20040817 026

Special Report D-84-2

LOGAM TECHNICAL/PROGRAMMER MANUAL VOLUME III (Revised)

Systems Analysis Division
Systems Analysis and Evaluation Office
US Army Missile Command
Redstone Arsenal, Alabama 35898

BALLISTIC MISSILE

February 1985

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898

Approved for Public Release; Distribution Unlimited

PLEASE RETURN TO:

BMD TECHNICAL INFORMATION CENTER
BALLISTIC MISSILE DEFENSE ORGANIZATION
7100 DEFENSE PENTAGON
WASHINGTON D.C. 20301-7100

DTIC

RM 1021, 1 NOV 81 PREVIOUS EDITION MAY BE USED

U6703

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Special Report D-84-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) LOGAM Technical/Programmer Manual, Volume III		5. TYPE OF REPORT & PERIOD COVERED Revised 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) DAAH01-82-B-A508
9. PERFORMING ORGANIZATION NAME AND ADDRESS John M. Cockerham & Assoc., Inc. 301 Randolph Avenue S.E. Huntsville, AL		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AMCMS 63310.0740012
11. CONTROLLING OFFICE NAME AND ADDRESS Commander US Army Missile Command ATTN: DRSMI-DS, Redstone Arsenal, AL 35898		12. REPORT DATE February 1985 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Logistics	Maintenance	Systems Analysis
Cost Analysis	Test Equipment	Availability
Operational Support Cost	Manpower	
Supply	Maintenance Policy Evaluation	
Provisioning	Design-Logistics Interface	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This manual describes the Logistic Analysis Model (LOGAM) and its use for evaluating logistic operations as applied to US Army materiel systems. The objective is to develop methodology for generating quantitative data for analysis of activities necessary to equip, operate, maintain, and support a materiel system. LOGAM is a deterministic model, analytical in design through its sensitivity feature, and highly versatile in its ability to evaluate many alternatives rapidly and inexpensively. Through the sensitivity median,

ABSTRACT (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Continued)
support alternatives are tested for evaluating life cycle costs and for recommending optimum repair levels; repair versus discard at failure; manpower, provisioning and test equipment requirements; table of organization and equipment adjustment or development; and other operational elements by quantities and costs.

FORWARD

The Logistic Analysis Model LOGAM Technical/Programmer Manual Volume III (Revised) was written under Contract DAAH01-82-B-A508. The work was performed with the US Army Missile Command under the general technical cognizance of Mr. Raymon S. Dotson, Systems Analysis Division, Systems Analysis and Evaluation Office, US Army Missile Command, Redstone Arsenal, Alabama. The program also produced two companion documents entitled LOGAM Users Manual Volume II and LOGAM Executive Summary Volume I.

TABLE OF CONTENTS

SECTION	TITLE	PAGE
1.0	Introduction	1-1
2.0	Job Control Language	2-1
3.0	Special Features and Mathematical Formulations	
3.1	Maintenance Policy Selection	3.1-1
3.2	Provisioning Rules	3.2-1
3.3	Sensitivity Testing	3.3-1
3.4	Availability Accumulator	3.4-1
3.5	LOGAM/AR570-2 Maintenance Data I/F	3.5-1
3.6	TOE Operation and Support	3.6-1
4.0	Detailed Program Description	
4.1	Program LOGAM	4.1-1
4.2	LOGAM Function Routines	4.2-1
4.2.1	BLKDAT	4.2-2
4.2.2	AB	4.2-7
4.2.3	D	4.2-8
4.3	LOGAM Availability Routines	4.3-1
4.3.1	AVAIL	4.3-2
4.3.2	DEF	4.3-10
4.3.3	SPOL	4.3-13
4.4	LOGAM Maintenance Flow	4.4.1
4.4.1	RSFRCT	4.4-2
4.4.2	MFLOW	4.4-5

