6  I  6ESUM 1  I  
         T SSET 7  I  7SUM 1  I  
         8SITE E  I  9COST 1  F  
         10QTY 2  I  11PTYPE E  I  
         12PCOST 1  F  13PQTY 2  I  
         14ETYPE E  I  15ECOST 1  F  
         16EQTY 2  I  17STYPE E  I  
         18SCOST 1  F  19SQTY 2  I  
         20FINISH 0  F  21SITID 0  I  
         22CODE 0  I  23BASE 0  I  
         24V1  0  I  25V2  0  I  
         26V3  0  I  27ETIME 0  F  
         28TST 1  F  29TTSST 0  F  
         30NOR 1  F  31TNOR 0  F  
         32CEFT 1  F  33TCEFT 0  F  
         34TSUM 0  F  35FLEET 0  I  
         36FOLLOW 0  I  

*IDFTC MAIN
   MAIN ROUTINE
C
C PLANET - COST/EFFECTIVENESS PROGRAM
C INPUT - ABC BINARY OUTPUT TAPE
C
C REWIND 9
LET KFLAG = 0
1 IF (FOLLOW) EQ (1), GO TO 3
C READ LABEL RECORD
X 2 READ (9) 11,12,13,14,15,16,17,18,19,110,111,112
STORE 12 IN CODE
STORE 16 IN BASE
STORE 17 IN V1
STORE 18 IN V2
STORE 19 IN V3
STORE 111 IN FOLLOW
STORE 112 IN ETIME
GO TO 5
C READ DETAIL RECORD  
X 3 READ (9) 11,12,13,14,15,16,17,18,F9,F10  
    LET FOLLOW = 0  
    GO TO 2  
C 5 IF (ETIME) GE (FINISH), GO TO 990  
    IF (KFLAG) EQ (1), GO TO 8  
    IF (ETIME) EQ (0.0), GO TO 8  
C COMPUTE INITIAL LOGISTICS COSTS  
    CALL LIMPUT  
    LET KFLAG = 1  
C TEST FOR RECORD TYPE  
C 8 IF (CODE) EQ ( 3), GO TO 990  
    IF (CODE) EQ (10), GO TO 10  
    IF (CODE) EQ (20), GO TO 20  
    IF (CODE) EQ (30), GO TO 30  
    IF (CODE) EQ (100), GO TO 100  
    IF (CODE) EQ (200), GO TO 200  
    IF (CODE) EQ (1200), GO TO 1200  
    IF (CODE) EQ (1210), GO TO 1210  
    IF (CODE) EQ (1220), GO TO 1220  
    GO TO 1  
C INITIAL PERSONNEL QUANTITY  
10 IF (PQTY(BASE,V1)) LS (V3), LET PQTY(BASE,V1) = V3  
    GO TO 1  
C INITIAL EQUIPMENT QUANTITY  
20 IF (EQTY(BASE,V1)) LS (V3), LET EQTY(BASE,V1) = V3  
    GO TO 1  
C INITIAL SPARES QUANTITY  
30 IF (SQTY(BASE,V1)) LS (V3), LET SQTY(BASE,V1) = V3  
    GO TO 1  
C DEMAND ARRIVAL  
100 CALL DOWN  
    GO TO 1  
C END OF MAINTENANCE  
200 CALL UP  
    GO TO 1  
C CHANGE IN PERSONNEL  
1200 LET VALUE = FLOAT(V3) * PCOST(V2)  
    LET PSUM(BASE) = PSUM(BASE) + VALUE  
    CALL PREPVALUE  
    GO TO 1  
C CHANGE IN SPARE PARTS  
1210 LET VALUE = FLOAT(V3) * SCOST(V2)  
    LET SSUM(BASE) = SSUM(BASE) + VALUE  
    CALL SREPVALUE
GO TO I

CHANGE IN EQUIPMENT
1220 LET VALUE = FLOAT(V3) * ECOST(V2)
   LET ESUM(BASE) = ESUM(BASE) + VALUE
   CALL EREPT(VALUE)
   GO TO I

END OF SIMULATION
990 CALL DONE
END

*IBFTC COMPUT
SUBROUTINE COMPUT

C
C COMPUTE INITIAL PERSONNEL COSTS PER BASE
DO TO 10, FOR EACH EBASE I
DO TO 9, FOR EACH PTYPE J
LET PSUM(I) = PSUM(I) + (FLOAT(PQTY(I,J)) * PCOST(J))
9 LOOP
10 LOOP

C
C COMPUTE INITIAL EQUIPMENT COSTS PER BASE
DO TO 20, FOR EACH EBASE I
DO TO 19, FOR EACH ETYPE J
LET ESUM(I) = ESUM(I) + (FLOAT(EQTY(I,J)) * ECOST(J))
19 LOOP
20 LOOP

C
C COMPUTE INITIAL SPARE PART COSTS PER BASE
DO TO 30, FOR EACH EBASE I
DO TO 29, FOR EACH STYPE J
LET SSUM(I) = SSUM(I) + (FLOAT(SQTY(I,J)) * SCOST(J))
29 LOOP
30 LOOP
RETURN
END

*IBFTC DOWN
SUBROUTINE DOWN

C
C DEMAND ARRIVAL
C RECORD DOWN TIME OF SITE
C READ DETAIL RECORD

C
READ (9) I1, I2, I3, I4, I5, I6, I7, I8, F9, F10
LET FOLLOW = 0
STORE II IN SITID

C
TEST TO SEE IF SITE IS ALREADY DOWN
FIND FIRST, FOR EACH WORK OF SET(BASE), WITH (ID(WORK)) EQ (SITID)
X, IF NONE, GO TO 10
LET CT100(I) = CT100(WORK) + 1
RETURN

C
10 CREATE WORK
LET ID(WORK) = SITID
LET START(WORK) = ETIME
LET CT100(WORK) = 1
FILE WORK IN SET(BASE)
RETURN
END

*IBFCTC UP

SUBROUTINE UP

END OF MAINTENANCE
RECORD UP TIME OF SITE
READ DETAIL RECORD
READ (9) 11,12,13,14,15,16,17,18,F9,F10
LET FOLLOW = 0
STORE 11 IN SIT1D

TEST TO SEE IF THIS IS LAST OUTSTANDING DEMAND FOR SITE
FIND FIRST, FOR EACH WORK OF SET(BASE), WITH (ID(WORK)) EQ (SIT1D)
IF (CT100(WORK)) EQ (1), GO TO 10
LET CT200(WORK) = CT200(WORK) + 1
IF (CT200(WORK)) LS (CT100(WORK)), RETURN

10 LET STOP(WORK) = ETIME
LET NOR(BASE) = NOR(BASE) + (STOP(WORK) - START(WORK))
REMOVE WORK FROM SET(BASE)
DESTROY WORK
RETURN
END

*IBFCTC DONE

SUBROUTINE DONE

COMPUTE SITE COSTS AND TOTAL SITE TIME PER BASE

DO TO 10, FOR EACH EBASE I
LET TOTAL = 0.0
DO TO 9, FOR EACH SITE J
LET SUM(I) = SUM(I) + (FLOAT(QTY(I,J)) * COST(J))
LET TOTAL = TOTAL + FLOAT(QTY(I,J))
9 LOOP
LET TST(I) = TOTAL * FINSH
10 LOOP

COMPUTE NOR TIME FOR OUTSTANDING DEMANDS

DO TO 20, FOR EACH EBASE I
19 IF SET(I) IS EMPTY, GO TO 20
REMOVE FIRST WORK FROM SET(I)
LET NOR(I) = NOR(I) + (FINSH - START(WORK))
DESTROY WORK
GO TO 19
20 LOOP

TEST TO SEE IF STATISTICS ARE BY BASE OR FOR ENTIRE FLEET
IF (FLEET) NE (0), CALL SUMRY

DO TO 30, FOR EACH EBASE I
LET CEFT(I) = (SUM(I)+PSUM(I)+ESUM(I)+SSUM(I))/ (TST(I)-NOR(I))
30 LOOP
   CALL OUTPUT
C
   REWIND 9
   CALL EXIT
END
*IBFTC SUMRY
SUBROUTINE 'SUMRY
C
   COMPUTE STATISTICS FOR ENTIRE FLEET
C
   DO TO 10, FOR EACH EBASE I
   LET TSUM = TSUM + SUM(I) + PSN(1) + TSU(1) + SSU(1)
   LET THOR = THOR + NOR(I)
   LET TTST = TTST + TS(I)
10 LOOP
C
   LET TCEFT = TSUM / (TTST - THOR);
   CALL TOUT
   REWIND 9
   CALL EXIT
END
*IBFTC PREPT
REPORT PREPT(VALUE)
X
   *** TYPE *** PERSONNEL AT BASE *** COSTING ******.**
   V3    V2    BASE
X
   TIME = ******.****** 1 x
   ETIME
END
*IBFTC EREPT
REPORT EREPT(VALUE)
X
   *** TYPE *** EQUIPMENT AT BASE *** COSTING ******.**
   V3    V2    BASE
X
   TIME = ******.****** 1 x
   ETIME
END
*IBFTC SREPT
REPORT SREPT(VALUE)
X
   *** TYPE *** SPARE PART AT BASE *** COSTING ******.**
   V3    V2    BASE
X
   TIME = ******.****** 1 x
   ETIME
END
*IBFTC OUTPUT
REPORT OUTPUT
X
   COST/EFFECTIVENESS FOR EACH EBASE 1
X
X
X
```plaintext
R  SIMULATED DAYS
FINSH
CEFT(1)

*IBFTC TOUT
REPORT TOUT
X

COST/EFFECTIVENESS FOR
FLEET = **

R  SIMULATED DAYS
FINSH
CEFT

BENTRY  END
ENTRY  MAIN

1  36
   2  7 1 2 2 1
   8  0 0
   9  1 R 2 8
   50.00
    25.00
  10  2 R 2 1 2 8 RN 2(16)
   4  6
   11  0 R
   12  1 R 6 11
   12.00
    12.00
    14.00
    14.00
    16.00
    16.00
   13  2 2 2 1 6 11
   14  0 R
   15  1 R 4 14
    6.00
    9.00
    10.00
    11.00
   16  2 2 2 1 4 14
   17  0 R
   18  1 R 5 17
   19  2 2 2 1 5 17
```
20 0 R
71 27 0 Z
28 1 Z 2 1
29 0 Z
30 1 Z 2 1
31 0 Z
32 1 Z 2 1
33 0 Z
34 0 Z
35 0 Z
36 0 Z
END INITIALIZATION

-200- 25.00000 ENDSIM

FLEET
Program 9

BASE SHOPS MAINTENANCE

CAPABILITY
IX. BASE SHOPS MAINTENANCE CAPABILITY

The Base Shops Maintenance Capability program is used to display the outputs from the Bench Repair Simulator. The report consists of five parts: the input to each shop, its output, and the reparable repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of simulation.

An example of the output display is shown in Figs. 10 to 23. Figure 19 is the display of shop statistics, showing the system (or Unit or item) arrivals and departures, reparables in process, and repair cycle times. A separate display is presented for each shop. The example display is for shop No. 3.

The first line of data shows the time at which the statistics were taken. Notice that the report is for day ending 14.000. Since "time" began at time 0.000 in the simulation, and the report is initialized for seven-day periods, the fourteenth day will end at time 14.000 (not 14.999). The next line entry shows the number of reparables that entered the shop (15) for the period and the sum of all reparables entering the shop (30) as of the report period.

The third line entry is the serviceables departing the shop (returned to serviceable stock) (8), during the period, and the sum of all items processed to date (20). The difference between the arrivals and departures is presented as the number in process (10).

The next line displays a distribution of the reparables in process. The average time in process (for the twenty that were processed) is 4.75 days, the maximum time was 10 days, and the minimum 1 day. The standard deviation for the distribution is 2.37.

The repair time distribution is presented both for the period (just 7 days) and accumulated for the fourteen days.

Figure 20 displays the activity Queueing Factors for the period. Column 1 lists the activities in sequence. Columns 2, 3, and 4 list the
distribution of the quantity of reparables processed by each activity for the period. Column 5 lists the average time that the reparables spend in queue behind each activity awaiting some resource. The average queue time is displayed as work time; i.e., off-shift time is not included.

Figure 21 is the Personnel Utilization report. For each personnel type, listed in col. 1, the sum of all personnel on duty for all shifts of the period (of course, the period may be only one shift) is presented in col. 2. The utilization factor, which is the time actually engaged in a process divided by the total duty time available, is presented in col. 3 for each personnel type. The balance of the display is devoted to the man-hours used at each activity for each personnel type. For example, personnel type 2 worked at activities 5, 10, and 11, and a total of 93.25 man-hours were used during the simulation period (seven days).

Figure 22 is the Equipment Utilization report. By equipment type, listed in col. 1, the quantity is listed in col. 2; cols. 3, 4, and 5 list the time the equipment was used, the idle time, and the downtime (all in decimal-days). Note that the summation of these three columns is equal to 14 equipment days for Equipment types 1 and 2, and 2' equipment days for Equipment type 3. This is the total time available for the equipment. Off-shift time is not deducted.

Column 6 is a count of the number of times the shop equipment failed during the period (in this example, 7 days). Column 7 is the utilization factor for the equipment, computed by dividing the total time available (equipment days) into the time in use. Column 8 lists the activities where the equipment was used.

Figure 23 is the display of the Queueing factors for each component spare part. Column 1 lists the spare part ID number. Column 2 lists the quantity or authorized stock level of each spare part. Column 3 lists the number of demands for each spare part during the period. Column 4, 5, and 6 list the distribution of the quantity of unfulfilled demands (average, maximum, and minimum) for each spare part type. Column 7 lists the average queue time—the average time required to fill the demand.
SHOP 3
SYSTEM ARRIVALS, DEPARTURES, IN-PROCESS, AND REPAIR CYCLE TIMES

DATA FOR PERIOD ENDING DAY 14.000

REPARABLES ENTERING SHOP THIS PERIOD 15, TO DATE 30
SERVICEABLES DEPARTING SHOP THIS PERIOD 8, TO DATE 20
NUMBER IN PROCESS 10

<table>
<thead>
<tr>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
<th>STD DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.75</td>
<td>10</td>
<td>1</td>
<td>2.37</td>
</tr>
</tbody>
</table>

REPARABLES IN PROCESS

REPAIR TIME

THIS PERIOD 1.99 3.03 0.96 0.69
TO DATE 1.37 3.03 0.88 0.67

Fig. 19

ACTIVITY QUEUEING FACTORS FOR PERIOD JUST COMPLETED

<table>
<thead>
<tr>
<th>ACT. NO.</th>
<th>NO. OF REPS IN QUEUE</th>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
<th>AVG QUEUE TIME (IN WORK-HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.33</td>
<td>3.0</td>
<td>0.0</td>
<td>0.40</td>
<td>0.80</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>3.0</td>
<td>0.0</td>
<td>0.43</td>
<td>0.63</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>1.0</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>1.0</td>
<td>0.0</td>
<td>0.09</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>1.10</td>
<td>9.0</td>
<td>0.0</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>6</td>
<td>0.01</td>
<td>1.0</td>
<td>0.0</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>7</td>
<td>3.18</td>
<td>8.0</td>
<td>0.0</td>
<td>1.38</td>
<td>1.38</td>
</tr>
<tr>
<td>8</td>
<td>0.00</td>
<td>1.0</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>4.15</td>
<td>10.0</td>
<td>0.0</td>
<td>1.44</td>
<td>1.44</td>
</tr>
<tr>
<td>10</td>
<td>0.27</td>
<td>3.0</td>
<td>0.0</td>
<td>0.38</td>
<td>0.38</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>1.0</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>2.0</td>
<td>0.0</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>13</td>
<td>0.01</td>
<td>1.0</td>
<td>0.0</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Fig. 20
PERSONNEL UTILIZATION

<table>
<thead>
<tr>
<th>TYPE</th>
<th>UTIL</th>
<th>NO. HOURS</th>
<th>MAN-</th>
<th>NO. HOURS</th>
<th>MAN-</th>
<th>NO. HOURS</th>
<th>MAN-</th>
<th>NO. HOURS</th>
<th>MAN-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>42</td>
<td>0.09</td>
<td>1</td>
<td>4.41</td>
<td>2</td>
<td>11.82</td>
<td>3</td>
<td>4.63</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>0.32</td>
<td>5</td>
<td>20.40</td>
<td>10</td>
<td>66.10</td>
<td>11</td>
<td>8.75</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>0.27</td>
<td>4</td>
<td>32.03</td>
<td>9</td>
<td>40.70</td>
<td>10</td>
<td>32.05</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>0.40</td>
<td>6</td>
<td>32.22</td>
<td>7</td>
<td>24.18</td>
<td>8</td>
<td>22.05</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 21

EQUIPMENT UTILIZATION

<table>
<thead>
<tr>
<th>TYPE</th>
<th>QUANTITY</th>
<th>TIME IN USE</th>
<th>IDLE TIME</th>
<th>DOWN TIME</th>
<th>NO. OF FAILURES</th>
<th>UTILIZATION FACTOR</th>
<th>ACTIVITY NO.</th>
<th>WHERE USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.67</td>
<td>13.15</td>
<td>0.18</td>
<td>3</td>
<td>0.048</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.26</td>
<td>12.35</td>
<td>0.39</td>
<td>11</td>
<td>0.060</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.70</td>
<td>19.29</td>
<td>0.01</td>
<td>1</td>
<td>0.081</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 22

QUEUEING FACTORS BY COMPONENT SPARES TYPE

<table>
<thead>
<tr>
<th>COMPONENT TYPE</th>
<th>QUANTITY</th>
<th>DEMANDS THIS PERIOD</th>
<th>NO. OF REPS IN QUEUE</th>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
<th>AVG QUEUE TIME (IN WORK-HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>26</td>
<td>0.00</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>0.96</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>21</td>
<td>4.30</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>14</td>
<td>0.00</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>27</td>
<td>12.79</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Fig. 23
INITIALIZATION

The Base Shops Maintenance Capability report program requires the initialization of 83 variables. Only 12 require values, however. Table 5, Variable Description and Initialization Table--Base Shop Capability, contains the information required to initialize the report program. An example initialization data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

The input to this program is the binary tape generated by the Bench Repair Simulation Program; this tape is read from logical unit #9.

The input tape consists of 12-word label records with the following format:

Word 1 - irrelevant.
Word 2 - IDD - a four-digit number identifying the "event" or "occurrence" represented by this record.
Words 3, 4, 5 - irrelevant.
Word 6 - INBASE - the base number.
Words 7, 8, 9, 10 - IV1, IV2, IV3, LADDR. These fields are used to store various items of information, depending on the value of IDD.
Word 11 - INDIC - 1 if the next record is a detail record (to be skipped), 0 otherwise.
Word 12 - RTIME - current simulated time.

When a label record is read, the value of INBASE is compared with the constant permanent attribute called BASE; if they are unequal, the record is skipped. (Thus it would require \( n \) runs of this analysis program to process all the data from an \( n \)-base simulation run, each time changing the value of BASE.)

If the new RTIME is greater than the previous one, subroutine CLOCK is called to check for the end of the operating shift and the end of the report period. If the report period has ended, subroutine ENDPRD is called to generate the reports. Subroutine CLOCK also updates TIME, which is the actual work time elapsed since the beginning of
<table>
<thead>
<tr>
<th>VARIABLE DESCRIPTION AND INITIALIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BASE SHOP CAPABILITY</strong></td>
</tr>
</tbody>
</table>

Table 6

-207-
simulation. (The automatically defined system variable TIME is used, in order to take advantage of the ACCUMULATE statement.)

Then the appropriate subroutine is called to process the label record. To each significant IDD number, there corresponds a subroutine: e.g., subroutine NEXTAC is called whenever IDD equals 4400. If IDD does not match any of the significant numbers, it is skipped.

If IDD = 3, the end of simulation has been reached; the program terminates after writing the last set of reports.

Error tests intended for the debugging phase have been left in the program, sprinkled throughout. If an error is encountered, this means that something is amiss in this program, in the simulation program, or in the initialization deck. Subroutine ERROR is called, which terminates after outputting the current value of RTIME and a four-letter abbreviation identifying the routine in which the error was detected. For instance, "REA2" refers to the second error condition in subroutine READY.

Subroutine SNAP outputs a "snapshot" of all permanent and temporary variables, as an aid to debugging. The user may insert, at any point, a call to SNAP with an identifier of one to four letters and/or digits; e.g., CALL SNAP (4HNAME). In this example, "NAME" is the identifier. In the current version of the program, ERROR calls SNAP before terminating.

PERMANENT VARIABLES

This list is complete except for attributes denoting first-of-set or last-of-set, and attributes used only to keep track of time in an ACCUMULATE statement such as TQSZA (these always have names beginning with "T").

RTIME - current simulated time; it is obtained from each label record as it is read in.

STIME - the "RTIME" of the previous label record.

TIME - (a variable automatically defined by the system) - number of work days elapsed since the beginning of simulation. For example, suppose there are 40 work-hours in a week. Then if RTIME = 7.0, TIME will be equal to 1.6667 or 1-2/3 (which is 40 divided by 24).
PTIME - the value of "TIME" at the end of the previous report period.
ENDSH - the "RTIME" at which the current shift will end.
ENDPD - the "RTIME" at which the current period will end.
ETIME - the value of "RTIME" at the end of the previous report period.
CURPD - the length in work days (using "TIME") of the period just completed.
CURP - the length in simulated time (using "RTIME") of the period just completed.
CURSH - number of current shift (on a weekly cycle).
CURAC - activity number associated with current label record.
CURSP - spare part number associated with current label record.
CUREP - I.D. number of REP associated with current label record.
SHOP - permanent entity, of which the following are attributes:
  RIN - number of reps entering this shop this period.
  ROUT - number of reps leaving shop this period.
  TRIN - total number of reps in shop (since the beginning of simulation).
  TROUT - total number of reps that have left this shop.
  MAXR - maximum number of reps in shop this period.
  MINR - minimum number of reps this shop this period.
  RIP - number of reps currently in process in this shop.
  RIPS - a running sum of all the values that RIP has assumed during this period.
  RIPSQ - a running sum-square total of all the values that RIP has assumed during this period. E.g., if RIP has had the values 2, 3, 4, 3, 2 in this period, then RIPS is 2 + 3 + 4 + 3 + 2 or 14, and RIPSQ is 2^2 + 3^2 + 4^2 + 3^2 + 2^2 or 42.
  RTS - sum of the repair times of all reps leaving shop this period.
  RTSQ - sum of squares of repair times of all reps leaving shop this period.
  TRTS - sum of RTS for all periods to date.
  TRTSQ - sum of RTSQ for all periods to date.
  MXRT - maximum repair time for shop this period.
MINRT - minimum repair time for shop this period.
MAXRT - maximum repair time for shop, all periods.
MINRT - minimum repair time for shop, all periods.

ACTIV - activity; a permanent entity, of which the following are attributes:
QSZA - current queue size at this activity.
CQSZA - cumulative total of QSZA, this period.
MAXSA - maximum value of QSZA, this period.
MINSA - minimum value of QSZA, this period.
TXAQ = "time in activity queue, summed;" the total time, in work days, that reps have spent in the queue for this activity.
AVQS - average queue size at this activity.
AVTAQ - average time in queue for this activity.
AQOUT - number of reps that have left the queue of this activity during this period.

PTYPE - personnel type; a permanent entity, of which the following are attributes:
QTYS - total number of this type of personnel.
CQTY - number of man-days for this personnel type for this period.

ETYPE - equipment type; a permanent entity with the following attributes:
QTYE - total quantity of this equipment type.
MFAL - number of failures of this type of equipment during this period.
INUSE - quantity of this equipment type currently in use.
CINUS - cumulative total of INUSE, this period.
DOWN - quantity of this type of equipment that is currently down.
CDOWN - cumulative total of DOWN, this period.

SPTYP - spare part type; a permanent entity with the following attributes:
QTYSP - quantity of spares of this type available at beginning of simulation.
DMAND - number of demands for this type of part during this period.
FILL - number of times that such a demand was filled.
QSP - queue size for this type of part.
CQSP - cumulative total of QSP, this period.
MKQSP - maximum value of QSP, this period.
MNQSP - minimum value of QSP, this period.
TISQS - total time, in work days, that reps have spent in the queue for this type of part.
AVQSP - average value of QSP, this period.
AVTSQ - average time in queue for this type of spare part.
RPTYP - rep type; a permanent entity with the following attribute:
     SHPNO - number of the shop to which this type of rep belongs.
SHIFT - a permanent entity with the following attribute:
     SCHED - 1 if this is a work shift; 0 if this is an off shift.
QTYPR - a permanent attribute with two subscripts:
     first subscript: PTYPE
     second subscript: SHIFT
     meaning: the quantity of personnel of this type, on duty during this shift.
LENSH - the length of a shift.
PEROD - the length of a report period.
BASE - the number of the base for this run; all label records pertaining to any other base will be ignored.

TEMPORARY VARIABLES
REP - a temporary entity with the following attributes:
     QTDE - the value of "TIME" when the rep entered the queue for an activity.
     BTDE - the value of "RTIME" when the rep entered the system.
     IDNO - the I.D. number of the rep: a number obtained from the label record, representing the absolute storage address of the rep in the simulation run.
     QFLAG - a number which is equal to zero unless the rep is in the queue for an activity, in which case QFLAG equals the number of that activity.
SFLAG - equal to zero unless rep is in the queue for a spare part, in which case SFLAG equals the number of that type of part. 
SPTIM - the value of "TIME" when the rep entered the queue for a spare part. 
PLOAD, SLOAD, PACTQ, SACTQ - attributes associated with the sets LOAD and ACTQ. 

DUMMY - a temporary entity whose purpose is to save information to be output in Table 4. It has two attributes: 
ACNO - the number of an activity at which this type of equipment is to be used. 
SSET - successor in the set called "SET." 
ENTRY - a temporary entity having to do with the utilization of personnel at different activities. Its attributes are: 
ACNO - the number of an activity at which this type of personnel is used. 
WKING - number of personnel of this type working at activity whose number equals ACNO. 
CWKING - cumulative total of WKING, this period. 
TWKING - the value of "RTIME" when CWKING was last updated. 
PLIST, SLIST - attributes associated with the set called "LIST." 

SETS
LOAD - a set with one subscript, ranked on BTIME. 
owner: SHDP 
member: REP 
The LOAD of each SHDP consists of all the reps that are currently in process in that shop. 

ACTQ - a set with one subscript, ranked on RTIME. 
owner: ACTIV 
member: REP 
ACTQ is the queue of all reps currently waiting at an activity. 

SET - a FIFO set with one subscript. 
owner: ETYPE 
member: DUMMY
SET is the set of all activities at which this type of equipment can be used. This information is to be output in Table 4.

LIST - a set with one subscript, ranked on ACNO.

owner: PTYPE
member: ENTRY

LIST has one ENTRY for each activity at which this type of personnel has been used during this report period.

**Standard Names for Local Variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IACNO</td>
<td>always means activity number</td>
</tr>
<tr>
<td>IEQNO</td>
<td>always means equipment number</td>
</tr>
<tr>
<td>IPERNO</td>
<td>always means personnel number</td>
</tr>
<tr>
<td>ISPNO</td>
<td>always means spare part number</td>
</tr>
<tr>
<td>IREPNO</td>
<td>always means rep number</td>
</tr>
<tr>
<td>ID or IDREP</td>
<td>always means I.D. number of rep</td>
</tr>
<tr>
<td>IQTY</td>
<td>always means quantity or number</td>
</tr>
</tbody>
</table>
1 REP 8
  T CTIME 1 F
  T ATIME 2 F
  T ICNC 3 I
  T CFLAG 41/2 I
  T SFLAG 42/2 I
  T SPTIP 5 F
  T SLCAD 6 I
  T SACTG 7 I
  T PLCAD 81/2 I
  T PACTG 82/2 I

+1 DLPMY2
  T SSET 2 I

+1 ENTRY8
  T ACNC 1 I
  T WNING 2 F
  T CKNNG 3 F
  T TKNG 4 F
  T PLIST 5 I
  T SLIST 6 I

  1RTIME F
  2STIME F
  3PTIME F
  4ERESP F
  5SECPEC F
  6CLRPEC F
  7CURSH I
  8CURAC I
  9CURSP I
  10SPCCP E
  11FLCFC I
  12LCFCF I
  13RIN I
  14RCUT I
  15TRIN I
  16TRPCT I
  17TRAPI I
  18TRPII I
  19RIP I
  20RIFS I
  21RIIFSC I
  22PTS I
  23RTSC I
  24RTST I
  25TRTSC I
  26PRHRT I
  27PRHDR I
  28TPRHT I
  29TPHRD I
  30CACTIV F
  31CSZI F
  32CCSZI F
  33TCSZI F
  34TPACS F
  35TPACS F
  36TACSC F

  LCACI *ETIME L
  ACTQI *ETIME L

  SET I *
-215-

32AVCSA I F
36AVTAC I F
37FACTC I I
38LACTC I I
39ACCUT I I
40OPTYPE E
41FLIST I I
42LIST I I
43GTS I I
44CTY I F
45TCTY I F
46ETYPY E
47CTY I I
48FAIL I I
50IAUSE I F
51CANUS I F
52TINUS I F
53CCWA I F
54CCWAI F
55TECGFI F
56FSET I I
57LSET I I
60STYPY E
61CTYSP I I
62DPANC I I
63FILL I I
64FSPC I I
65LSPC I I
66CSP I F
67CGSP I F
68CTSP I F
69PKCSP I F
70PRCSP I F
71TSCS I F
72AVGSP I F
74YSCS I F
75CCUREP I F
74CCURPF F
75ETIPY F
76RTYPY E
77SMPCAC I F
78SMRFT E
79SHECIC I F
80GTYPY 2 F
81AEHCIC FC
82PERCC FC
83RASE IC

*IBFTC PAIN

MAIN RTCUT
CALL PRELIM

C .......READ A LABEL RECORD
READ (9) K,ICODE,N,K,INPACE,IV1,IV2,IV3,1ADDR,INCIC,T
LET RTYPE = T

C .......IF THERE IS A DETAIL RECORD, SKIP OVER IT
IF (INCIC.EQ.1) REAC (9) JUNK

C .......TERMINATE IF AN ENDSIP RECORD (WITH ICC=3) IS ENCOUNTERED
IF (1IOD) NE (3), GC FC 30
CALL CLCC
LET ENPDO = RTYPE
CALL ENDPRO
STOP

C       SKIP THIS RECORD IF IT DOES NOT PERTAIN TO THE RIGHT BASE

3C      IF (INBASE) NE (BASE), GC TO 10
        IF (RTIME) GR (STIME), CALL CLCK

C       CALL THE APPROPRIATE ROUTINE FOR THIS ICC NUMBER

IF (IDD) EC (4000), GO TC 50
IF (IDD) EC (4002), GC TC 52
IF (IDD) EC (4003), GC TC 54
IF (IDD) EC (4004), GC TC 56
IF (IDD) EC (4005), GO TC 58
IF (IDD) EC (4200), GC TC 60
IF (IDD) EC (4401), GC TC 62
IF (IDD) EC (4402), GC TC 64
IF (IDD) EC (4403), GC TC 66
IF (IDD) EC (4404), GC TC 68
IF (IDD) EC (4405), GC TC 70
IF (IDD) EC (4406), GC TC 72
IF (IDD) EC (4407), GC TC 74
IF (IDD) EC (4408), GC TO 76
IF (IDD) EC (4700), GC TO 78
IF (IDD) EC (4450), GC TC 80
IF (IDD) EC (4451), GC TC 82
IF (IDD) EC (4452), GC TC 84
IF (IDD) EC (4453), GC TC 86

C       FOR ANY OTHER VALUE OF ICC, SKIP THIS RECORD

GO TC 10

9C      CALL ACCTVY (IV1)
        GO TC 1C

92      CALL ECATAC (IV1)
        GC TC 10

94      CALL PRISHEL (IV1, IV2, IV3)
        GO TC 10

96      CALL ECUP (IV1, IV2)
        GC TC 10

98      CALL SPARES (IV1, IV2)
        GC TC 10

6C      CALL ARRIV (IV1, IADDR)
        GO TC 10

62      CALL DEPART (IV1, IADDR)
        GO TO 10

64      CALL NEXTAC (IV1, IV3, IACCR)
        GC TC 10

66      CALL READY (IV3)
        GO TO 10

68      CALL ASIMPR (IV1, IV3)
        GC TC 10

7C      CALL RLSPSR (IV1, IV3, IACCR)
        GO TC 1C

72      CALL ASCII (IV1, IV3)
        GO TO 10

74      CALL RLESEC (IV1, IV3)
        GO TO 10

76      CALL FAIL (IV3)
        GO TO 10

78      CALL RESTOR (IV3)
        GO TO 10

8C      CALL SPAVL (IV1)
        GO TO 10

82      CALL IBDPO (IV1, IV2)
        GC TC 10
CALL SPRET (IVI)
GO TO 10
CALL LVSPQ (IVI, IAADDR)
GO TO 10
END

*IBFTC PRELIM

SUBROUTINE PRELIM

C       INITIALIZE SOME SYSTEM VARIABLES
LET ENOPD = PERCD
LET ENOSH = LEASH
LET CURSH = 1

C       INITIALIZE EACH MINIMUM TO A VERY LARGE NUMBER
DC TC 2C, FOR EACH SHOP I
LET PMT(I) = 10000.
LET TNPRT(I) = 10000.

2C
LCOP
RETURN
END

*IBFTC CLOCK

SUBROUTINE CLOCK

C       THIS ROUTINE KEEPS TRACK OF TIME, END-CF-PERIOD, AND END-OF-SHIFT.
C       THE BEGINNING OF SIMULATION, WHEREAS 'RTIME' IS THE CURRENT SIMULATED TIME.

2C
LET T = APINI (RTIME, ENOSH, ENOPD)
C       UPDATE TIME IF SOME WORK-TIME HAS ELAPSED, THAT IS, IF THIS IS A WORKING SHIFT
C       IF (SCHEDICURSH) EC (1), LET TIME = TIME + T - STIME
C       UPDATE STIME
LET STIME = T
IF (RTIME) EC (T), GC TC 100
C       UPDATE THE NO. OF PAY-DAYS (CCVT) FOR EACH PERSONNEL TYPE I
DO TC 40, FOR EACH PTYPE I
ACUMulate GVTYRT(I,CURSH) INTO CCGVT(I) SINCE TGTY(I)
LOOP
C       THERE IS AN END-OF-SHIFT AND/OR AN END-OF-PERIOD. DETERMINE WHICH COMES FIRST.
IF (ENOSH) LE (ENDP), GC TC 30
C       END-OF-PERIOD
CALL ENOPD
GO TO 20
C       END-OF-SHIFT. UPDATE CURSH AND ENOSH.
9C
LET CURSH = PCC (CURSH, NSHIFT) + 1
LET ENOSH = ENOSH + LEASH
GO TO 20
1C
RETURN
END

*IBFTC ENOPD

SUBROUTINE ENOPD

C       END OF A REPORT PERIOD.
C       COMPUTE CURPD AND CURPO.
LET S = STIME
LET CURP = S - ETIME
LET ETIME = S
LET CURPD = TIME - PTIME
LET PTIME = TIME
C       IF NO WORK-TIME HAS ELAPSED, CAN'T OUTPUT ANYTHING
IF (CURPD) EQ (0.), GC TC 100
C       OUTPUT THE REPORTS FOR THIS PERIOD
CALL CLVI
CALL CLT2
CALL CLT3
CALL CLT4
CALL CLT5
LET ENDP = ENDPD + PERCC
END

**10FTC ACTIVITY**

SUBROUTINE ACTIVITY (IACNC)

C............THIS ROUTINE IS CALLED WHEN ICC=4COO. (AT BEGINNING OF RUN)
            IF (IACNC) GR (NACTIV), CALL EPAR (4-PACTV)
            LET CUPAC = IACNC
            RETURN
            END

**10FTC EGATAC**

SUBROUTINE EGATAC (IECNC)

C............THIS ROUTINE IS CALLED WHEN ICC=4CO2. (AT BEGINNING OF RUN)
            IF (IECNC) GR (NETYPE), CALL EPAR (4-ECAT)
            RETURN
            END

**10FTC PASNEL**

SUBROUTINE PASNEL (IPERAC, ICTY, ISHIFT)

C............THIS ROUTINE IS CALLED WHEN ICC=4003. (AT BEGINNING OF RUN)
            IF (IPERAC) GR (NATYPE), CALL EPAR (4-PASN)
            LET GTSET(IPERAC) = GTSET(IPERAC) + ICTY
            LET GTVIP(IPERAC, ISHIFT) = ICTY
            RETURN
            END

**10FTC EQUIP**

SUBROUTINE EQUIP (IECNC, ICTY)

C............THIS ROUTINE IS CALLED WHEN ICC=4004. (AT BEGINNING OF RUN)
            IF (IECNC) GR (NETYPE), CALL EPAR (4-EQUI)
            RETURN
            END