TABLE OF CONTENTS CONTINUED

SECTION	TITLE	PAGE
4.5	LOGAM Provisioning Routines	4.5-1
4.5.1	BASIC	4.5-2
4.5.2	IOL	4.5-9
4.5.3	MPROV	4.5-10
4.5.4	REPLC	4.5-19
4.5.5	STKDS	4.5-21
4.5.6	REORDR	4.5-26
4.6	LOGAM Test Equipment & Manpower Demands	4.6-1
4.6.1	TEMWF	4.6-2
4.6.2	PERSON	4.6-15
4.7	Subroutine LGCST	4.7-1
4.8	TOE Operation & Support Costs	4.8-1
4.8.1	OPER	4.8-2
4.8.2	ONE	4.8-6
4.8.3	TWO	4.8-10
4.8.4	THREE	4.8-15
4.8.5	FOUR	4.8-17
4.8.6	FIVE	4.8-19
4.8.7	SIX	4.8-21
4.8.8	SEVEN	4.8-25
4.8.9	SUPTMH	4.8-27
4.8.10	SGAPER	4.8-33
4.8.11	TOEPER	4.8-38

TABLE OF CONTENTS CONTINUED

SECTION	TITLE	PAGE
4.9	LOGAM Input Routines	4.9-1
	4.9.1.FINLE	4.9-2
	4.9.2 READFL	4.9-10
	4.9.3 IMPAK	4.9-10
	4.9.4 PARSV	4.9-10
	4.9.5 DCODR	4.9-11
	4.9.6 VALUE	4.9-12
	4.9.7 MXINP	4.9-12
	4.9.8 COLINE	4.9-12
4.10	LOGAM Output Routines	4.10-1
	4.10.1 INPOP	4.10-2
	4.10.2 IDVOP	4.10-5
	4.10.3 IPVOP	4.10-13
	4.10.4 TTLOP	4.10-16
	4.10.5 EIGHT	4.10-18
	4.10.6 SUMRY	4.10-27
	4.10.7 PAGE	4.10-35
	4.10.8 SENSIT	4.10-36
5.0	Program LOGAM Input Definitions	
5.1	LOGAM Inputs	5-1
5.2	NAMELIST Inputs	5.2-1
	5.2.1 NAMELIST/L/ Variable Description	5.2-31

TABLE OF CONTENTS CONTINUED

SECTION	TITLE	PAGE
6.0	Program LCGAM Output Definitions	
6.1	LOGAM Provisioning Outputs	6-1
6.2	Individual IRU Outputs	6-4
6.3	Summarized LRU Case Outputs	6-10
6.4	Cost Totals Outputs	6-15
6.4.1	Case Totals	6.4-1
6.4.2	Grand Total Outputs	6.4-3
6.5	Maintenance Manpower Outputs	6-18
6.6	System Support Costs Outputs	6.6-1
6.6.1	System Maintenance Case Totals	6.6-1
6.6.2	System O&S Case Totals	6.6-3
6.6.3	System Maintenance Grand Totals	6.6-4
6.7	Sensitivity Outputs	6.7-1
6.7.1	Sensitivity With Inhibited Print	6.7-1
6.7.2	Sensitivity With LRU Case Outputs	6.7-2
6.8	Output of LOGAM Inputs ($L)	6.8-1
7.0	Sample Inputs	7-1
8.0	Sample Outputs	8-1
9.0	LOGAM Flowchart	9.1

FIGURES

FIGURE	TITLE	PAGE
1	Maintenance Policy Matrix.....	3.1-2
2	Equipment Level (E) - Maintenance Flow.....	3.2-3
3	Direct Support (O) - Maintenance Flow.....	3.2-4
4	General Support (I) - Maintenance Flow.....	3.2-5
5	Depot Level (D) - Maintenance Flow.....	3.2-6
6	Sample Sensitivity Input.....	3.3-8