**10FTC SPARES**

SUBROUTINE SPARES (ISPNC, ICTY)

C............THIS ROUTINE IS CALLED WHEN ICC=4005. (AT BEGINNING OF RUN)
            IF (ISPNC) GR (NSTYPE), CALL EPAR (4-SPRN)
            LET GTSIP(ISPNC) = ICTY
            RETURN
            END

**10FTC ARRIV**

SUBROUTINE ARRIV (IPERAC, IC)

C............THIS ROUTINE IS CALLED WHEN ICC=4200.
            IF A REP HAS ENTERED THE SYSTEM, CREATE A TEMPOARY RECORD FOR
            IT, PILE IT INTO THE APPROPRIATE SHOP, AND UPDATE THE
            STATISTICS FOR THIS SHOP.
            CREATE REP
            LET GTIME(REP) = RTIME
            LET IDN(REP) = ID
            LET ISHCP = SHCP(REP)
            LET NEURIP = RIP(ISHCP) + 1
            LET RIP(ISHCP) = NEURIP
            LET RIN(ISHCP) = RIN(ISHCP) + 1
            LET PAIR(ISHCP) = PAIR(NEURIP, PAIR(ISHCP))
LET FRIP = NEWRIP
LET RIPS(ISHSCP) = RIPS(ISHSCP) + FRIP
LET RIPSC(ISHSCP) = RIPSC(ISHSCP) + FRIP
FILE *EP IN LCAC(ISHSCP)
RETURN
*IBFTC DEPART
SUBROUTINE DEPART (IREPAC, IC)
C ........ THIS ROUTINE IS CALLED WHEN ICC=1401.
C ........ A REP HAS LEFT THE SYSTEM, REMOVE AND DESTROY IT.
LET ISHSCP = SHP pac(IREPAC)
FIND FIRST REP, FOR EACH REP IN LCAC(ISHSCP), WITH
*  (IDNC(REP))(IC(IDC), WHERE REP, IF NICE, CALL ERROR (4HCEPA)
REMOVE REP FRP LCAC(ISHSCP)
LET NEHRIP = RIP(ISHSCP) - 1
LET RIP(ISHSCP) = NEHRIP
LET ALCUT(ISHSCP) = ALCUT(ISHSCP) + 1
LET PIRN(ISHSCP) = PIRN (NEHRIP, PIRN(ISHSCP))
LET FRIP = NEHRIP
LET RIPS(ISHSCP) = RIPS(ISHSCP) + FRIP
LET RIPSC(ISHSCP) = RIPSC(ISHSCP) + FRIP
LET REPTIP = ATPE - BTPE(REP)
LET PXRT(ISHSCP) = PXRT (REPTIP, PXRT(ISHSCP))
LET PRTF(ISHSCP) = PRTF (REPTIP, PRTF(ISHSCP))
LET RTS(ISHSCP) = RTS(ISHSCP) + REPTIP
LET RTSC(ISHSCP) = RTSC(ISHSCP) + REPTIP
DESTROY REP
RETURN
END
*IBFTC NEXTAC
SUBROUTINE NEXTAC (IREPAC, IACNC, ICREP)
C ........ THIS ROUTINE IS CALLED WHEN ICC=4400.
C ........ A REP IS SUBMITTED (OR SUBMITTED) TO THIS ACTIVITY.
LET CLAC = IACNC
C ........ DO NOTHING IF ACTIVITY = 0 (RECEIVING)
   IF (IACNC) EC (0), GC TO 50
   LET CLREP = ICREP
C ........ FIND THE REP BY SEARCHING THE LCAC OF THE APPROPRIATE SHOP.
   FIND FIRST REP, FOR EACH REP IN LCAC(ISHSCP)(IREPAC), WITH
*  (IDNC(REP))(EC (ICREP)), WHERE REP, IF NONE, CALL
   ERROR (4MWE)
   LET K = CPFLAG(REP)
C ........ IF CPFLAG EQUALS THE ACT. NC. THIS REP IS ALREADY IN THE
C ........ CLEO FOR THIS ACTIVITY, SC TO ACTIIXE.
   IF (K) EC (IACNC), GC TO 50
C ........ IF REP IS ALREADY IN SOME CTPE QUEUE, THIS IS AN ERROR.
   IF (K) RE (0), CALL E*RC (4MWEZ)
C ........ IF CPFLAG = 0, FILE IT INTO THE QUEUE FOR THIS ACTIVITY.
   LET CPFLAG(REP) = IACNC
   ACC CSZAC(IACNC) IATC CCZAC(IACNC) SIAE TOSZAC(IACNC), ADD 1.
   LET PXSCA(IACNC) = PXSCA (CSZAC(IACNC), MXSCA(IACNC))
   LET CTPE(REP) = CTPE
   FILE REP IN ACTC(IACNC)
RETURN
END
*IBFTC READY
SUBROUTINE READY (IACNC)
C ........ THIS ROUTINE IS CALLED WHEN ICC=4400.
C ........ THIS REP IS READY TO BE WORKED ON, REMOVE IT FROM CLEO FOR
C ........ THIS ACTIVITY, AND TAKE STATISTICS.
ACC QSZ(A(IACNC)) INTO QSZ(A(IACNC)) SINCE QSZ(A(IACNO), ACC = 1.
IF (QSZ(A(IACNC)) LE (-1.)), CALL ERROR (4MREA3)
FIND FIRST, FOR EACH REP IN AC1C(IACNO), WITH (IACNO(REP)) EQ
(CUREP), WHERE REP, IF NCNE, CALL ERROR (4MREA2)
REMOVE REP FROM AC1C(IACNC)
LET PMCSA(IACNC) = APIA1 (CSZA(IACNC), WNZSA(IACNO))
LET TIACS(IACNC) = TIACS(IACNC) + T1PE - QT1ME(REP)
LET ACUTY(IACNC) = ACUTY(IACNC) + 1
LET QFLAG(REP) = 0
RETURN
END

*IBFIC ASINPR
SUBROUTINE ASINPR (IPIRNC, ICTY)
C .......... THIS ROUTINE IS CALLED WHEN ICC = 4460.
C .......... ASSIGN PERSONNEL TO AN ACTIVITY.
LET G = ICTY
LET R = RT1PE
LET IACNC = CURAC
IF (IACNC) EQ (0), CALL ERRCP (4MASPR)
C .......... IF THIS IS THE FIRST TIME (DURING THIS REPORT PERIOD) THAT
C .......... PERSONNEL OF THIS TYPE HAVE BEEN ASSIGNED TO THIS
C .......... ACTIVITY, CREATE A NEW ENTRY AND FILE IT INTO LIST FOR
C .......... THIS PERSONNEL IN. IN ANY CASE, TAKE STATISTICS.
FIND FIRST, FOR EACH ENTRY OF LIST(IPIRNC), WITH
(A1CNC(ENTRY)) EQ (IACNC), WHERE ENTRY, IF NONE, GO TO 2C
LET b = WK1NG(ENTRY)
LE: CW1NG(ENTRY) = CW1NG(ENTRY) + W * (I-RWK1NG(ENTRY))
LET BK1NG(ENTRY) = W + C
GO TO 50
2C CREATE ENTRY
LET ACNC(ENTRY) = IACNC
LET WNK1NG(ENTRY) = C
FILE ENTRY IN LIST(IPIRNC)
5C LET TW1NG(ENTRY) = R
RETURN
END

*IBFIC RE3SPA
SUBROUTINE RE3SPA (IPIRNC, ICTY, IACNC)
C .......... THIS ROUTINE IS CALLED WHEN ICC = 4590.
C .......... PERSONNEL HAVE BEEN RELEASED FROM THIS ACTIVITY. TAKE
C .......... STATISTICS.
FIND FIRST, FOR EACH ENTRY OF LIST(IPIRNC), WITH
(A1CNC(ENTRY)) EQ (IACNC), WHERE ENTRY, IF NONE, CALL
ERROR (4MHALPA)
LET b = WK1NG(ENTRY)
LET R = RT1PE
LET CW1NG(ENTRY) = CW1NG(ENTRY) + W * (I-RWK1NG(ENTRY))
LET TW1NG(ENTRY) = R
LET BK1NG(ENTRY) = W - FLCAT(ICTY)
RETURN
END

*IBFIC ASINEQ
SUBROUTINE ASINEQ (IEQNC, ICTY)
C .......... THIS ROUTINE IS CALLED WHEN ICC = 4470.
C .......... ASSIGN EQUIPMENT.
ACCMULATE INUSE(IEGNO) INTO CINUS(IEGNO) SINCE
TINUS(IEGNO), ADD FLCAT(ICTY)
RETURN
END

*IBFIC ALESEQ
SLBRCLTIME RLESEC (IEGNC, IGTY)
C ........THIS RCLTINE IS CALLED WHEN ICC=4560.
C ........RELEASE EQUIPMENT.
    ACCUPLICATE INUSE(IEGCAC) INTO (INUSE(IEGNO)) SINCE
    TINUS(IEGCAC), ACC -FLCAT(IGTY)
    IF (INLS(IEGCAC)) LE (-1.), CALL ERROR (4HRL2)
    RETURN
END
*IBFTC FAIL
SLBRCLTIME FAIL (IEGNC)
C ........THIS RCLTINE IS CALLED WHEN ICC=4600.
C ........EQUIPMENT FAILURE.
    LET NFAIL(IEGNC) = NFAIL(IEGNC) + 1
    LET T = RTIME
    LET COCWN(IEGCAC) = CCCHA(IEGCAC) + CCWN(IEGNC) *
    (T - TCCWN(IEGNO))
    LET TDCHN(IEGNC) = T
    LET DCHW(IEGCAC) = DCHW(IEGCAC) + 1.
    RETURN
END
*IBFTC RESTOR
SLBRCLTIME RESTOR (IEGNC)
C ........THIS RCLTINE IS CALLED WHEN ICC=4700.
C ........EQUIPMENT RESTORED.
    LET T = RTIME
    LET COCWN(IEGCAC) = CCCHA(IEGCAC) + CCWN(IEGNC) *
    (T - TCCWN(IEGNC))
    LET TDCHN(IEGNC) = T
    LET DCHW(IEGCAC) = DCHW(IEGCAC) - 1.
    IF (DCHW(IEGCAC)) LE (-1.), CALL ERROR (4HRL3)
    RETURN
END
*IBFTC SPAVL
SLBRCLTIME SPAVL (ISPNC)
C ........THIS RCLTINE IS CALLED WHEN ICC=4490.
C ........THERE IS A DEPANC FOR A SPARE, THE SPARE IS AVAILABLE, SO
C ........THE DEPANC IS IMMEDIATELY FILLED.
    LET DMAND(ISPNC) = DMANC(ISPNC) + 1
    LET FILL(ISPNC) = FILL(ISPNC) + 1
    LET XCSP(ISPNC) = APAX1 (CSP(ISPN0)+1., MXCSP(ISPNC))
    RETURN
END
*IBFTC INSPQ
SLBRCLTIME INSPQ (ISPNC, IREPNC)
C ........THIS RCLTINE IS CALLED WHEN ICC=4455.
C ........THERE IS A DEPANC FOR AN UNAVAILABLE SPARE PART.
C ........FIND FIRST, FOR EACH REP IN LCAC(ISPNC),(IREPNC), WITH
C ........  (IDNC(REP)) EC (CUREP), WHERE REP, IF NONE, CALL
C ........ERROR (4HINS3) LET S = SFLAG(REP)
C ........IF SFLAG = SPARE PART NO., THIS REP IS ALREADY IN QUEUE FOR
C ........  THIS PART, SC EC NOTHING.
C ........  IF (S) EC (ISPNC), GC TQ 50
C ........IF REP IS ALREADY IN QUEUE FOR A DIFFERENT PART, CALL ERROR.
C ........  IF (S) NE (0), CALL ERROR (4HINS2)
C ........IF SFLAG = 0, PUT IT IN QUEUE FOR THIS SPARE PART NO.
C ........LET SFLAG(REP) = ISPAC
C .........LET DMAND(ISPNC) = DMANC(ISPNC) + 1
C .........ACC CSP(ISPNC) INTO CCSF(ISPNC) SINCE TCSF(ISPNC), ADC 1.
C .........LET MXCSP(ISPNC) = APAX1 (CSP(ISPN0), MXCSP(ISPNC))
LET SPTM(REP) = TIME
RETURN
END

*IBFCT LSPT
SUBROUTINE LSPT (ISPNCE, ICREP)
C ........THIS ROUTINE IS CALLED WHEN IIC=4001.
C ........IF REP IS WAITING FOR THIS PART, RC NOTHING.
C ........TAKE THIS REP OUT OF THE QUEUE FOR THIS SPARE PART.
C
LET ISPNC = CSPNC
LET LNL(ISPNC) = LNL(ISPNO) + 1
ACC C5P(ISPNC) INT C5P(ISPNC) SINCE ISP(ISPNO), ADD -1.
IF (C5P(ISPNC)) LE (-1.), CALL ERROR (4HLV5)
LET RMQSP(ISPNC) = APIR (C5P(ISPNC), RMQSP(ISPNO))
FIND FIRST, FOR EACH REP IN LCAC(ISPNC,ICREP), WITH
* (ICDRC(REP)) GC (ICREP), WHERE REP, IF NONE, CALL
* ERROR (4HLV5)
LET TISC(S(ISPNO)) = TISC(S(ISPNC)) + TIME - SPTM(REP)
LET SFLAG(REP) = 0
RETURN
END

*IBFCT OUT
SUBROUTINE OUT
C ........GENERATE A "TABLE 1" REPORT FOR EACH SHCP.
DO TO 100, FOR EACH SHCP I
LET INR = INR(I)
LET INCL = INCL(I)
LET F1C = IFCL(I)
LET FRTS = IFRT(I)
LET FRTSC = IFRTSC(I)
LET IRIP = IRIP(I)
LET FRIP = FRIP(I)
LET [INR = [INR(I) + INR
LET TRIM(I) = TRIM
LET ITPCUT = ITPCUT + ITPCUT
LET TPCUT(I) = ITPCUT
LET TOTAI = TRIM + ITPCUT + 1
CALL SITDEV (TOTAL, RIPS(I), RIPS(I), TAVN, STCVN)
IF (TPCUT(I) EQ 0), LET PART(I) = 0.
CALL SITDEV (FRCUT, FRTS, FRTSC, TAVT, STCVT)
LET FRTS = FRTS(I) + FRTS
LET FRTS(I) = FRTS
LET FRTSC = FRTSC(I) + FRTSC
LET FRTSC(I) = FRTSC
CALL SITDEV (FRTOUT, FRTS, FRTSC, TAVT, STCVT)
IF (TPCUT(I) EQ 0), GC TC 60
LET TPXHAR(I) = ARAP (PART(I), TPXHAR(I))
LET TPXHAR = APIAR (PART(I), TPXHAR(I))
LET TPXHAR(I) = TPXHAR
GO TC 60
LET TPXHAR = 0.
CALL TAB1 (I, AVN, STDVR, AVT, STOVT, TAVT, FTMNRT, TSTDVT)

RESET VARIABLES FOR NEXT REPORT PERIOD.

LET RIN(I) = 0
LET ROLT(I) = 0
LET PAXR(I) = IFIP
LET PINR(I) = IFIP
LET RIPS(I) = FRIP
LET RIPSQ(I) = FRIP * FR.P
LET RTS(I) = 0.
LET RTSG(I) = 0.
LET RMT(I) = 0.
LET RNPRT(I) = 10000.

RETURN

END

SUBROUTINE STDEV (TOTAL, SUP, SUPSG, AVG, STCV)

RDTIME TO COMPLETE A PERCENT STANDARD DEVIATION.

LET AVG = SUP / TCTAL
LET STDV = SQRT((APAXI(SUPSG/TCTAL - AVG*AVG, 0.1))

LET AVG = 0.
LET STDV = 0.

RETURN

END

REPORT TAB1 (I, AVN, STDVR, AVT, STOVT, TAVT, FTMNRT, TSTDVT)

SYSTEM ARRIVALS, CEP

AND REPAIR

REPAIRABLES ENTERING SHOP THIS PE

SERVICEABLES DEPARTING SHOP THIS

REPAIRABLES IN PROCESS 100.0%

REPAIR TIME 100.0%

THIS PERIOD 100.0%

TC DATE 100.0%

END

PRC 1

ARTURES, IN-PROCESS,

CYCLE TIMES

DATA FOR PERIOD ENDING DAY 100.000

ENDPD

RIDD 100.00, TC DATE 100.000

RIN(I) TRIN(I)

PERIOD 100.00, TC DATE 100.000

RCLT(I) TRCLT(I)
/*IBFTC CLT2
SUBRCLTIME CLT2
C ..........GENERATE TABLE 2.
  LET C = CURPD
  DO TC 5C, FOR EACH ACTIV I
  ACCLPLATE QSZ(A(I)) INTO CCSZA(I) SINCE TGSZA(I)
  LET AVGSA(A(I)) = CCSZA(I) / C
  IF (AQCL(T(I))) EC (0), GC TC 50
  LET AQTIA(A(I)) = CECHR(TIACS(I)) / FLCAT(AGOUT(I))
5C
  LCOP
  CALL TAB2
  DO TC 10C, FOR EACH ACTIV I
  LET CCSZA(I) = 0.
  LET TIACS(I) = 0.
  LET FGSZA = QSZ(A(I))
  LET MXCSA(I) = FGSZA
  LET AVCSA(I) = FGSZA
  LET AQCLIT(I) = 0
10C
  LCOP
  RETURN
END
*/

/*IBFTC TAB2
REPORT TAB2
X ACTIVITY QUE
X FOR PERIOD J
X NO. OF REPS
X ACT.
X NO. AVG MAX
X
X FOR EACH ACTIV I, WITH (MXCSA(I)) ER (0.)
END

LEING FACTCARS
LST COPPLEED
IN QUELE
AVG QLECE TIEPE
PEN (X LCRK-HCURS)
0. 0.00 0.00
1) PXQSA(I) AVIAC(I)
*/

/*IBFTC CLT3
SUBRCLTIME CLT3
C ..........GENERATE TABLE 3.
C ..........BEGIN BY WRITING THE HEADING.
CALL THTHED
LET S = STIME
DO TC 100, FOR EACH PTYPE I, WITH (QTVS(I)) ER (C)
LET SLP = 0.

ICE DC TC 2C, FCR EACH ENTRY CF LIST(I)

C .......BRING 'CWRKNGENTRY)' UP TO CATE BEFORE ADDING IT INTO SUM.

LET C = CWRKNGENTRY + WKNGENTRY + (S-TWKNGENTRY)

LET CWRKNGENTRY = C

LET SLP = SLP + C

2C REPEAT 10

LET UTIL = SUM / CQTY(I)

CALL TB3LIN (I, UTIL)

LET CQTY(I) = 0.

C .......EMPTY CLT EACH 'LIST' SET.

3C DC TC 50, FCR EACH ENTRY CF LIST(I)

REMOVE ENTRY FRC P LIST(I)

DESTROY ENTRY

5C REPEAT 30

1CC LCOP

RETURN

END

*IBFTC TB3HED

REPORT TB3HED

PERSONNEL

TCR AT ACTIVITY

PERS UTIL PAN- PAN- MAN- MAN-
TYPE QTV FACT NC HOURS AC HOURS NC HOURS NO HOURS

EAC

UTILIZATION

PAN- PAN- PAN- PAN- MAN- MAN-
NO HOURS NC HOURS AC HOURS AC HOURS NO HOURS

EAC

*IBFTC TB3LIN

REPORT TB3LIN (I, UTIL)

000 00 0.00

I QTVS(I) UTIL

9 FOR EACH ENTRY CF LIST(I), WITH (FLIST(I)) NE (0)

00 00 00 00 00 00 00 00 00 00 00

9(ACHC(ENTRY),CEC-R(CWKNGENTRY))

END

*IBFTC OLTA

SLORCLTIME OLTA

C .......GENERATE TABLE 4.

CALL TB4HED

LET CLR = CURP

DC TC 50, FCR EACH ETYPE I, WITH (CTYE(I)) GE (0)

LET TOTAL = CUR * FLOAT(CTYE(I))

LET C = CURUS(I)

LET UTIL = C / TOTAL

LET FIDLE = TOTAL - C - CECW(I)

CALL TB4LIN (I, FIDLE, UTIL)

LET NFAL((I) = 0
LET C0CWA(I) = 0.
LET CINUS(I) = 0.
5C
LC0P
RETURN
END

*IBFIC TB4MED
REPORT TB4MED

EQUIPMENT U

EQUIPMENT TYPE QUANTITY IN USE TIME ICLE DOWN N

TIME TIME FA

END

UTILIZATION

0. OF UTILIZATION ACTIVITY NCS.
ILLRES FACTOR WHERE USED

END

*IBFIC TB4LIN
REPORT TB4LIN (I, FIDLE, UTIL)

12 FOR EACH ITEM OF SET(I)

END

*IBFIC TL4S

SUBROUTINE TL4S

CALL TABS

DO TC 10C, FOR EACH SPTYP I

LET C0CSP(I) = 0.
LET CINUS(I) = 0.
LET FQSP = QSP(I)
LET PNCSP(I) = FQSP
LET PNQSP(I) = FQSP
LET QCSP(I) = 0.
LET CINUS(I) = 0.
LET C0CWA(I) = 0.

5C
LC0P
RETURN
END

*IBFIC TABS

REPORT TABS

QUEUEING FACTORS BY REPARABLE QUEUE LENGTHS

PERCENT TYPE QUANTITY DEPRACS

THIS PERIOD
FOR EACH SPTYP I, WITH (GTYSPI(I)) GR (0)

COMPONENT SPARES TYPE
AND TIMES BY COPPENTS
NO. OF REPS IN QUEUE

END

AVG QUEUE TIME

END

*IBFTC ERROR
SUBROUTINE ERROR (NAME)
CALL ERREPT (NAME)
CALL SNAP (GGERER)
STOP
END

*IBFTC ERREPT
REPORT ERREPT (NAME)

JOB TERMINATED AT TIME **0,000** BECAUSE OF ERROR IN SUBROUTINE **00**

STOP (ABBRE

END

A#'
HE
VIATION)

END

*IBFTC SNAP
SUBROUTINE SNAP (LABEL)
CALL SNPI (LABEL)
CALL SNP2 (I), FOR EACH SHTC I
RETURN
END

*IBFTC SNPI
REPORT SNPI (LABEL)

RTIME STIME PTIME ENCPD CL

0,00000 0,00000 0,00000

ENCPD CL

SHOPS - RIN RCly TRIM TRCUT PAHR PIRP RIP RIPS RIPSQ AT

I RIN(I) ROUT(I) TRIM(I) TRCUT(I) PAHR(I) PIRP(I) RIP(I) RIPS(I)

FOR EACH SHTC I

ACTIVITIES ---

CSZA CCSZA TGSZA

I CSZA(I) CCSZA(I) TGSZA(I)

FOR EACH ACTIV I, WITH (MXQSP(I)) NE QO).

EQUIPMENT TYPES ---

FAIL IMUSE CIMUS TIMUS

I FAIL(I) IMUSE(I) CIMUS(I) TIMUS(I)

FOR EACH ETYPES I, WITH (GSTE(I)) NE QO)

SPARE PART TYPES ---

BRAND FILL QSP CSQSP

I BRAND(I) FILL(I) QSP(I) CSQSP
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>STOP</td>
</tr>
<tr>
<td>12</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>4</td>
<td>PTYPE</td>
</tr>
<tr>
<td>3</td>
<td>ETYP</td>
</tr>
<tr>
<td>5</td>
<td>SPTYP</td>
</tr>
<tr>
<td>4</td>
<td>RPTYP</td>
</tr>
<tr>
<td>4(12)</td>
<td>SHIFT</td>
</tr>
<tr>
<td>(41.1)</td>
<td>LENS</td>
</tr>
<tr>
<td>7.C</td>
<td>PMROC</td>
</tr>
<tr>
<td>1</td>
<td>BASE</td>
</tr>
</tbody>
</table>

Note: The table entries are in the form of codes, which might require specific context or a reference to understand their full meaning.
Program 10

NETS PROGRAM
X. NRTS PROGRAM

The NRTS (not reparable this station) data display shows the reparable shipped off base for repair. It displays the pipeline time distribution for the reparable. This program is used primarily as an input to the Depot Transportation Simulator when the simulators are operated separately.

An example of the output display is shown in Fig. 24. The reparable ID number is listed in column 1, followed by the quantity that was shipped to the depot during the period. The minimum, average, and maximum pipeline times are next displayed, followed by the standard deviation of the distribution. As with other reports, the reporting period and base number are initialized values.

INITIALIZATION

The NRTS Program requires the initialization of 16 variables. Only three, however, require values. The variable description and initialization table (Table 7) contains the information required to initialize the report program. An example data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

This is a supplement to the Bench Repair Analysis Program, having a very similar logic (only much simpler), and using the same input tape made up of 12-word label records. The output is a report (every period) on the passage of reparable through the NRTS cycle.

Only those records with IDD = 6000 or 4900 are of concern to this program; all others are skipped. As before, all records with base number not equal to the system attribute BASE are skipped.

Because no ACCUMULATE statements are used, this program does not use the automatically defined system variable TIME, nor does it need a CLOCK subroutine.

An ERROR routine is present, and is called in case of an error condition from several places in the program.
### NRTS DATA

**BASE 1 PERIOD ENDING DAY 30.000**

<table>
<thead>
<tr>
<th>REP NO</th>
<th>NRTS QUANTITY</th>
<th>NRTS DELAY TIME</th>
<th>MIN</th>
<th>AVG</th>
<th>MAX</th>
<th>STD DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>10.566</td>
<td>13.658</td>
<td>17.940</td>
<td>3.126</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5.018</td>
<td>9.181</td>
<td>14.283</td>
<td>2.578</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>2.086</td>
<td>2.529</td>
<td>3.669</td>
<td>0.471</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5.967</td>
<td>8.536</td>
<td>11.769</td>
<td>2.224</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3.136</td>
<td>3.136</td>
<td>3.136</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 24**

### Table 7

**VARIABLE DESCRIPTION AND INITIALIZATION:**

<table>
<thead>
<tr>
<th>NRTS</th>
<th>Mode</th>
<th>Initiation No.</th>
<th>Array No. of Attributes</th>
<th>Array No. of Variables To Be Used In Eq. 2 (Col. 1)</th>
<th>Linear Description of Variables To Be Initialized</th>
<th>Parameter Name</th>
<th>Var. Name</th>
<th>Entry Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>None</td>
<td>1</td>
<td>LR5</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7777</td>
<td>2</td>
<td>LR7</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>None</td>
<td>3</td>
<td>LR9</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>None</td>
<td>4</td>
<td>LR11</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>None</td>
<td>5</td>
<td>LR14</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>None</td>
<td>6</td>
<td>LR15</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>None</td>
<td>7</td>
<td>LR16</td>
<td>1</td>
</tr>
<tr>
<td>1-6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>None</td>
<td>8</td>
<td>LR17</td>
<td>1</td>
</tr>
</tbody>
</table>

---

*Note: Table 7 details variable initialization for the NRTS model.*
PERMANENT VARIABLES

TIM - current simulated time, as read from the current label record.
PTIME - the time at the beginning of the current report period.
ENDPD - the time at which the current report period will end.
CURPD - the length of the report period that has just ended.
BASE - the base number for this run.
PEROD - the length of a report period.
RPTYP - reparable type; a permanent entity with the following attributes:
  FLOAD - first of the set called LOAD.
  LLOAD - last of the set called LOAD.
  TOTAL - the total number of reparables of this type completing the
           NRTS cycle in this period.
  SUM - the sum of their NRTS times.
  SUMSQ - the sum-square of their NRTS times.
  AVG - the mean NRTS time for this type of reparable during this
        period.
  STDV - the standard deviation of NRTS time for this type of
         reparable.
  MAX - the maximum NRTS time for this type of reparable during
        this period.

TEMPORARY VARIABLES

REP - reparable part; a temporary entity with the following attributes:
  IDNO - I.D. number; a number obtained from the label record that
         uniquely identifies this reparable.
  BTIME - the time at which this reparable entered the NRTS cycle.
  PLOAD - predecessor in LOAD.
  SLOAD - successor in LOAD.

SETS

LOAD - a singly-subscripted set, ranked on BTIME.
  owner: RPTYP
  member: REP
*T REP 4
+ T IDNO 1 I
+ T BTIME 2 F
+ T PLOAD 3 I
+ T SLOAD 4 I
+ TIM F
+ 2PTIME F
+ 3ENDPD F
+ 4CURPD F
+ 5BASE IC
+ 6PEROD FC
+ 7RPTYP E
+ 8LOAD 1 I
+ 9LOAD 1 I
+ 10TOTAL 1 I
+ 11SUM 1 F
+ 12SUMSQ 1 F
+ 13AVG 1 F
+ 14STDV 1 F
+ 15MAX 1 F
+ 16MIN 1 F

*IBFTC MAIN

MAIN ROUTINE

C......INITIALIZE ENDPD.
LET ENDPD = PEROD
C......INITIALIZE EACH MINIMUM TO A HIGH NUMBER.
LET MIN(I) = 10000, FOR EACH RPTYP I
C......READ A LABEL RECORD.
X 10 READ (9) K, IDD, K, K, INBASE, NO, K, K, IADDR, INDIC, T
LET TIM = T
C......IF THERE IS A DETAIL RECORD, SKIP IT.
X IF (INDIC.EQ.1) READ (9) JUNK
IF (IDD) NE (3), GO TO 30
C......END OF SIMULATION, (IDD=3,) TERMINATE AFTER PRINTING THE
C......LAST REPORT.
LET ENDPD = TIM
CALL ENDPDRD
STOP
C......IGNORE THIS RECORD UNLESS INBASE = RASE.
30 IF (INBASE) NE (RASE), GO TO 10
C......TEST FOR END OF PERIOD.
IF (TIM) OR (ENDPD), CALL ENDPDRD
C......IF THIS RECORD IS RELEVANT (IDD = 6000 OR 4900), CALL A
C......SUBROUTINE TO PROCESS IT. OTHERWISE SKIP IT.
IF (IDD) NE (6000), GO TO 50
CALL START (NOI, IADDR)
GO TO 10
50 IF (IDD) NE (4900), GO TO 10
CALL FINISH (NOI, IADDR)
GO TO 10
END

*IBFTC ENDPDRD
SUBROUTINE ENDPDRD
C ........THIS ROUTINE IS CALLED AT THE END OF EACH PERIOD,
LE T = ENDPD
LET CURPD = T - PTIME
LET PTIME = T
C ........OUTPUT NOTHING IF NO TIME HAS ELAPSED SINCE END OF PREVIOUS
C ........ PERIOD.
   IF (CURPD) EQ (0.), GO TO 100
   DO TO 30, FOR EACH RFTYP I
   LET FTOT = TOTAL(I)
   CALL STDDEV (FTOT, SUM(I), SUMSQ(I), *AVG(I), *STOV(I))
   IF (MIN(I)) EQ (10000.), LET MIN(I) = 0.
30  LOOP
   CALL REPORT
C ........RESET ALL RUNNING TOTALS.
   DO TO 50, FOR EACH RPTYP I
   LET MIN(I) = 10000.
   LET MAX(I) = 0.
   LET SUM(I) = 0.
   LET SUMSQ(I) = 0.
   LET TOTAL(I) = 0
50  LOOP
   LET ENDPD = ENDPD + PERIOD
100 RETURN
END

*ISFTC START

SUBROUTINE STAKT (NO, ID)
C ........THIS ROUTINE IS CALLED FOR ID = 6000.
C ........A REP HAS JUST ENTERED THE NRTS CYCLE.
   IF (NO) GR (NRPTYP), CALL ERROR (4HSTAR)
   CREATE REP
   LET BTIME(REP) = TIN
   LET IDNO(REP) = ID
   FILE REP IN LOAD(NO)
   RETURN
END

*ISFTC FINISH

SUBROUTINE FINISH (NO, ID)
C ........THIS ROUTINE IS CALLED FOR ID = 4900.
C ........A REP HAS JUST LEFT THE NRTS CYCLE.
   IF (NO) GR (NRPTYP), CALL ERROR (4HFINI)
   FIND THE REP, TAKE STATISTICS, AND DESTROY IT.
   FIND FIRST, FOR EACH REP OF LOAD(NO), WITH (IDNO(REP)) EQ
   *(ID), WHERE REP, IF NONE, CALL ERROR (4HFFIN2)
   REMOVE REP FROM LOAD(NO)
   LET T = TIM - BTIME(REP)
   LET MAX(NO) = MAXI(T, MAX(NO))
   LET MIN(NO) = MINI(T, MIN(NO))
   LET SUM(NO) = SUM(NO) + T
   LET SUMSQ(NO) = SUMSQ(NO) + T*T
   LET TOTAL(NO) = TOTAL(NO) + 1
   DESTROY REP
   RETURN
END

*ISFTC STDDEV
SUBROUTINE STDDEV (TOT, S, SC, AVE, STD)

C ROUTINE TO TAKE A MEAN AND STANDARD DEVIATION.
IF (TOT) LE (0.), GO TO 50
LET AVE = S / TOT
LET STD = SQRT(AMAX1(SQRT(TOT - AVE*AVE, 0.)))
GO TO 100

50 LET AVE = 0.
LET STD = 0.
100 RETURN
END

*IBFTC REPORT
REPORT REPORT

  X  NRTS  X  PASS
  X  NO   NRTS  MIN
  X  *    **   *,**
  X  I    TOTAL(I)  MIN(I)
  
FOR EACH RPTYP I, WITH (TOTAL(I)) NE (0)
END

*IPFTC REPORT
REPORT ERREPT

  X  SUBROUTINE ERROR (NAME)
  X  CALL ERREPT (NAME)
  X  STOP
END

*IPFTC ERRREPT
REPORT ERREPT (NAME)

  X  JOB TERMINATED AT TIME  **,** BECAUSE OF ERROR IN SUBROUTINE ***
  X  NA
  X  (APRRE
  X  END)

END
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
Program 11

DEPOT TRANSPORTATION CAPABILITY
XI. DEPOT TRANSPORTATION CAPABILITY

The Depot Transportation Capability Output Program is a two-part program displaying the cargo delivered to each base(s) and the utilization of each transport vehicle.

CARGO

The Cargo Report (see Fig. 25) displays the quantity of cargo moved throughout the simulated period. The quantity is specified in terms of weight, volume, and units for each type of cargo. Note the separate specification of both the quantity of cargo that is loaded aboard some carrier (TP) and the quantity of cargo delivered (TD) by a carrier. The same distinction is made for base deliveries (column 4) and for the depot (column 5). Column 6 specifies the quantity of each cargo type that is in process (in transit) as of the report time.

INITIALIZATION

The cargo output program requires the user to initialize eighteen variables. Table 8 and its accompanying text will facilitate use of this program, and help the user understand the conceptual basis for the resultant report.

OUTPUT PROGRAM

The input to the cargo output program is the tape generated by the Depot Transportation Simulation Program. The tape is read from logical unit No. 9.

This input tape consists of twelve-word records in the following form:

Word 1 - irrelevant.
Word 2 - a four-digit number identifying the occurrence represented by this record.
Words 3 through 5 - irrelevant.
Word 6 - the base number.
CARGO REPORT FOR BASE 2 PERIOD ENDING 4.00

<table>
<thead>
<tr>
<th>CARGO TYPE</th>
<th>TOPS DELIVERED</th>
<th>UNITS DELIVERED</th>
<th>VOLUME DELIVERED</th>
<th>BASE UNITS</th>
<th>DEPOT UNITS</th>
<th>IN PROCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP</td>
<td>TD</td>
<td>TP</td>
<td>TD</td>
<td>TP</td>
<td>TD</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 25
Table 8

VARIABLE DESCRIPTION AND INITIALIZATION:
CARGO

<table>
<thead>
<tr>
<th>Array Number</th>
<th>Number of Sub-arrays</th>
<th>Number of Zeros Value</th>
<th>Array Number to Be Expanded to Fig. 3 Col.</th>
<th>Description of Variables to Be Initialized</th>
<th>Parameter System Variable Name</th>
<th>Entity Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>7</td>
<td>3-27</td>
<td>Arrays Default</td>
<td>CRAY 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
<td>Den of boxes contained in the simulation.</td>
<td>BASE 0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td></td>
<td>Den of reportable contained in the simulation.</td>
<td>BPROC 0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
<td>T</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td></td>
<td>T</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td></td>
<td>T</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td></td>
<td>T</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td></td>
<td>T</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td></td>
<td>Specify the volume of each reportable contained in the simulation.</td>
<td>VOLUME A</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>First box number to be printed by this report.</td>
<td>FIRST A</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>1</td>
<td></td>
<td>Last box number to be printed by this report.</td>
<td>LAST A</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1</td>
<td></td>
<td>Instead sequence of these numbers to be printed</td>
<td>INSEQUENCE A</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Words 7 through 10 - the content of these words varies with the individual values of Word 2.

Word 11 - irrelevant.

Word 12 - current simulated time.