TABLES

TABLE	TITLE	PAGE
1	Policy "G" Factors	3.1-3
2	Sensitivity Input Listing	3.3-4
3	SGA For MOS 35B	3.5-2
4	TOE Line Items	3.6-3
5	TOE Field Inputs and Mnemonics	3.6-4
6	TOE Mnemonics Explanation and Use	3.6-5

SECTION 1

INTRODUCTION

SECTION 1 - INTRODUCTION

The LOGistic Analysis Model (LOGAM) provides a tool for the evaluation of alternate support postures for Army equipment. The program is structured to perform logistics analyses in maintenance support situations where the emphasis is on the support channels required for a diversity of operating equipments. LOGAM can be used to evaluate alternate maintenance postures on the basis of Life Cycle Costs (LCC).

Although operational and maintenance costs are emphasized, the model accounts for development and investment costs of prime and test equipment, spares and facilities. In addition to LOGAM maintenance costs, LOGAM has the capability to evaluate theater O&M costs from a Table of Organization and Equipment (TOE). TOE maintenance personnel costs can be evaluated from personnel data input by the NAMELIST/L/ data array (T) and/or on the AR570-2 MACRIT data file. Costs are printed at the theater level (case total) using both the LOGAM and DA PAM 11-4 format.

LOGAM maintenance analysis is based on a four tier support system, i.e., Organization, Direct Support, General Support and Depot. The test equipment and manpower demands are determined by the flow of material at a support echelon generated by the maintenance incident rate, mean time between maintenance actions, the "on" time fraction, scrap rate, false "no go" rate, and attrition. The maintenance demands and spares requirements at a support echelon are a result of the maintenance policy (s) used. LOGAM has 20 different maintenance policies to select from. The user can elect to choose any one of these policies or any combination of policies.

This document includes a section that describes the LOGAM features such as the use of maintenance policies and a section that details the lines of FORTRAN code. All input and output variables are described in their respective sections and there are sections that include sample input and output cases. A functional level flowchart of the main program is included to illustrate the LOGAM logic flow.

The page numbering system used in this document was selected in order to simplify the insertion and deletion of program changes as LOGAM is continued to be enhanced. Each page number will have its section number as the first digit. In some sections the second level of the section will have its number as the second digit. For example in section 4, section 4.1 would begin with page number 4.1-1 and section 4.2 would start with page 4.2-1. Using this system the entire document doesn't have to be resequenced if pages are either inserted or deleted.

A list of the remaining sections of this document and a brief description of each are given below.

Section 2 - The control card section that lists the job control language (JCL) needed to execute LOGAM on MICOM's CYBER 74 computer using the NOS operating system.

Section 3 - The LOGAM features which lists the primary formulas used for evaluating the maintenance flows and provisioning. Included in this section are discussions on the system availability accumulation, sensitivity analysis, MACRIT data inputs and a TOE O&S cost analysis.

Section 4 - The detailed program description which documents the purpose of each statement or groups of statements. Each subroutine is documented individually but the subroutines are grouped as to their functional use. Each functional grouping will have its own page numbering scheme. There are 10 functional groups beginning with the main program (4.1) and ending with the output routines (4.10). The descriptions in this section should be helpful not only in modifying the code but also to grasp the use of the model.

Section 5 - The input definitions that describe the "one-time" system level inputs, the individual LRU case leader cards and each variable of the NAMELIST/L/ individual LRU data.

Section 6 - The output definitions that describe the individual LRU case outputs, case and grand totals in a LOGAM format, personnel requirements, and system operations and support costs in the DA PAM 11-4 format.

Section 7 - The sample inputs which is a listing of a case run.

Section 8 - The sample outputs that are annotated to match the output descriptions of section 6.

Section 9 - The functional level flowchart that illustrates the overall logic flow of the program.

Appendix A - The program listing which is a bound computer output that is a separate part of this documentation.