When a record is read from tape 9, the value of CNT is compared with current simulated time (Word 12); if simulated time equals or exceeds the value of CNT, steps are taken to issue a report. Thus, CNT is used to control the report interval. If CNT exceeds the value of simulated time, the current record's information is processed as follows.

Word 6 is checked to determine if the record contains information about a relevant base. There is provision in the cargo output program for reporting on all, or selected, bases, in any order desired by the user.

For all relevant bases, Word 2 is compared with internal four-digit codes in order to select only records that are "interesting" to the program. At this point, the program transfers control to segments unique to each code, which will in turn process the individual data records.

When Word ? is found to equal 5000, the end of simulation has been reached. A final report is issued, and the cargo output program terminates.

PERMANENT VARIABLES

The following list is complete except for attributes denoting first- or last-of-set.

CNT - the report interval, specified by the user.

BASE - a permanent entity representing the number of bases in the simulation.

NRPS - the total number of unique reparables involved in the simulation.

TTON - sum of the tonnage arriving at each base.

TVOL - sum of the volume arriving at each base.

TUNT - sum of the number of units arriving at each base.

TDPO - sum of the number of base units arriving at each base (base unit = reparable that originates at a base rather than a depot).

TBAS - sum of the number of depot units arriving at each base.

LMT - a debugging aid.

WNT - a list of weights for individual reparable types.

VOL - a list of volumes for individual reparable types.
TEMPORARY VARIABLES

REP - a temporary entity with the following attributes:
  BFLG - a system flag indicating that the reparable is a base unit.
  SYSTA - the reparable type number (as opposed to serial number, which is a machine address).
  WATE - weight of the particular reparable.
  VOLU - volume of the particular reparable.

RPORT - a temporary entity with the following attributes:
  IUNIT - a running sum of units delivered to each base. It is reset to zero after each report interval.
  TOTON - a running sum of total weight delivered to each base. It is reset to zero after each report interval.
  TVVOL - a similar running sum for total volume.
  DPU - a similar running sum for total depot units.
  BASU - a similar running sum for total base units.
  INPRO - a running sum of "in-process" reparables (total units - (sum of depot and base units)).

SETS

BREP - a singly-subscripted FIFO set consisting of all temporary REPs attached to each base.
  Owner: BASE
  Member: REP

RPQ - a doubly-subscripted FIFO set consisting of all temporary RPORTs filed by base and reparable type.
  Owners: REPS, BASE
  Member: RPORT

BSET - a FIFO set, with no subscripts, which contains the numbers of those bases for which a report is to be issued.
  Owner: SYSTEM
  Member: BASE
* * *
1 LRT 0 FE
2 BASS E 1
3 NRPS 0 FC
4 UNIT 2 1
5 TVOL 7 1
6 UNIT 2 1
7 TOPC 2 1
8 TRAS 2 1
9 LRT 0 FC
10 WHT 1 1
11 FRPC 2 1
12 LRT 2 1
13 VOL 1 1
14 UNTET 0 1
15 LANTET 0 1
16 SHEET 1 1
17 FRREP 1 1
18 LREP 1 1

* T REP &
 T TPC 11/2 1
 T SYST 21/2 1
 T WAT 22/2 1
 T VOLL 31/2 1
 T SBREP 51/2 1

* T REP &
 T UNIT 11/2 1
 T IOUNIT 12/2 1
 T TVOL 21/2 1
 T TOPC 22/2 1
 T PACU 31/2 1
 T SREP 52/2 1
 T TRPREP 61/2 1

*TRFC MAIN
 MAIN

C * MAIN PREPARES CARGO PARAMETERS.
C
C LET GNT = CNT
1 READ FROM 9, ID, IDS, KOD, KPY, IR, IRA, ILA, ILB, ITH, ITH, ITH, TYME
 FORMAT(516,56,41,56,41,56,41,56,10,56,12,56,53,5)
 IF (TYME) ) (SIGN), GO TO 200
 CALL REPORT(SIGN)
100 LET GNT = CNT + CNT
C
C......NOT REPORT TIME: PROCESS RECORD.
C
200 FILE FIRST, FOR EACH K IN BSFT, WITH (K) FC(10), IF NONE, GO TO 1
IF (IDS) EQ (5000), GO TO 500
IF (IDS) EQ (5001), GO TO 500
IF (IDS) EQ (5002), GO TO 500
IF (IDS) EQ (5500), GO TO 550
GO TO 1
C
C......IDS = 5000, END OF SIMULATION.
C
500 CALL REP(UNIT)
STOP
C
550 CREATE PIP
STORE WHITELB IN WAIT(KFP)
STORE VOLUMEB IN VOLUME(KFP)
STORE ILB IN SYSTAI(KFP)
IF (IDS) NE (5000), LET HFG(KFP) = 1
FILE KFP IN KREP(16)
GO TO 1
END

*IBFD KLMP2
SUBROUTINE REPORT(UNIT)
C
C......COLLECT PARAMETERS.
C
C......INCREMENT 'THIS PERIOD' DATA.
C
101 GO TO 10, FOR EACH IN LST
10 IF = 7, FOR J = (1)(NRPS)
CREATE REPORT CALLED LIST
1 DO TO 3, FOR EACH KEP IN PRP(1P), WITH SYSTAI(KFP) FC (J)
LET UNIT(LIST) = UNIT(LIST) + 1
LET TOTN(LIST) = TOTN(LIST) + WATE(KFP)/2000
LET TOTVOL(LIST) = TOTVOL(LIST) + VOLUME(KFP)
LET HFG(KFP) = 1, GO TO 2
LET NPSKLIST = NPSKLIST + 1
GO TO 3
2 LET PASU(LIST) = PASU(LIST) + 1
3 REPEAT 1
C
C......COMPUTE 'IN PROCESS' TOTAL FOR THIS REPORT PERIOD.
C
LET KPSEP = (UNIT(LIST) - (PASU(LIST) + ONMCU(LIST)))
STORE KPSEP IN INPSEQLST)
C
C......FILE REPORT IN 'UPUF OF REPORTS MY BASE AND SYSTEM.'
C
FILE LIST IN KPO(J,IP)
C
C......INCREMENT 'TO DATE' COUNTERS.
C
LET TTN(J,IB) = TTN(J,IB) + TTON(LIST)
LET TVOL(J,IB) = TVOL(J,IB) + TVOL(LIST)
LET TUT(J,IP) = TUT(J,IP) + TUNIT(LIST)
LET TDPO(J,IB) = TDPO(J,IB) + TDPO(LIST)
LET TRAS(J,IB) = TRAS(J,IP) + TRAS(LIST)

C
..PROCEED TO RLPS AT NEXT BASE. QUEUE.
C
9 LOOP
C
..PROCURE NEXT BASE.
C
10 REPEAT 101
C
..THERE IS NOW A REPORT FOR EACH REP. BY BASE.
C
C
..CALL REPORT GENERATOR PREDUDE.
C
CALL FORMINGT)
C
..HOUSEKEEP BEFORE RETURNING TO SUPERVISOR.
C
301 DU 1) 30, FOR EACH IB IN RSET
C
22 IF BREP(1B) IS EMPTY, GO TO 23
   . REMOVE FIRST J FROM BREP(1B)
   . DESTROY REP CALLED J
   GO TO 22
23 DO TO 24, FOR J = (J)(NRPS)
   IF RPQ(J,IB) IS EMPTY, GO TO 24
   REMOVE FIRST REPORT FROM RPQ(J,IB)
   DESTROY REPORT
24 LOOP
30 REPEAT 301
RETURN
C
END
*IBFC BLOC3
SUBROUTINE FORMINGT)
C
..PREPARE TO PRINT HEADING.
C
LET KNT = 0
LET MARK= 0
1 DO TO 6, FOR EACH IB IN RSET
2 CALL FORMIB(IB,2NT)
   LET KNT = KNT + 10
   IF(MARKIEQ(11)), GO TO 4
C
C
..PRINT DATA RECORD.
C
3 DO TO 4, FOR EACH J IN RPQ(K,IB)
CALL FORMR(K,TONTON(J),TTON[K],UNIT(J),TUNT(K),TOVOL(J),
1TVOL(K),DPOU(J),DPOU[K],RASU(J),RASU[K],INPRO(J))

LET MARK = 0
LET KNT = KNT + 1
IF (KNT) EQ (55), GO TO 7
4 REPEAT 3
5 LOOP
6 REPEAT 1
RETURN
7 LET KNT = C
LET MARK = 1
GO TO 2
END

*IBFTC FORMH
REPORT FORMH(1B,GNT)

CARGO REPORT FOR BASE

X
X
X
CARGO DELIVERED
TONS UNITS
UNITS DELIVERED
X
TYPE TP TD
X

* PERIOD ENDING **
IB GNT
VOLUME BASE DEPOT
DELIVERED UNITS UNITS IN
TP TL TP TL TP TD PROCESS

*IPFTC FORMR
REPORT FORMR(IA,IB,IC,ID,IE,IF,IG,L,M,N,IC,IP)

X
X
X

* * * * * * * * * *
IF T L M N I0 I1

*ENTRY MAIN

1 1A
1 0 R 0.5
2 0 R 10
3 0 R 17
4 8 2 Z 17 3 10 2
9 0 Z
10 1 R 17 3
011386 10386 8356 7768 8768 9768 7573 8573 9573 14449 14449 200
00250 14685 10462 10462 10462
11 12 2 7 17 3 10 2
13 1 R 17 3
0110 11C 7C 370 370 370 140 140 140 140 10 10 40 40
14 0 R 2
15 0 R 2
16 1 R 10 2
17 1 4 8 3
17 1 P 1 7 11 2
The UTIL (utilization) program describes the Cargo Carrier Utilization for the simulation. The utilization of each vehicle (by ID number) for each vehicle type is listed. Figure 26 is an example of the information contained in this portion of the report.

For each vehicle, col. 3 lists the time the vehicle was available for service, col. 4 the time lost due to maintenance, and col. 5 the time involved in loading the vehicle. Idle time, listed in col. 6, is the sum of maintenance downtime and loading time. The utilization factor is the sum of maintenance time, loading time, and intransit time, divided by the total simulated time to date.

INITIALIZATION

The UTIL output program requires the user to initialize nine variables. Table 9 and its accompanying text will facilitate the use of the program and help the user understand the conceptual basis for the resultant report.

OUTPUT PROGRAM

The input to UTIL is the tape generated by the Depot Transportation Simulation program. The tape is read from logical unit No. 9.

This input tape consists of twelve-word records in the following format:

Word 1 - irrelevant.
Word 2 - a four-digit number identifying the occurrence represented by this record.
Words 3 through 5 - irrelevant.
Word 6 - the base number.
Words 7 through 10 - the content of these words varies with the individual values of Word 2.
Word 11 - irrelevant.
Word 12 - current simulated time.

When a record is read from tape 9, the value of GNT is compared with current simulated time (Word 12); if simulated time equals or
### UTILIZATION OF VEHICLE TYPE 4 FOR PERIOD ENDING 3.00

<table>
<thead>
<tr>
<th>VEHICLE TYPE</th>
<th>VEHICLE ID</th>
<th>TIME AVAILABLE</th>
<th>MAINTENANCE DOWNTIME</th>
<th>LOADING TIME</th>
<th>IDLE TIME</th>
<th>UTILIZATION FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>23951</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>23943</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>23935</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Fig. 26

### Table 9

**VARIABLE DESCRIPTION AND INITIALIZATION:**

**UTILIZATION**

<table>
<thead>
<tr>
<th>Array Number</th>
<th>Description of Variable to Be Initialized</th>
<th>Permanent System Variable Name</th>
<th>Entry Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Report Interval</td>
<td>VNY</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>Number of large carton types</td>
<td>WLASS</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>FXDGQ</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>FLATV</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>LIBQU</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>First vehicle number to be reported - by type</td>
<td>POST</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>Last vehicle number to be reported - by type</td>
<td>LVSET</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>Desired sequence of reports on selected vehicle types</td>
<td>YVSET</td>
<td>A</td>
</tr>
</tbody>
</table>
exceeds the value of \( GNT \), steps are taken to issue a report. Thus, \( GNT \) is used to control the report interval. If \( GNT \) exceeds the value of simulated time, the current record's information is processed as follows.

Word 2 is compared with internal four-digit codes in order to select, for further processing, only records that are relevant to this program's objective. Once such a record is identified, program control is transferred to one of several unique segments corresponding to individual four-digit codes.

When Word 2 is found to equal 5000, the end of simulation has been reached. A final report is issued, and UTIL terminates.

**PERMANENT VARIABLES**

- **CNT** - the report interval, specified by the user.
- **NCARS** - total number of vehicles in the simulation.
- **FBUSQ** - machine address representation of first member of the set BUSQ.
- **LBUSQ** - the last member of BUSQ.
- **FLSTQ** - machine address representation of first member of the set LSTQ.
- **LLSTQ** - the last member of LSTQ.
- **FVSET, LVSET** - first and last members of the set VSET.
- **SVSET** - a list of successive members of VSET, starting with the successor to member FVSET.

**TEMPORARY VARIABLES**

- **BUS** - a temporary entity with the following attributes:
  - **TYPE** - an integer specifying the particular kind of vehicle.
  - **SRNO** - a machine address identifying an individual member of any one TYPE.
  - **MAJOR** - a running sum, for this vehicle, of major maintenance elapsed time.
  - **MINOR** - a running sum, for this vehicle, of minor maintenance elapsed time.
FLITE - a running sum of total travel time accrued by this vehicle.
LOAD - a running sum of time taken to load this vehicle.
T501 - T509 - these represent the simulated times relative to unique events associated with the vehicle during the simulation. They are used to compute the four preceding attributes.
LIST - a temporary entity with the following attributes:

MOPT - total downtime due to maintenance for this vehicle, this report period.
LODTM - total time spent in loading this vehicle during this period.
IDLE - vehicle idle time during this report period.
UTL - utility factor for an individual vehicle during this report period.
THISL - machine ID of this particular report.

SETS

BUSQ - a singly-subscripted FIFO set consisting of all temporary entities called BUS (the vehicles).
Owner: SYSTEM
Member: BUS

LSTQ - a singly-subscripted FIFO set consisting of all temporary entities called LIST (the individual reports).
Owner: SYSTEM
Member: LIST
ICNT C FC
NCARS 0 I
BHUSC 1 I
LHUSC 1 I
FSTWO I I
LLSTQ I I
FVSET 0 I
LVSET 0 I
9SUSET 1 I

* BUS 84

T SUSUSQ 2177 I
T TYPE 2721 I
T SHHL 3112 I
T MAJOR 4 F
T MINOR 5 F
T FLITE 6 F
T LOAD 7 F
T TSO 8 F
T TSO 11 F
T TSO 12 F
T TSO 13 F
T TLS0214 I
T TUG 15 F

* LIST 8

T LVNIT 2 F
T LUI T 3 F
T DOLI 4 F
T DULI 5 F
T SLSNO 6 I
T DULI 7 I

* IRFIC MAIN

MAIN

C......UTILITY FACTOR OUTPUT PROGRAM. SET - UP.
LFT GNT = CNT
1 READ FROM 9, 10, IDS, KOD, KAY, KPH, LB, LTA, LTB, LDC, IHR, INT, TYM
FORMAT (S16, S6, 13, S6, S16, S6, 16, S6, 12, S6, 03, 5)
IF (MYMFLS (CNT)), GO TO 2
CALL UPFRP (CNT)
LET GNT = GNT + CNT

2 IF (IDS) EQ (S010), GO TO 5
IF (IDS) EQ (S010), GO TO 10
IF (IDS) EQ (S020), GO TO 20
IF (IDS) EQ (S030), GO TO 30
IF (IDS) EQ (S040), GO TO 40
IF (IDS) EQ (S050), GO TO 50
IF IDS EQ (5090), GO TO 60
IF IDS EQ (5200), GO TO 70
IF IDS EQ (5300), GO TO 80
GO TO 1
C
C ORDER PARAMETERS.
C
90 CREATE BUS CALLED K
STORE ILA IN TYPE(K)
STORE ITR IN SRNO(K)
FILE K IN BUSQ(ILA)
GO TO 1
70 FIND FIRST, FOR EACH K IN BUSQ(ILA), WITH (SRNC(K))EQ(ITR), IF
NONE, GO TO 1
 IF (T504(K)) LS (GNT-CNT), LET FUGE(K) = FUGE(K) + ((GNT-CNT) - T504(K))
LET MAJOR(K) = MAJOR(K) + (TYME - T504(K))
GO TO 1
C
60 DO TO 59, FOR J = (1)(NCARS)
FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNC(K))EQ(ITR), IF NONE,
1GO TO 59
STORE TYME IN T509(K)
GO TO 1
59 LOOP
GO TO 1
5C DO TO 49, FOR J = (1)(NCARS)
FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNC(K)) EQ (ITR), IF NONE,
1GO TO 49
 IF (T504(K)) LS (GNT-CNT), LET FUGE(K) = FUGE(K) + ((GNT-CNT) - T504(K))
LET FLITE(K) = FLITE(K) + (TYME - T504(K))
GO TO 1
49 LOOP
GO TO 1
40 DO TO 39, FOR J = (1)(NCARS)
FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNC(K)) EQ (ITR), IF NONE,
1GO TO 39
STORE TYME IN T504(K)
 IF (T502(K)) EQ (0.), GO TO 41
 IF (T502(K)) LS (GNT-CNT), LET FUGE(K) = FUGE(K) + ((GNT-CNT) - T502(K))
LET LOAD(K) = LOAD(K) + (T504(K) - T502(K))
GO TO 42
41 IF (T501(K)) LS (GNT-CNT), LET FUGE(K) = FUGE(K) + ((GNT-CNT) - T501(K))
LET LOAD(K) = 0.
42 IF (T502(K)) EQ (0.), LET MINOR(K) = MINOR(K) + (T504(K) - T501(K))
 IF (T502(K)) NF (0.), LET T502(K) = 0.
GO TO 1
39 LOOP
GO TO 1
30 GO TO 20
20 DO TO 19, FOR J = (1)(NCARS)
FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNC(K)) EQ (ITR), IF NONE,
1GO TO 19
 IF (T5C1(K)) LS (GNT-CNT), LET FUGE(K) = FUGE(K) + ((GNT-CNT) - T5C1(K))
LET MINOR(K) = MINOR(K) * (TYME - T501(K))
STORE TYME IN T502(K)
GO TO 1
19 LOOP
GO TO 1
10 DO TO 9, FOR J = (1)(MCARS)
FIND FIRST, FOR EACH K IN BUSQ(J), WITH (SRNO(K))FQ(ITR), IF NONE,
GO TO 9
STORE TYME IN T501(K)
GO TO 1
9 LOOP
GO TO 1
5 CALL UPREP(GNT)
STOP
FND
*IRFTC BLK6
SUBROUTINE UPREP(GNT)
C
C......COLLECT PARAMETERS.
C
1 DO TO 4, FOR EACH IV IN VSET
2 DO 3, FOR EACH K IN BUSQ(IV)
CREATE LIST CALLED L
LET THISL(L) = SRNO(K)
LET MNTOT(L) = MNTOT(L) + (MAJOR(K) + MINOR(K))
LET LOOTM(L) = LOOTM(L) + LOAD(K)
LET IDL=(L)-IDE(L)+(CNT-((FLITE(K)+MNTOT(L)+LOOTM(L))-FUGE(K)))
LET UTIL(L) = UTIL(L) + (CNT-IDLE(L))/ CNT
FILE L IN LSTQ(IV)
3 REP FAT 2
4 REPEAT 1
CALL FORM(GNT)
C
C......HOUSEKEEP BEFORE RETURNING TO SUPERVISOR.
C
5 DO TO 8, FOR EACH IV IN VSET
61 DO TO 6, FOR EACH BUS IN BUSQ(IV)
LET MAJOR(IBUS) = 0.
LET MINOR(IBUS) = 0.
LET FLITE(IBUS) = 0.
LET LOAD(IBUS) = 0.
LET FUGE(IBUS) = 0.
6 REPEAT 61
C
7 IF LSTQ(IV) IS EMPTY, GO TO 8
REMOVE FIRST LIST FROM LSTQ(IV)
DESTROY LIST
GO TO 7
8 REPEAT 5
RETURN
END
*IRFTC BLK6
SUBROUTINE FORM(GNT)
C
C......PREPARE TO PRINT HEADING.
C
LET KNT = 0
LET MKRT = 0
1 DO TO 5, FOR EACH IV IN VSFT
2 CALL FORMH(IV, GNT)
   LET KNT = KNT + 10
   IF (MARK) EQ(1), GO TO 4
C
C......PRINT DATA RECORD.
C
3 DO TO 4, FOR EACH L IN LSTQ(IV)
   IF (UTL(L)) EQ(-0.), LET UTL(L) = 0.
   LET AVL = CNT - MNTD(L)
   CALL FORMR(IV, TL, AVL, MNTD(L), LODT(L), MNTD(I), UTL(1))
   LET MKRT = 0
   LET KNT = KNT + 1
   IF (KNT) EQ(55), GO TO 6
4 REPEAT 3
5 REPRTAT 1
RETURN
6 LET KNT = 0
LET MKRT = 1
GO TO 2
END
*IBFTC FORMR
REPORT FORMR(IV, L, A1, A2, A3, A4, A5)