With a knowledge of logistics and an understanding of the FORTRAN language there should be sufficient information contained in this document to provide an understanding of the program's logic and use. Each section of this document provides a specific type of information; therefore, the sections should be used in conjunction with each other to get a complete view of the program.

SECTION 2

JOB CONTROL LANGUAGE

SECTION 2 - JOB CONTROL LANGUAGE

This section discusses the Job Control Language (JCL) required to use the LOGAM program. The JCL here is for a program library that is cataloged on a permanent disk file. The following statements describe the JCL for updating, compiling, and executing the LOGAM program on MICOM's CDC Cyber 74 Computer system.

JOBNO,T77,CM150000.
This statement includes the job number, execution time in octal seconds, and the central memory size, respectively. The job number is a fictitious one, a legitimate one would be provided by computer operations personnel. The time limit is set here because program execution will not complete within the system default time.

ACCOUNT, UN=username, PW=password, AC=accounts, OP=A2, J=0001
This statement is the account information that includes the users name, password, and account number. The letters in this statement that are upper case will always be input. The lower case letters is the information provided by the user. The password and account numbers are provided to the user by computer operations personnel.

LIMIT, 500.
This statement reserves 500 blocks of disk space to store the LOGAM outputs.

ATTACH,OLDPL,Filename,ID=idname, CY=N.
This statement attaches the LOGAM program library from a permanent file into the local file OLDPL for updating later. The information in lower case letters is provided by the user where "filename" must be obtained from computer operations personnel.

UPDATE,F.
This statement performs a full update of the LOGAM program library stored on file OLDPL and provides a COMPILE file for later compilation.

RFL,150000.
This statement sets the central memory field length to 150K octal locations to provide sufficient memory to perform the compilation of the LOGAM FORTRAN code.

FTN,I=COMPILE,R=2.
This statement uses the FORTRAN compiler (FTN) to compile the output file from the UPDATE command to generate an executable program file (LGO).

LGO.
This statement loads and executes the relocatable program file (LGO).

SECTION 3

LOGAM FEATURES

SECTION 3 - LOGAM FEATURES (MAINTENANCE POLICIES)

3.1 Maintenance Policy Selection

The logistic and maintenance support system possibilities which may be considered comprise twenty basic maintenance policies with four possible levels of inventory support for each. The 20 basic maintenance policies are summarized in Figure 1. LOGAM additionally allows the analyst to split maintenance policy and stock location - this leads to a number of combinations which are essentially unlimited.

3.1.1 Policy "G" Factors - The LOGAM deployment matrix shows four possible levels of maintenance support: at the Equipment proper, at a Direct Support level, at a General Support level, and at a Depot level. The model additionally assumes that faults are identified in accordance with the LRU removal rate E at the equipment level. LOGAM also provides three levels of maintenance support capability: unit checkout, fault isolation of the unit to a faulty module, and module test and repair. It provides three levels of logistic discard: unit, module, and part.

The maintenance levels at which work is performed and the test equipment, test, and repair manpower locations are specified by "G" factors. These are the same "G" terms illustrated on Figure 1. The same factors are used to define the flow of maintenance work in the postulated system or given deployment. These factors are described in Table 1.

The matrix of the "G" factors as structured to form the maintenance policies that are built into the LOGAM formulation is shown in Figure 1. This matrix identifies the support posture options available within the LOGAM model. These alternatives are designated GA through GT in the upper part of the matrix. Twenty alternatives are available and as discussed previously, they can be combined so that a percentage of work is accomplished by one policy with the balance being accommodated by other policies selected from the matrix. In the matrix, X indicates that the options listed around the perimeter of the chart apply for the block in which the X is located. A blank in a block indicates that there is no applicable action taking place.

For example, the X in the fourth column from the left in the fourth row from the top is to be interpreted in the following way:

> (Start at the left-hand edge of the chart)
>
> For the maintenance policy designated "GD," test equipment will be located at DIRECT SUPPORT. Test equipment at Direct Support can isolate faults to the level of the failed MODULE. Repair will be accomplished by discarding and replacing the failed MODULE.

FIGURE 1.
MAINTENANCE POLICY MATRIX

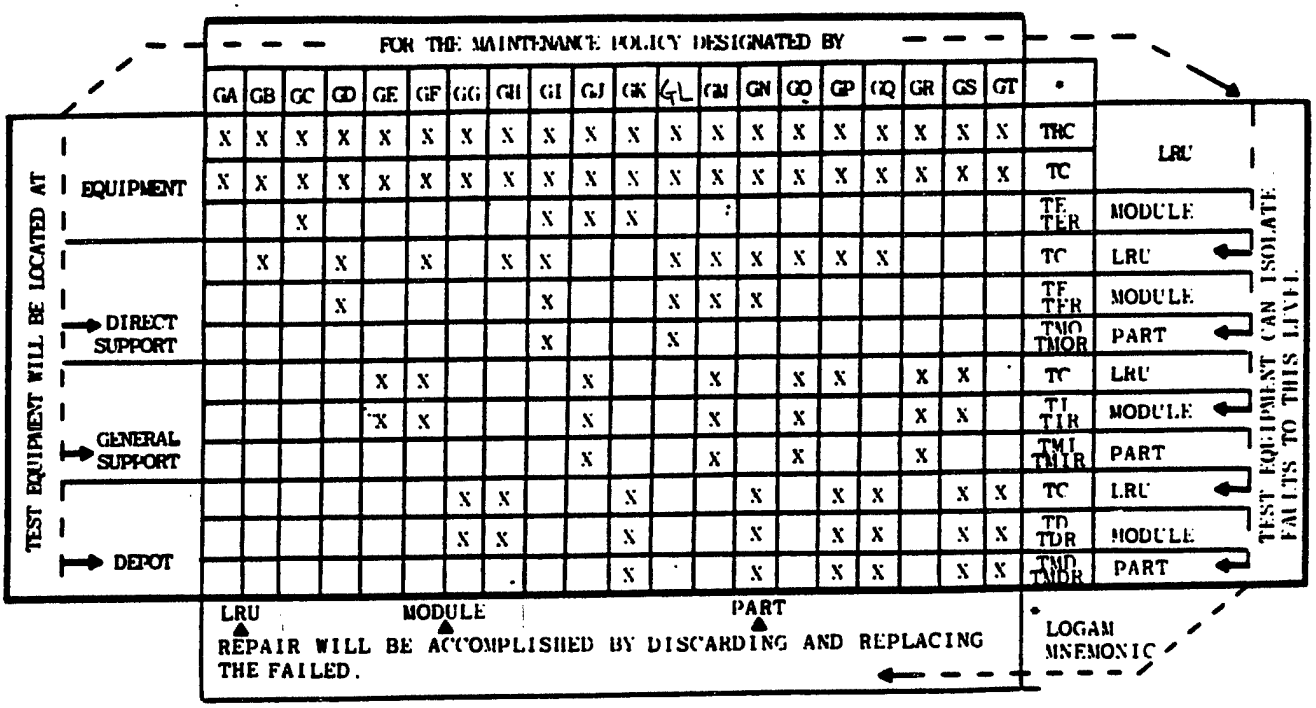


TABLE 1
POLICY "G" FACTORS

GA Specifies a policy of discard at failure. There are no maintenance support activities. All failures, false "no go" indications, and attrition rate inputs result in LRU discard. Only LRUs are stocked in the supply system. There is no demand for modules or parts.

GB Similar to GA but here is a provision to detect false "no go's" at Direct Support and only failed and attrited LRUs are discarded. There is no demand for module or part stock.

GC Specifies LRU repair at equipment level by removing and replacing a defective module. The defective module is discarded.

GD Specifies LRU repair at Direct Support by removing and replacing a defective module. The defective module is discarded.