* ENTRY MAIN
1 9
   1 OR 1.0
   2 OR 4
   3 617 42 1
   7 OR 1
   8 OR 4
   9 1D 42 4(12)
   2 3 4
Program 12

DEPOT MAINTENANCE CAPABILITY
The Depot Maintenance Capability program is used to display the outputs from the DR&O Simulator. The report consists of five parts: the input to each depot, its output and the repairable repair times for the period(s) of time selected; queueing and utilization factors for each resource group (personnel and equipment groups); queueing factors for each component spare part type; stock levels, component spare repair times, stockouts, and demands for each component spare part; and detailed information for each activity about its performance during each period of simulation.

An example of the output display is shown in Figs. 27 to 31. Figure 27 is a display of depot statistics showing the system (or Unit or item) arrivals and departures, repairables in process, and repair cycle times. A separate display is presented for each depot. The example display is for depot No. 3.

The first line of data shows the time at which the statistics were taken. Notice that the report is for day ending 14.000. Since "time" began at time 0.000 in the simulations and the report is initialized for seven-day periods, the fourteenth day will end at time 14.000 (not 14.999). The next line entry shows the number of repairables that entered the depot (15) for the period and the sum of all repairables entering the depot (30) as of the report period.

The third line entry is the serviceables departing the depot (returned to serviceable stock) (8), during the period, and the sum of all items processed to date (20). The difference between the arrivals and departures is presented as the number in process (10).

The next line displays a distribution of the repairables in process. The average time in process (for the twenty that were processed) is 4.75 days, the maximum time was 10 days, and the minimum 1 day. The standard deviation for the distribution is 2.37.

The repair time distribution is presented both for the period (just 7 days) and accumulated for the fourteen days.

Figure 28 displays the activity Queueing Factors for the period. Column 1 lists the activities in sequence. Columns 2, 3, and 4 list the
distribution of the quantity of reparables processed by each activity for the period. Column 5 lists the average time that the reparables spend in queue behind each activity awaiting some resource. The average queue time is displayed as work time; i.e., off-shift time is not included.

Figure 29 is the Personnel Utilization report. For each personnel type, listed in col. 1, the sum of all personnel on duty for all shifts of the period (of course, the period may be only one shift) is presented in col. 2. The utilization factor, which is the time actually engaged in a process divided by the total duty time available, is presented in col. 3 for each personnel type. The balance of the display is devoted to the man-hours used at each activity for each personnel type. For example, personnel type 2 worked at activities 5, 10, and 11, and a total of 93.25 man-hours were used during the simulation period (seven days).

Figure 30 is the Equipment Utilization report. By equipment type, listed in col. 1, the quantity is listed in col. 2; cols. 3, 4, and 5 list the time the equipment was used, the idle time, and the downtime (all in decimal-days). Note that the summation of these three columns is equal to 14 equipment days for Equipment types 1 and 2, and 21 equipment days for Equipment type 3. This is the total time available for the equipment. Off-shift time is not deducted.

Column 6 is a count of the number of times the depot equipment failed during the period (in this example, 7 days). Column 7 is the utilization factor for the equipment, computed by dividing the total time available (equipment days) into the time in use. Column 8 lists the activities where the equipment was used.

Figure 31 is the display of the Queueing factors for each component spare part. Column 1 lists the spare part ID number. Column 2 lists the quantity or authorized stock level of each spare part. Column 3 lists the number of demands for each spare part during the period.

Columns 4, 5, and 6 list the distribution of the quantity of unfilled demands (average, maximum, and minimum) for each spare part type. Column 7 lists the average queue time—the average time required to fill the demand.
DEPOT 3
SYSTEM ARRIVALS, DEPARTURES, IN-PROCESS, "O REPAIR CYCLE TIMES

DATA FOR PERIOD ENDING DAY 14.000

REPARABLES ENTERING DEPOT THIS PERIOD 15, TO DATE 30
SERVICEABLES DEPARTING DEPOT THIS PERIOD 0, TO DATE 20

NUMBER IN PROCESS 10

REPARABLES IN PROCESS AVG MAX MIN STD DEV
4.75 10 1 2.37

REPAIR TIME

THIS PERIOD AVG MAX MIN STD DEV
1.99 3.03 0.96 0.69
1.37 3.03 0.88 0.67

Fig. 27

ACTIVITY QUEUEING FACTORS
FOR PERIOD JUST COMPLETED

<table>
<thead>
<tr>
<th>ACT. NO.</th>
<th>NO. OF REPS IN QUEUE</th>
<th>AVG</th>
<th>MAX</th>
<th>MIN</th>
<th>AVG QUEUE TIME (IN WORK-HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.33</td>
<td>3</td>
<td>0</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>3</td>
<td>0</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>1</td>
<td>0</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.10</td>
<td>9</td>
<td>0</td>
<td>0.76</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.18</td>
<td>8</td>
<td>0</td>
<td>1.38</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.00</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4.15</td>
<td>10</td>
<td>0</td>
<td>7.44</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.27</td>
<td>3</td>
<td>0</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>1</td>
<td>0</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.00</td>
<td>2</td>
<td>0</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 28
### PERSONNEL UTILIZATION

<table>
<thead>
<tr>
<th>PERS TYPE</th>
<th>QTY</th>
<th>UTIL FACT</th>
<th>MAN HOURS</th>
<th>MAN HOURS</th>
<th>MAN HOURS</th>
<th>MAN HOURS</th>
<th>MAN HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42</td>
<td>0.09</td>
<td>4.41</td>
<td>11.02</td>
<td>4.83</td>
<td>3.01</td>
<td>5.79</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>0.32</td>
<td>5.40</td>
<td>10.20</td>
<td>4.10</td>
<td>11.75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>0.27</td>
<td>4.03</td>
<td>9.70</td>
<td>8.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>0.40</td>
<td>32.22</td>
<td>12.18</td>
<td>22.05</td>
<td>9.63</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 29

### EQUIPMENT UTILIZATION

<table>
<thead>
<tr>
<th>EQUIPMENT TYPE</th>
<th>QUANTITY</th>
<th>TIME IN USE</th>
<th>IDLE TIME</th>
<th>DOWN TIME</th>
<th>NO. OF FAILURES</th>
<th>UTILIZATION FACTOR</th>
<th>ACTIVITY NO. WHERE USED</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.67</td>
<td>13.15</td>
<td>0.18</td>
<td>3</td>
<td>0.048</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.26</td>
<td>12.35</td>
<td>0.39</td>
<td>11</td>
<td>0.090</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.70</td>
<td>19.29</td>
<td>0.01</td>
<td>1</td>
<td>0.081</td>
<td>5</td>
</tr>
</tbody>
</table>

Fig. 30

### QUEUEING FACTORS BY COMPONENT SPARES TYPE

#### REPARABLE QUEUE LENGTHS AND TIMES BY COMPONENTS

<table>
<thead>
<tr>
<th>COMPONENT TYPE</th>
<th>QUANTITY</th>
<th>DEMANDS THIS PERIOD</th>
<th>NO. OF REPS IN QUEUE</th>
<th>AVG QUEUE TIME (IN WORK-HOURS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>26</td>
<td>AVG</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>MAX</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>21</td>
<td>MIN</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>18</td>
<td>AVG</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>27</td>
<td>MAX</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Fig. 31
INITIALIZATION

The Depot Capability report program requires the initialization of 83 variables. Only 12 require values, however. The Depot Capability Variable Description and Initialization Table (Table 10) contains the information required to initialize the report program. An example initialization data deck listing follows the "Output Program" listing.

OUTPUT PROGRAM

The input to this program is the binary tape generated by the DM&O Simulation Program; this tape is read from logical unit No. 9.

The input tape consists of 12-word label records with the following format:

Word 1 - irrelevant.
Word 2 - IDD - a four-digit number identifying the "event" or "occurrence" represented by this record.
Word 3, 4, 5 - irrelevant.
Word 6 - INBASE - the depot number.
Words 7, 8, 9, 10 - IV1, IV2, IV3, LADDR. These fields are used to store various items of information, depending on the value of IDD.
Word 11 - INDIC - 1 if the next record is a detail record (to be skipped), 0 otherwise.
Word 12 - RTIME - current simulated time.

When a label record is read, the value of INBASE is compared with the constant permanent attribute called BASE; if they are unequal, the record is skipped. (Thus it would require n runs of this analysis program to process all the data from an n-base simulation run, each time changing the value of BASE).

If the new RTIME is greater than the previous one, subroutine CLOCK is called to check for the end of the operating shift and the end of the report period. If the report period has ended, subroutine ENDPKD is called to generate the reports. Subroutine CLOCK also updates TIME, which is the actual work time elapsed since the beginning of
Table 10

VARIABLE DESCRIPTION AND INITIALIZATION:
DEPOT MAINTENANCE CAPABILITY

| Array Number | Number of Subscribers | Mode | Initialization to | Initial Value in | Array Number of Attributes to Be Exported in Fig. 5 Col. | Description of Variable to Be Initialized |
|--------------|-----------------------|------|------------------|------------------|----------------------------------------------------------|
| 1-10         | 0                     | 1    | V                | 2                | 10                                                       | Total number of depots                        |
| 11-20        | 1                     | 1    | V                | 3                | 10                                                       | Total number of Activities                    |
| 31-36        | 1                     | 1    | V                | 4                | 10                                                       | Total number of Personnel Types               |
| 41-45        | 1                     | 1    | V                | 5                | 10                                                       | Total number of Equipment Types               |
| 51-56        | 1                     | 1    | V                | 6                | 10                                                       | Total number of Spare Part Types              |
| 61-65        | 1                     | 1    | V                | 7                | 10                                                       | Total number of Repairable Types             |
| 71-75        | 1                     | 1    | V                | 8                | 10                                                       | List depot number where each repairable type is processed (one data card/repair type) |
| 81-85        | 1                     | 1    | V                | 9                | 10                                                       | Total number of shifts per week               |
| 91-95        | 1                     | 1    | V                | 10               | 10                                                       | Insert a "1" for each shift worked and a "0" for each shift not worked (one data card/shift) |
| 101-105      | 1                     | 1    | V                | 11               | 10                                                       | Insert quantity of Personnel type "P" at shift "P" (one data card/personnel type) |
| 111-115      | 1                     | 1    | V                | 12               | 10                                                       | Total number of hours per shift               |
| 121-125      | 1                     | 1    | V                | 13               | 10                                                       | Report interval in decimal days               |
| 131-135      | 1                     | 1    | V                | 14               | 10                                                       | Base number                                  |

Notes: E = Entity, A = Attribute.
simulation. (The automatically defined system variable TIME is used, in order to take advantage of the ACCUMULATE statement.)

Then the appropriate subroutine is called to process the label record. To each significant IDD number, there corresponds a subroutine: e.g., subroutine NEXTAC is called whenever IDD equals 4400. If IDD does not match any of the significant numbers, it is skipped.

If IDD = 3, the end of simulation has been reached; the program terminates after writing the last set of reports.

Error tests intended for the debugging phase have been left in the program, sprinkled throughout. If an error is encountered, this means that something is amiss in this program, in the simulation program, or in the initialization deck. Subroutine ERROR is called, which terminates after outputting the current value of RTIME and a four-letter abbreviation identifying the routine in which the error was detected. For instance, "REA2" refers to the second error condition in subroutine READY.

Subroutine SNAP outputs a "snapshot" of all permanent and temporary variables, as an aid to debugging. The user may insert, at any point, a call to SNAP with an identifier of one to four letters and/or digits; e.g., CALL SNAP (4RNAME). In this example, "NAME" is the identifier. In the current version of the program, ERROR calls SNAP before terminating.

**PERMANENT VARIABLES**

This list is complete except for attributes denoting first-of-set or last-of-set, and attributes used only to keep track of time in an ACCUMULATE statement such as TQSZA (these always have names beginning with "T").

RTIME - current simulated time; it is obtained from each label record as it is read in.

STIME - the "RTIME" of the previous label record.

TIME - (a variable automatically defined by the system) - number of workdays elapsed since the beginning of simulation. Suppose there are 40 work-hours in a week. Then if RTIME = 7.0, TIME will be equal to 1.6667 or 1-2/3 (which is 40 divided by 24).
PTIME - the value of "TIME" at the end of the previous report period.
ENDSH - the "RTIME" at which the current shift will end.
ENDPD - the "RTIME" at which the current period will end.
ETIME - the value of "RTIME" at the end of the previous report period.
CURPD - the length in workdays (using "TIME") of the period just completed.
CURP - the length in simulated time (using "RTIME") of the period just completed.
CURSH - number of current shift (on a weekly cycle).
CURAC - activity number associated with current label record.
CURSP - spare part number associated with current label record.
CUREP - I.D. number of REP associated with current label record.
SHOP - permanent entity, of which the following are attributes:
   RIN - number of reps entering this depot this period.
   ROUT - number of reps leaving depot this period.
   TRIN - total number of reps in depot (since the beginning of simulation).
   TROUT - total number of reps that have left this depot.
   MAXR - maximum number of reps in depot this period.
   MINR - minimum number of reps this depot this period.
   RIP - number of reps currently in process in this depot.
   RIPS - a running sum of all the values that RIP has assumed during this period.
   RIPSQ - a running sum-square total of all the values that RIP has assumed during this period. E.g., if RIP has had the values 2, 3, 4, 3, 2 in this period, then RIPS is 2 + 3 + 4 + 3 + 2 or 14, and RIPSQ is 2² + 3² + 4² + 3² + 2² or 42.
   RTS - sum of the repair times of all reps leaving depot this period.
   RTSQ - sum of squares of repair times of all reps leaving depot this period.
   TRIS - sum of RTS for all periods to date.
   TRISQ - sum of RTSQ for all periods to date.
   MRT - maximum repair time for depot this period.
MNRT - minimum repair time for depot this period.
TMXRT - maximum repair time for depot, all periods.
TMNRT - minimum repair time for depot, all periods.

ACTIV - activity; a permanent entity, of which the following are attributes:
QSZA - current queue size at this activity.
CQSZA - cumulative total of QSZA, this period.
MNQSA - minimum value of QSZA, this period.
MNQSA - minimum value of QSZA, this period.
TIAQS - "time in activity queue, summed:" the total time, in workdays, that reps have spent in the queue for this activity.
AVQSA - average queue size at this activity.
AVTAQ - average time in queue for this activity.
AQOUT - number of reps that have left the queue of this activity during this period.

PTYPE - personnel type; a permanent entity, of which the following are attributes:
QTYS - total number of this type of personnel.

ETYPE - equipment type; a permanent entity with the following attributes:
QTYE - total quantity of this equipment type.
NTAIL - number of failures of this type of equipment during this period.

SPTYP - spare part type; a permanent entity with the following attributes:
QTYSP - quantity of spares of this type available at beginning of simulation.

D clam - number of demands for this type of part during this period.
FILL - number of times that such a demand was filled.
QSP - queue size for this type of part.
CQSP - cumulative total of QSP, this period.
MXQSP - maximum value of QSP, this period.
MNQSP - minimum value of QSP, this period.
TISQS - total time, in work days, that reps have spent in the queue for this type of part.
AVQSP - average value of QSP, this period.
AVTSQ - average time in queue for this type of spare part.
RPTYP - rep type; a permanent entity with the following attribute:
  SHPNO - number of the depot to which this type of rep belongs.
SHIFT - a permanent entity with the following attribute:
  SCHED - 1 if this is a work shift; 0 if this is an off shift.
QTYPR - a permanent attribute with two subscripts:
  first subscript: PTYPE
  second subscript: SHIFT
  meaning: the quantity of personnel of this type, on duty during this shift.
LENSH - the length of a shift.
PEROD - the length of a report period.
BASE - the number of the depot for this run; all label records pertaining to any other depot will be ignored.