GE Specifies LRU repair at General Support by removing and replacing a defective module. The defective module is discarded.

GF Specifies LRU repair at General Support with checkout performed at Direct Support to remove false "no go" LRUs before sending the work to General Support. LRU repair is by removal and replacement of a defective module and the defective module is discarded.

GG Specifies LRU repair at Depot. Defective modules are discarded.

GH Specifies LRU repair at Depot preceded by a checkout at Direct Support to screen false "no go's". Defective modules are discarded.

GI Specifies LRU repair at equipment level and module repair at Direct Support.

GJ Specifies LRU repair at equipment level and module repair at General Support.

GK Specifies LRU repair at equipment level and module repair at the Depot.

GL Specifies LRU and module repair at Direct Support.

TABLE 1 CONTINUED

GM Specifies LRU repair at Direct Support and module repair at General Support.

GN Specifies LRU repair at Direct Support and module repair at Depot.

GO Specifies checkout to screen false "no go's" at Direct Support followed by LRU and module repair at General Support.

GP Specifies checkout to screen false "no go's" at Direct Support followed by LRU repaired General Support and module repair at Depot.

GQ Specific LRU checkout to screen false "no go's" at Direct Support followed by LRU and module repair at Depot.

GR Specifies LRU and module repair at General Support.

GS Specifies LRU repair at General Support and module repair at Depot.

GT Specifies LRU and module repair at Depot.

SECTION 3 - LOGAM FEATURES (MAINTENANCE POLICIES)

The specification of a maintenance concept, input by the GA through GT fractions, may be mixed in any proportion summing to unity to represent the flow of work demand. For example, if:

GL = 0.6, GR = 0.25, GT = 0.15

60% of the LRU removals would be sent to Direct Support for repair, 25% would be sent to General Support for repair, and the remaining 15% would go to Depot. If FUO = .8 then eighty percent of the total LRUs arriving at Direct Support would be repaired. The other twenty percent would be scrapped since there is no general support in policy GL. Similarly FUI = .8 and FUD = .8 would act for General Support and Depot.

Care should be taken in developing a support concept through the selection of multiple G fractions since the workload flow between maintenance levels may be different even though the concepts developed have the same support levels. For example, lets assume one wants to model a typical 4 level maintenance concept. This could be done by combining either policies GL, GM and GN or GD, GE and GG. Assuming that GL=GD=.6 and GM=GE=.25 and GN=GG=.15 the difference can be highlighted. In the first grouping of policies (GL, GM, and GN) the flow of LRUs from Equipment (E) to Direct Support (DS) is equal to (F+FNG)*(GL+GM+GN)=(F+FNG)*1.0. In the second grouping of policies (GD, GE, and GG) the flow of LRUs from E to DS is equal to (F+FNG)*(GD)=(F+FNG)*.6. Thus the flow of LRUs in the second grouping of policies is .4 less than the first. This is due to the fact that the grouping policies GL, GM, and GN all have a DS support level in the policy matrix while in the grouping of policies, GD, GE, and GG only the GD policy has a DS support level in the policy matrix. Thus it is critical to understand how the workload flows are structured for a combination of G policies.

SECTION 3 - LOGAM FEATURES (PROVISIONING RULES)

3.2 Provisioning Rules

LOGAM has three rules for determining initial provisions. The method of determining the type of rules to use is dictated by the input value for AYZP. A negative value for AYZP will use a predetermined (input) method and the adequacy of the provisions are left for the user to determine. A zero value of AYZP will use the LOGAM supply rule and a positive value will use the maintenance rule.

3.2.1 Predetermined Supply Levels - In this case the analyst inputs the quantity of LRUs (QTE, QTO, QTI, QTD), modules (QTME, QTMO, QTMI, QTMD), and parts (QTPO, QTPI, QTPD). The effect that these values have on availability is computed and modified if necessary.

3.2.2 LOGAM Supply Rules - For LRUs QTE is examined first. This represents the modified input value of LRU stock at "E" (equipment) at this point of the program. If its value is greater than 0.5, it is assumed that the input value is valid and the value is the number of LRUs to be stocked. However, if QTE is less than 0.5, it is assumed that this may not be the actual value wanted. H(1), the "stock at E level", is examined to determine if stock is intended to be placed at "E" (it is if H(1)=1). If the answer is yes, the IOL subroutine is called to compute this new QTE value. If no stock is intended to be placed at "E", the QTE value is ignored and in either case the needs for LRU stock at Direct Support, General Support, and Depot are successively explored and, where required, calculated by calling IOL.

3.2.2.1 Module and Part Stock - In the LOGAM supply rules module and part requirements are similarly examined and calculated. The logic is similar to that used in the LRU logic except that the "H" flags do not exist for modules and parts and there are only three supply levels for parts. Parts are not stocked at the Equipment level.

3.2.2.2 IOL Operation - The IOL subroutine is called with the safety stock coefficient (CKK), the number of supply locations (XD), the quantity tied up in scrap replenishment (BQU), the quantity tied up in float repairs (BQF) (this is zero for parts), and the round-up factor (Z). The IOL subroutine adds the scrap and repair quantities and multiplies by the number of supply locations (XD). The output (BQT) is set at zero and routine checks on XD and QUF are made for obvious errors in which case the subroutine returns with the output stock quantity (BQT) at zero.

The safety stock increment (BSQ, the standard deviation from the mean or expected value of stock tied up) is computed as the

square root of the mean (per the Poisson distribution).

The amount of safety stock is computed by multiplying the safety stock increment (BSQ) by the coefficient (CKK), an input quantity. This is given the name BQS.

The total stock is computed by adding the pipeline stock (BQU), the repair float stock (BQF), and the safety stock (BQS). It is given the name QUFS and is permitted to have a minimum value of zero.

The total stock (QUFS) is divided by XD to obtain the amount of stock per location, the round-up quantity (Z) is added, and the result is integerized to obtain an integer quantity per location. This integer quantity is multiplied by the number of locations (XD) to obtain total stock quantity (BQT) of LRUs, modules, or parts. Because of the rounding-up process, the resulting stock quantity always exceeds the expected value of demand for stock.

A seperate call to Subroutine IOL is made for LRUs, modules and parts at each supply echelon. The LOGAM mnemonics that match the IOL arguments are listed below.

XD	Number of supply points (EDS, ODS, DIS, DDS)
CKK	Safety stock factor for LRUs (CKUE, CKUO, CKUI, CKUD) Safety stock factor for modules (CKME, CKMO,CKMI, CKMD) Safety stock factor for parts (CKPO, CKPI, CKPD)
BQU	Pipeline scrap for LRUs (QUE, QUO, QUI, QUD) Pipeline scrap for modules (QME, QMO, QMI, QMD) Pipeline scrap for parts (QPO, QPI, QPD)
BQF	Pipeline repair float for LRUs (QFE, QFO, QFI, QFD) Pipeline repair float for modules (QFME, QFMO, QFMI, QFMD)
BQT	LRU provisions (QTE, QTO, QTI, QTD) Module provisions (QTME, QTMO, QTMI, QTMD) Part provisions (QTPO, QTPI, QTPD)

3.2.3 LOGAM Maintenance Rules

LOGAM Maintenance Rules allow for determination of initial provisioning quantities of LRU's, modules, and parts based upon time delays due to the maintenance concept for concepts selected. Figures 2 through 5 depict the logic used to generate the repaired, screened, evacuated and scrap flows associated with the maintenance concepts.