TEMPORARY VARIABLES

REP - a temporary entity with the following attributes:
  QTIME - the value of "TIME" when the rep entered the queue for an activity.
  ETIME - the value of "RTIME" when the rep entered the system.
  IDNO - the I.D. number of the rep: a number obtained from the label record, representing the absolute storage address of the rep in the simulation run.
  QFLAG - a number which is equal to zero unless the rep is in the queue for an activity, in which case QFLAG equals the number of that activity.
  SFLAG - equal to zero unless rep is in the queue for a spare part, in which case SFLAG equals the number of that type of part.
SPIM - the value of "TIME" when the rep entered the queue for a spare part.

PLOAD, SLOAD, PACTQ, SACIQ - attributes associated with the sets LOAD and ACTQ.

DUMMY - a temporary entity whose purpose is to save information to be output in Fig. 28. It has two attributes:

ACNO - the number of an activity at which this type of equipment is to be used.

SSET - successor in the set called "SET."

ENTRY - a temporary entity having to do with the utilization of personnel at different activities. Its attributes are:

ACNO - the number of an activity at which this type of personnel is used.

WKING - number of personnel of this type working at activity whose number equals ACNO.

CWKING - cumulative total of WKING, this period.

TWKING - the value of "RTIME" when CWKING was last updated.

PLIST, SLIST - attributes associated with the set called "LIST."

SETS

LOAD - a set with one subscript, ranked on BTIME.

Owner: SHOP

Member: REP

The LOAD of each SHOP consists of all the reps that are currently in process in that shop.

ACTQ - a set with one subscript, ranked on BTIME.

Owner: ACTIV

Member: REP

ACTQ is the queue of all reps currently waiting at an activity.

SET - a FIFO set with one subscript.

Owner: ETYPE

Member: DUMMY

SET is the set of all activities at which this type of equipment can be used. This information is to be output in Fig. 30.
LIST - a set with one subscript, ranked on ACNO.

Owner: PTYPE
Member: ENTRY

LIST has one ENTRY for each activity at which this type of personnel has been used during this report period.

Standard Names for Local Variables

IACNO always means activity number
IEQNO always means equipment number
IPERNO always means personnel number
ISPNO always means spare part number
IREPNO always means rep number
ID or IDREP always means I.D. number of rep
IQTY always means quantity or number
### DEPOT CAPABILITY OUTPUT PROGRAM

#### T REP 8

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T CTIME</td>
<td>F</td>
</tr>
<tr>
<td>T BTIME</td>
<td>F</td>
</tr>
<tr>
<td>T IDNC</td>
<td>I</td>
</tr>
<tr>
<td>T QFLAG</td>
<td>I</td>
</tr>
<tr>
<td>T SFLAG</td>
<td>I</td>
</tr>
<tr>
<td>T SPIN</td>
<td>F</td>
</tr>
<tr>
<td>T TLOAD</td>
<td>I</td>
</tr>
<tr>
<td>T SACTQ</td>
<td>I</td>
</tr>
<tr>
<td>T PLOAD</td>
<td>I</td>
</tr>
<tr>
<td>T PACTQ</td>
<td>2/2</td>
</tr>
</tbody>
</table>

#### T DUMMY2

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET</td>
<td>2</td>
</tr>
</tbody>
</table>

#### T ENTRY8

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T ACNO</td>
<td>I</td>
</tr>
<tr>
<td>T WKING</td>
<td>F</td>
</tr>
<tr>
<td>T CWKNG</td>
<td>3</td>
</tr>
<tr>
<td>T TWKNG</td>
<td>4</td>
</tr>
<tr>
<td>T PLIST</td>
<td>5</td>
</tr>
<tr>
<td>T SLIST</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRTIME</td>
<td>F</td>
</tr>
<tr>
<td>2STIME</td>
<td>F</td>
</tr>
<tr>
<td>3PTIME</td>
<td>F</td>
</tr>
<tr>
<td>4ENCSH</td>
<td>F</td>
</tr>
<tr>
<td>5ENGCW</td>
<td>F</td>
</tr>
<tr>
<td>6CPRP</td>
<td>F</td>
</tr>
<tr>
<td>7CURSH</td>
<td>I</td>
</tr>
<tr>
<td>8CURRC</td>
<td>I</td>
</tr>
<tr>
<td>9CURSP</td>
<td>I</td>
</tr>
<tr>
<td>10SHCP</td>
<td>E</td>
</tr>
<tr>
<td>11FLCAD</td>
<td>1</td>
</tr>
<tr>
<td>12LLCAD</td>
<td>1</td>
</tr>
<tr>
<td>13RIN</td>
<td>I</td>
</tr>
<tr>
<td>14RGUI</td>
<td>1</td>
</tr>
<tr>
<td>15TRIN</td>
<td>1</td>
</tr>
<tr>
<td>16TRMC</td>
<td>1</td>
</tr>
<tr>
<td>17PAKR</td>
<td>1</td>
</tr>
<tr>
<td>18PRIP</td>
<td>1</td>
</tr>
<tr>
<td>19RIP</td>
<td>1</td>
</tr>
<tr>
<td>20RIPS</td>
<td>1</td>
</tr>
<tr>
<td>21RIPSC</td>
<td>1</td>
</tr>
<tr>
<td>22RTS</td>
<td>F</td>
</tr>
<tr>
<td>23RTSC</td>
<td>1</td>
</tr>
<tr>
<td>24RTS</td>
<td>1</td>
</tr>
<tr>
<td>25RTSC</td>
<td>1</td>
</tr>
<tr>
<td>26MTXRT</td>
<td>1</td>
</tr>
<tr>
<td>27MTXRT</td>
<td>1</td>
</tr>
<tr>
<td>28TPXRT</td>
<td>1</td>
</tr>
</tbody>
</table>
*IBFTC MAIN

MAIN ROUTINE
CALL PRELIP
C ...... READ A LABEL RECORD
READ (9) K, IOD, K, K, INBASE, IV1, IV2, IV3, IADDR, INCIC, T
LET RTIME = T
C ...... IF THERE IS A DETAIL RECORD, SKIP OVER IT
X IF (INDIC.EQ.11) READ (9) JUNK
C ...... TERMINATE IF AN ENDSIM RECORD (WTH ICC=3) IS ENCOUNTERED
    IF (IOD) NE (3), GO TO 30
    CALL CLOCK
    LET ENDPD = RTIME
    CALL ENDORE
    STOP
C ...... SKIP THIS RECORD IF IT DOES NOT PERTAIN TO THE RIGHT BASE
3C IF (INBASE) NE (BASE), GC TC 10
    IF (RTIME) GE (STIME), CALL CLOCK
C ...... CALL THE APPROPRIATE ROUTINE FOR THIS ICD NUMBER:
    IF (IOD) EQ (7000), GO TC 50
    IF (IOD) EQ (7002), GC TC 52
    IF (IOD) EQ (7003), GC TC 54
    IF (IOD) EQ (7004), GC TC 56
    IF (IOD) EQ (7005), GC TC 58
    IF (IOD) EQ (7200), GC TC 60
    IF (IOD) EQ (6000), GC TC 62
    IF (IOD) EQ (7400), GC TC 64
    IF (IOD) EQ (7401), GC TC 66
    IF (IOD) EQ (7460), GC TC 68
    IF (IOD) EQ (7550), GC TC 70
    IF (IOD) EQ (7370), GC TC 72
    IF (IOD) EQ (7560), GC TC 74
    IF (IOD) EQ (7600), GC TC 76
    IF (IOD) EQ (7700), GC TC 78
    IF (IOD) EQ (7350), GC TC 80
    IF (IOD) EQ (7355), GC TC 82
    IF (IOD) EQ (7800), GC TC 84
    IF (IOD) EQ (7801), GC TC 86
C ...... FOR ANY OTHER VALUE OF ICD, SKIP THIS RECORD
    GO TO 10
5C CALL ACTVTY (IV1)
    GO TC 1C
52 CALL EQATAC (IV1)
    GO TC 10
54 CALL PRSNEI (IV1, IV2, IV3)
    GO TO 1C
56 CALL EQUIP (IV1, IV2)
    GO TO 10
58 CALL SPARES (IV1, IV2)
    GO TO 1C
60 CALL ARRIV (IV1, IADDR)
    GO TO 10
62 CALL DEPART (IV1, IADDR)
    GO TO 10
CALL NEXTAC (IV1, IV3, IADDR)
GO TO 10
CALL READY (IV3)
GO TO 10
CALL ASINPR (IV1, IV3)
GO TO 10
CALL RLESPR (IV1, IV3, IADDR)
GO TO 10
CALL ASINEC (IV1, IV3)
GO TO 10
CALL RLESEQ (IV1, IV3)
GO TO 10
CALL FAIL (IV3)
GO TO 10
CALL RESTOR (IV3)
GO TO 10
CALL SPAVL (IV1)
GO TO 10
CALL IASPO (IV1, IV2)
GO TO 10
CALL SPRET (IV1)
GO TO 10
CALL LVSPQ (IV1, IADDR)
GO TO 10
END

*IBFTC PRELIP

SUBROUTINE PRELIP

C .......INITIALIZE SCPE SYSTEM VARIABLES
LET ENDPD = PERCD
LET ENDSH = LENSN
LET CURSH = 1

C .......INITIALIZE EACH PAPPUM TC A VERY LARGE NUMBER
DO TO 20, FOR EACH SHOP I
LET MNRT(I) = 1000C.
LET TNRT(I) = 10000.

20 LOOP
RETURN
END
*IBFTC CLOCK

SUBROUTINE CLOCK
C       THIS ROUTINE KEEPS TRACK OF TIME, ENC-CF-PERIOD, AND ENC-OFF
C       SHIFT. 'TIME' IS THE ACTUAL WORK-TIME ELAPSED SINCE
C       THE BEGINNING OF SIMULATION, WHEREAS 'RTIME' IS THE
C       CURRENT SIMULATED TIME.
 2C LET T = AMnwRTIME, ENDsh, ENCpD)
C       UPDATE TIME IF SOME WORK-TIME HAS ELAPSED, THAT IS, IF THIS
C       IS A WORKING SHIFT
    IF (SCHEDICURSH)) EQ (1), LET TIME = TIME + T - STIME
C       UPDATE THE NO. OF PAH-DAYS (CGTY) FOR EACH PERSONNEL TYPE I
C       DO TO 40, FCR EACH PTYPE I
C       ACCUMULATE QTYPR1(CURSH) INTO CGTY(I) SINCE TQTY(I)
C       THERE IS AN END-CF-SHIFT AND/OR AN ENC-CF-PERIOD, DETERMINE
C       WHICH CAPE FIRST.
C       END-OF-PERIOD
C       CALL ENOPRD
C       GO TO 20
C       END-OF-SHIFT. UPDATE CURS1 AND EndSH.
C       LET CURS1 = MOD (CURSH, #SHIFT) + 1
C       LET ENOSH = ENOSH + LEASH
C       GO TO 20
C      RETURN
C      END

*IBFTC ENOPRD

SUBROUTINE ENOPRD
C       END OF A REPORT PERIOD.
C       COMPLETE CURP AND CLRPD.
C       LET S = STIME
C       LET CURP = S - ETIME
C       LET ETIME = S
C       LET CLRPD = TIME - PTIME
C       LET PTIME = TIME
C       IF NO WORK-TIME HAS ELAPSED, DON'T OUTPUT ANYTHING
C       IF (CLRPD) EQ 10., GC TC 100
C       OUTPUT THE REPORTS FOR THIS PERIOD
C       CALL OUT1
C       CALL OUT2
C       CALL OUT3
C       CALL OUT4
C       CALL OUT5
C       LET ENOPD = ENOPD + PERCD
C      RETURN
C      END
*IBFTC ACTIVITY

SUBROUTINE ACTIVITY (IACNC)

C .......... THIS ROUTINE IS CALLED WHEN ICC=7000. (AT BEGINNING OF RUN)
          IF (IACNO) GR (NACTIV), CALL ERROR (4HACTV)
          LET CURAC = IACNC
          RETLRN
          END

*IBFTC EQATAC

SUBROUTINE EQATAC (IEGNC)

C .......... THIS ROUTINE IS CALLED WHEN ICC=7002. (AT BEGINNING OF RUN)
          IF (IEGNO) GR (NETYPE), CALL ERROR (4HEQAT)
          C .......... SAVE ACTIVITY NUMBERS FOR REPORT NC. 4
          CREATE DLMPY CALLED ITEP
          LET ACNODETCITEM = CURAC
          FILE ITEP IN SET(IEGNO)
          RETLRN
          END

*IBFTC PRSNEL

SUBROUTINE PRSNEL (IPERAC, ICTY, ISHIFT)

C .......... THIS ROUTINE IS CALLED WHEN ICC=7003. (AT BEGINNING OF RUN)
          IF (IPERNO) GR (NPTYPE), CALL ERROR (4HPRSN)
          LET QTYY(IPERNO) = QTYY(IPERAC) + ICTY
          LET QTYP(IIPERNC,ISHIFT) = ICTY
          RETLRN
          END

*IBFTC EQUIP

SUBROUTINE EQUIP (IEGNC, ICTY)

C .......... THIS ROUTINE IS CALLED WHEN ICC=7004. (AT BEGINNING OF RUN)
          IF (IEGNO) GR (NETYPE), CALL ERROR (4HEQUI)
          LET QTYE(IEGNC) = ICTY
          RETLRN
          END
SUBROUTINE SPARES (ISPAC, IQTY)
C
        IF (ISPAC) = (ISPAC) + 1
C
        RETURN
END

SUBROUTINE ARRIV (IREPAC, IC)
C
        CREATE REP
        LET BTIME(REP) = RTIME
        LET IDNO(REP) = ID
        LET ISHOP = SHPNO(REP)
        LET NEWRIP = RIP(ISHCP) + 1
        LET RIP(ISHOP) = NEWRIP
        LET RIN(ISHOP) = RIN(ISHCP) + 1
        LET MAXR(ISHOP) = MAXO (NEWRIP, MAXR(ISHCP))
        LET FRIP = NEWRIP
        LET RIPS(ISHOP) = RIPS(ISHCP) + FRIP
        LET RIPSG(ISHOP) = RIPSG(ISHCP) + FRIP+2
        FILE REP IN LOAD(ISHOP)
        RETURN
END
SUBLTINE DEPART (IREPAC, IC)

C .........THIS ROUTINE IS CALLED WHEN ICC=6000.

C .........A REP HAS LEFT THE SYSTEM. REMOVE AND DESTROY IT.

C LET ISP1O = S10PNC(IREPAC)

C FIND FIRST REP, FOR EACH REP IA LCAC(I$HCPI), WITH

C (I$NC(REP))EQ(I$D), WHERE REP, IF A$NE, CALL ERROR (4+DEPA)

C REMOVE REP FROM LCAC(I$HCPI)

C LET NEWREP = R$IP(I$HCPI) - 1

C LET R$IP(I$HCPI) = NEWREP

C LET ROLTI$HCPI) = A$CUT(I$HCPI) + 1

C LET M$IPR(I$HCPI) = PINO (NEWREP, PINR(I$HCPI))

C LET FRIP = NEWREP

C LET RIPS(I$HCPI) = RIPS(I$HCPI) + FRIP

C LET RIPS$HCPI) = RIPS$HCPI) + FRIP

C LET R$TIP = RT$PE - BT$PE(REP)

C LET M$RTI$HCPI) = AMAX1 (REPTIP, M$RTI$HCPI)

C LET M$RTI$HCPI) = AMIN1 (REPTIP, M$RTI$HCPI)

C IF R$TIP = RT$SQ(I$HCPI) + REPTIP

C DESTROY REP

C RETURN

C END

C

SUBLTINE NEXTAG (IREPAC, IACAC, ICREP)

C .........THIS ROUTINE IS CALLED WHEN ICC=7400.

C .........A REP IS SUBMITTED (OR RE-SUBMITTED) TO THIS ACTIVITY.

C LET CURAC = IACAC

C .........DO NOTHING IF ACTIVITY = 0 (RECEIVING).

C IF (IACAC) EQ (10), QC TO 50

C LET CUREP = IDREP

C .........FIND THE REP BY SEARCHING THE LOAD CF THE APPROPRIATE DEPOT

C FIND FIRST REP, FOR EACH REP IA LCAC(I$HCPI), WITH

C (I$NC(REP)) EQ (IDREP), WHERE REP, IF NONE, CALL

C ERROR (4MEXT)

C LET K = O$FLAG(REP)

C .........IF O$FLAG EQUALS THE ACT, AC., THIS REP IS ALREADY IN THE

C ...........CLEAN FOR THIS ACTIVITY, SC CC NCLUDING.

C IF (K) FC (IACNC), GO TO 9C

C .........IF REP IS ALREADY IN SCPE OTHER QUEUE, THIS IS AN ERROR.

C IF (K) NE (0), CALL ERROR (4MEXT)

C .........IF O$FLAG = 0, FILE IT INTO THE QUEUE FOR THIS ACTIVITY.

C LET O$FLAG(REP) = IACNC

C ACC Q$ZAI(IACNO) INTO CGS2A(IACAC) SINCF TQSZA(IACNO), ADD I

C LET P#Q$S(IACNO) = AMAX1 (CGSZA(IACNC), P#Q$S(IACNO))

C LET Q$TPE(REP) = TPE

C FILE REP IN ACTG(IACNC)

C RETURN

C END
*IBFTC ASINPR

**ASINPR** (IACNC, ICTY)

C .......THIS ROUTINE IS CALLED WHEN ICC=746C.

C .......ASSIGN PERSONNEL TO AN ACTIVITY.

LET Q = ICTY

LET R = RTIME

LET IACNC = CURAC

IF (IACNC) EQ (IC), CALL ERRCR (4HASPR)

C .......IF THIS IS THE FIRST TIME (DURING THIS REPORT PERIOD) THAT

C .......PERSONNEL OF THIS TYPE HAVE BEEN ASSIGNED TO THIS

C .......ACTIVITY, CREATE A NEW ENTRY AND FILE IT INTO LIST FOR

C .......PERSONNEL NC. IN ANY CASE, TAKE STATISTICS.

FIND FIRST, FOR EACH ENTRY OF LIST(IPERNC), WITH

(AICNC(ENTRY)) EQ (IACNC), WHERE ENTRY, IF NONE, G0 TO 22

LET := WKING(ENTRY)

LET CHKNG(ENTRY) = CHKNG(ENTRY) + W * (R-TWKNG(ENTRY))

LET WKNG(ENTRY) = W + C

GO TO 50

22

CREATE ENTRY

LET ACNO(ENTRY) = IACNC

LET WKNG(ENTRY) = C

FILE ENTRY IN LIST(IPERNC)

50

LET TWKNG(ENTRY) = R

RETLAN

END

*IBFTC READY

**ASINPR READY** (IACNC)

C .......THIS ROUTINE IS CALLED WHEN ICC=7401.

C .......THIS REP IS READY TO BE WORKED ON. REMOVE IT FROM QLEL FOR

C .......THIS ACTIVITY, AND TAKE STATISTICS.

ACC QSAI(IACNC) INTO CGSZA(IACNC) SINCE TCSZA(IACNC), ACC -1

IF (QSAI(IACNC)) LE (-1), CALL ERRCR (4REAC)

FIND FIRST, FOR EACH REP IN ACTC(IACNC), WITH (IACNC(REP)) EQ

(CREP), WHERE REP, IF ACNE, CALL ERRCR (4RECA2)

REMOVE REP FROM ACTC(IACNC)

LET PNGSA(IACNC) = APIA1 (CSZ2I(IACNC)), PNGSA(IACNC)

LET IACSA(IACNC) = TIASI(IACNC) * TIME - QTIME(REP)

LET AQOLT(IACNC) = AQOUT(IACNC) + I

LET QFLAGI(REP) = 0

RETLAN

END
*IBFTC RLFSPR

SUBROUTINE RLFSPR (IPERNC, ICTY, IACAC)
C
..... THIS ROUTINE IS CALLED WHEN IOC=7550.
C
..... PERSONNEL HAVE BEEN RELEASED FROM THIS ACTIVITY. TAKE
C
..... STATISTICS.
C
...... FIND FIRST, FOR EACH ENTRY OF LIST(IPERNC), WITH
C
...... (AGNC(ENTRY)) EC (IACAC), WHERE ENTRY, IF NONE, CALL
C
...... ERRCR (4HRLEP)
C
       LET W = WKG(ENTRY)
C
       LET R = RTIME
C
       LET C=WKG(ENTRY) = C=WKG(ENTRY) + W * (R-TWKG(ENTRY))
C
       LET TWKG(ENTRY) = R
C
       LET WKG(ENTRY) = W - FLCAT(ICTY)
C
RETURN
C
END
C

*IBFTC ASINEQ

SUBROUTINE ASINEQ (IECNC, ICTY)
C
..... THIS ROUTINE IS CALLED WHEN IOC=7370.
C
..... ASSIGN EQUIPMENT.
C
...... ACCUMULATE INUSE(IECNC) INTO CINUS(IECNC) SINCE
C
...... TINUS(IECNC), ADD FLCAT(ICTY)
C
RETURN
C
END
C

*IBFTC RLESEC

SUBROUTINE RLESEC (IECNC, ICTY)
C
..... THIS ROUTINE IS CALLED WHEN IOC=7560.
C
..... RELEASE EQUIPMENT.
C
...... ACCUMULATE INUSE(IECNC) INTO CINUS(IECNC) SINCE
C
...... TINUS(IECNC), ADD FLCAT(ICTY)
C
...... IF (INUSE(IECNC)) LE (1-1.), CALL ERROR (4HRLEQ)
C
RETURN
C
END
*IRFTC FAIL

SUBROUTINE FAIL (IEGCAC)
C ........THIS ROUTINE IS CALLED WHEN ICC=7600.
C ........EQUIPMENT FAILURE.
   LET NFAIL(IEGNC) = NFAIL(IEGAC) + 1
   LET T = ATIME
   LET CDOWN(IEGNC) = CDOWN(IEGAC) + CCWN(IEGNC) *
   (T - TCCWN(IEGNC))
   LET TDOWN(IEGNC) = T
   LET DOWM(IEGNC) = DOWM(IEGAC) + 1.
   RETURN
END

*IRFTC RESTOR

SUBROUTINE RESTOR (IEGCAC)
C ........THIS ROUTINE IS CALLED WHEN ICC=7700.
C ........EQUIPMENT RESTORED.
   LET T = ATIME
   LET CDOWN(IEGNC) = CDOWN(IEGAC) + CCWN(IEGNC) *
   (T - TCCWN(IEGNC))
   LET TDOWN(IEGNC) = T
   LET DOWM(IEGNC) = DOWM(IEGAC) + 1.
   IF (DOWM(IEGNC)) LE (-1.), CALL ERRDEP (4*REST)
   RETURN
END

*IRFTC SPAVL

SUBROUTINE SPAVL (ISPNC)
C ........THIS ROUTINE IS CALLED WHEN ICC=7350.
C ........THERE IS A DEPAND FOR A SPARE. THE SPARE IS AVAILABLE, SO
C ........THE DEPAND IS IMMEDIATELY FILLED.
   LET DMAND(ISPNC) = DMAND(ISPNC) + 1
   LET FILL(ISPNC) = FILL(ISPNC) + 1
   LET MXSP(ISPNC) = AMAX (CSP(ISPNC)+1., MXSP(ISPNC))
   RETURN
END
*IBFIC INSPC

SUBROUTINE INSPC (ISPNC, IREPNC)
C
C       THIS ROUTINE IS CALLED WHEN IDC=7355.
C       THERE IS A DEMAND FOR AN UNAVAILABLE SPARE PART.
C       FIND FIRST, FOR EACH REP IN LCAC(SHPAC(IREPNC)), WITH
C          (IONC(REP)) EQ (CUREP), WHERE REP, IF NONE, CALL
C          ERROR (4HINS)
C       LET S * SFLAG(REP)
C       IF SFLAG = SPARE PART AC., THIS REP IS ALREADY IN QUEUE FOR
C       THIS PART. SC DC ACTMNING.
C       IF (S) EQ (ISPNC), GC TC 50
C       IF REP IS ALREADY IN QUEUE FCR A DIFFERENT PART, CALL ERROR
C       IF (S) NE (0), CALL ERROR (4HINS2)
C       IF SFLAG = 0, PUT IT IN QUEUE FCR THIS SPARE PART NO.
C       LET SFLAG(REP) = ISPNC
C       LET DMAND(IISPNC) = DFAND(IISPAC) + 1
C       ACC GSP(IISPNC) INTG CGSP(IISPNC) SINCE TCSP(IISPNO), ACC 1.
C       LET PNSGP(IISPNC) = OMAX (GSP(IISPNC), MnGSP(IISPNC))
C       LET SPTR=TIME
C       RETURN
C       END

*IBFIC SPRET

SUBROUTINE SPRET (ISPNC)
C
C       THIS ROUTINE IS CALLED WHEN IDC=7000.
C       ALL WE NEED FRC? THIS LABEL RECORD IS THE SPARE PART NO.
C       LET CURSP = ISPNC
C       RETURN
C       END

*IBFIC LVSPC

SUBROUTINE LVSPC (IREPAC, ICREP)
C
C       THIS ROUTINE IS CALLED WHEN IDC=7801.
C       A SPARE PART IS AVAILABLE.
C       IF NO REP WAS WAITING FCR THIS PART, CO NOTHING.
C       IF (IDREP) EQ (0), GC TC 50
C       TAKE THIS REP CFE THE QUEUE FCR THIS SPARE PART.
C       LET ISPNO = CURSP
C       LE) FILL(IISPNC) = FILL(ISPAC) + 1
C       ACC GSP(IISPNC) INTG CGSP(IISPNC) SINCE TCSP(IISPNO), ACC -1.
C       IF (GSP(IISPNC)) LE (-1,), CALL ERROR (4HLVSP)
C       LET NQSP(IISPNC) = AMIN (GSP(IISPNC), NQSP(IISPNC))
C       FInd FIRST, FOR EACH REP IN LLAC(SHPAC(IREPNC)), WITH
C          (IDNO(REP)) EQ (IDREP), WHERE REP, IF NONE, CALL
C          ERROR (4HLVSP)
C       LET TISGS(IISPNC) = TISGS(IISPAC) + TIME - SPTRM(REP)
C       LET SFLAG(REP) = 0
C       RETURN
C       END
*IRFTC OUT*

SUBROUTINE OLT1

C .......GENERATE A *TABLE 1* REPORT FOR EACH DEPOT.

DO TO ICC, FOR EACH SHCP I

LET IRIN = RIN(I)
LET IRCUT = ROUT(I)
LET FROLT = IROLT
LET FRAT = RTSI(I)
LET FRATS = RTSC(I)
LET IRIP = RIP(I)
LET FRIP = IRIP
LET ITIN = TRIN(I) + IRIN
LET TRIN(I) = TRIN
LET ITROLT = TRCLT(I) + IRCUT
LET TRCLT(I) = IRCUT
LET RTCLT = ITRCUT
LET TOTAL = IRIN + IRCUT + 1
CALL STODEV (TOTAL, RIPS(I), RTSC(I), *AVN, *STDN
IF (IROLT) EQ (0), LET PART(I) = 0.
CALL STODEV (FRAT, FTAT, FRAT, *AVT, *STDVT)
LET FTATS = TRSI(I) + FTAT
LET ARS(I) = FTATS
LET FRATSQ = RTSC(I) + FRAT
LET RTSC(I) = FTAT
CALL STODEV (FTAT, FTAT, FTAT, *AVT, *TNVT)
IF (ITROLT) EQ (0), GG TC 60
LET TNRT(I) = APAX1 (MXRT(I), TNRT(I))
LET FNMRT = AMAX1 (MNRT(I), TNRT(I))
LET TNRT(I) = FNRT
GO TO EC

6C LET FMNRT = 0.

BC CALL TAB1 (I, AVN, STDVN, AVT, STDVT, TAVT, FTMNRT, YSTDVT)

C .......RESET VARIABLES FOR NEXT REPORT PER ICC.

LET AIN(I) = C
LET ROLT(I) = 0
LET MAXR(I) = IRIP
LET MINR(I) = IRIP
LET RIPS(I) = FRIP
LET RIPSQ(I) = FRIP * FRIP
LET ATS(I) = 0.
LET RTS(I) = 0.
LET MXRT(I) = 0.
LET MNRT(I) = 10000.

ICC LOOP
RETURN
END
*IBFTC STDDEV

SUBROUTINE STDDEV (TCTAL, SUP, SUPSG, AVG, STDV)
C
C .....ROUTINE TO COMPUTE A MEAN AND STANDARD DEVIATION.
C
IF (TOTAL) LE (C.), GO TO 50
LET AVG = SUM / TCTAL
LET STDV = SQRT((APX1(SUPSG/TCTAL - AVG*AVG, 0.))
GO TO 100
50 LET AVG = C.
LET STDV = 0.
100 RETURN
END
*IBFIC TABI
REPORT TABI (I, AVN, STDVA, AVT, STCVT, TAVT, FTMNRT, TSTDVT)

SYSTEM ARRIVALS, DEP.
AND REPAIR

REPARABLES ENTERING DEPOT THIS:
SERVICEABLES DEPARTING DEPOT THI

REPARABLES IN PROCESS

REPAIR TIME
THIS PERIOD

TC GATE

END

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1
*IBFTC OUT2

SLBROLTIME OUT2

C GENERATE TABLE 2.

LET C = CURPD
DO TO 5C, FOR EACH ACTIV I
ACCLMLLATE GSZA(I) IATC CCSZA(I) SINCE TCSZA(I)
LET AVCSZA(I) = CCSZA(I) / C
IF (AQOUT(I)) EC (G), GC TC 5C
LET AVTAQ(I) = DECHR(TIACS(I)) / FLCAT(AQOUT(I))

5C
CALL TAB2
DO TO ICC, FOR EACH ACTIV I
LET CQSZA(I) = G,
LET TIACS(I) = 9.
LET F4SZA = QSZA(I)
LET PM5QA(I) = FCSZA
LET MNQ5A(I) = FCSZA
LET AQOUT(I) = 0

1CC
LCOP
RETLRN
END

*IBFTC TAB2

REPORT TAB2

ACTIVITY QU.
FOR PERIOD.
NO. OF REPS

ACT.
NO.  AVG  MAX

1  AVSQA(I)  MXSQA

FOR EACH ACTIV I, WITH (MXOSA(I)) GR (O,)

LEING FACTORS 1
LIST COMPLETED 12
IN QLELE

AVG QLELE TIME 1

MIN (IN WORK-HOURS)

1) MNOSA(I) AVTAQ(I)

FAD
*IBFTC OL3

SUBROUTINE OL3

C ***** BEGIN BY WRITING THE HEADING.
C
CALL T33HED
LET S = STIME
DO TO 1CC, FOR EACH PTYPE I, WITH (CTYS(I)) OR (C)
LET SUM = 0.
1C
DO TO 20, FOR EACH ENTRY CF LIST(I)
C
*BRING 'CBRNG(ENTRY)' UP TO DATE BEFORE ADDING IT INTO SUM.*
LET C = CBRNG(ENTRY) + WRNIG(ENTRY) * (S-WRNIG(ENTRY))
LET CBRNG(ENTRY) = C
LET SUM = SUM + C
2C
REPEAT 1C
LET UTIL = SUM / CQTY(I)
CALL T33LIN (I, UTIL)
LET CQTY(I) = 0.

C
*.EMPTY OUT EACH 'LIST' SET.*
3C
DO TO 50, FOR EACH ENTRY CF LIST(I)
REMOVE ENTRY FROM LIST(I)
DESTROY ENTRY
5C
REPEAT 3C
1C
LOCUP
RETURN
END

*IBFTC T33HED

REPORT T33HED

PERS  UTIL  PAN-  PAN-  MAN-  MAN-  PAN-
TYPE QTY FACT NO HOURS NC HOURS NC HOURS NO HOURS

END

Utilization

PAN-  PAN-  PAN-  PAN-
NO HOURS NC HOURS NC HOURS NC HOURS NC HOURS
END
*IBFTC TB3LIN

REPORT TB3LIN (1, UTILI)
X    000  00  0.00
X    I CTLYS(I) UTIL
    9 FOR EACH ENTRY CF LIST(I), WITH (FLIST(I)) NE (0)
    X    00  000.00  000.00  000.00  000.00  000.00
X    9(AHCN(ENTRY),DECW(CWNG(ENTRY)))
    END
X
X    00  000.00  000.00  000.00  000.00  000.00
    END

*IBFTC OUT4

SUBROUTINE OUT4
C        GENERATE TABLE 4.
CALL TB4MED
    LET CUR = CURP
    DO TO 5C, FOR EACH ETYPE I, WITH (CTYE(I)) GE (0)
    LET TOTAL = CUR * FLCAT(CTYE(I))
    LET C = CINUS(I)
    LET UTIL = C / TOTAL
    LET FIDLE = TOTAL - C - CODAW(I)
    CALL TB4LIN (I, FIDLE, UTIL)
    LET NFAIL(I) = 0
    LET CODAW(I) = 0
    LET CINLS(I) = 0.
  5C LCP
RETURN
END

*IBFTC TB4MED

REPORT TB4MED
X
X    EQUIPMENT  TYPE  QUANTITY  TIME  IDLE  DOWN
X    EQUIPMENT  TYPE  IN USE  TIME  TIME  F
END

UTILIZATION
  G. OF UTILIZATION ACTIVITY INC.
  ILURES  FACTOR  WHERE USED
END
*IBFTC TB4LIN
REPORT TB4LIN (I, FIDLE, UTIL)
12 FOR EACH ITEM OF SET(I)
X
END

*IBFTC OLTS
SUBROUTINE OLTS
C GENERATE TABLE 5.
LET C = CURPD
DO TO 50, FOR EACH SPTYP I
ACCLMULATE QSP(I) INTO CGSP(I) SINCE TCSP(I)
LET AVGSP(I) = CGSP(I) / C
IF (FILL(I)) EQ (.O), GC TO 50
LET AVTSQ(I) = DECHAMISQ(I) / FLCAT(FILL(I))
SC LOOP
CALL TABS
DO TO 100, FOR EACH SPTYP I
LET CGSP(I) = O.
LET TISGS(I) = C.
LET FQSP = QSP(I)
LET MXGSP(I) = FCSP
LET MNGSP(I) = FCSP
LET OMANN(I) = C
LET FILL(I) = 0
CC LOOP
RETURN
END
QUEUEING FACTORS BY REPAIRABLE QUEUE LENGTH

COMPONENT
TYPE        QUANTITY         DEMANDS
            THIS PERIOD

FOR EACH SPTYP I, WITH (GTYSPI) OR (0)

COMPONENT SPARES TYPE
AND TIPES BY COMPONENTS
WE. OF KEPs IN QUEUE

AVG QUEUE TYPE
AVG QUEUE TIME
(IA BCRR-HCURS)

END

SUBROUTINE ERROR (NAME)
CALL ERREPT (NAME)
CALL SNAP (4HERCR)
STOP
END

SUBROUTINE ERROR (NAME)
JCB TERMINATED AT TIME 00.000 BECAUSE OF ERROR IN SUBROUTINE * *
(ADDITION)
END

SUBROUTINE SNAP (LABEL)
CALL SNAP (LABEL)
CALL SNAPZ (T), FOR EACH SNAP I
RETURN
END
* IBFTC SNP2
  REPORT SNP2 (ISHCP)
  14 FOR EACH REP IN LOAD (ISHCP)
  REPS IN DEPUT 0

  IDNO 14(IDNC(REP))
  BTIME 14(BTIME(REP))
  QTIME 14(QTIME(REP))
  CFLAG 14(QFLAG(REP))

END

END DEPCT CAPABILITY OUTPUT PROGRAM
REFERENCES


### PLANET: PART V--REPORTS AND ANALYSIS LIBRARY

**Authors:** Voosen, B. J., S. Glaserman, R. J. Young, Judy Judd

**Report Date:** January 1969

**Contract or Grant No.:** F44620-67-C-0045

**Originator's Report No.:** RM-4663 PR

**DDC-1 Availability/Limitation Notices:**

**Abstract:** A user's and programmer's manual for the 12 programs comprising the Reports and Analysis Library of PLANET (Planned Logistics Analysis and Evaluation Technique), a logistics prediction and estimating tool designed to help the manager of a system understand its operation and to find a rationale for allocating resources efficiently. PLANET consists of four computer models that simulate Air Force logistics systems in a single or multibase environment. Whether the models are used singly or in various configurations, the output will be a tape listing of selected variables accumulated during the simulation. From this tape the desired reports are generated by using the Library of programs. The manager can select those programs best suited for analysis of his particular problem. Although PLANET is programmed in SIMSCRIPT, the user need not be a skilled programmer to conduct a simulation. Step-by-step instructions are included to permit the manager to assemble the data in a form acceptable to the simulations. Part 1 of the Memorandum contains a brief description of each of the reports and the SIMSCRIPT instructions needed to initialize any of the report programs. Part 2 is the library of programs, including the initialization requirements, a program description oriented to the skilled SIMSCRIPT programmer, and a listing of the SIMSCRIPT SOURCE program.

**Keywords:**
- PLANET (Maintenance Simulator Model)
- Bases
- Depots
- Maintenance
- Logistics
- Resource Management
- Weapon systems
- Computer simulation
- Computer programs