FIGURE 2
EQUIPMENT LEVEL (E) - MAINTENANCE FLOW

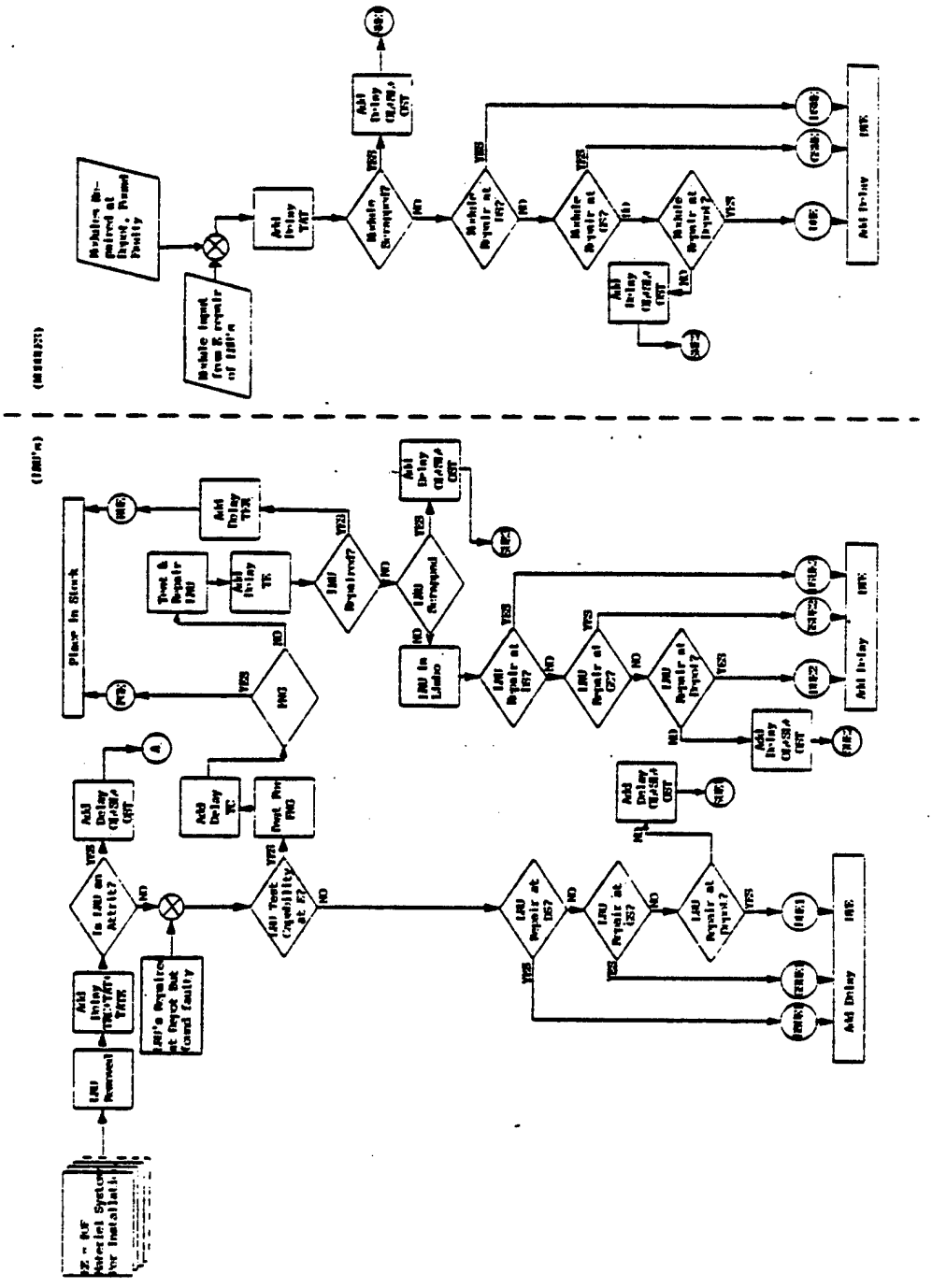


FIGURE 3
DIRECT SUPPORT LEVEL (O) - MAINTENANCE FLOW

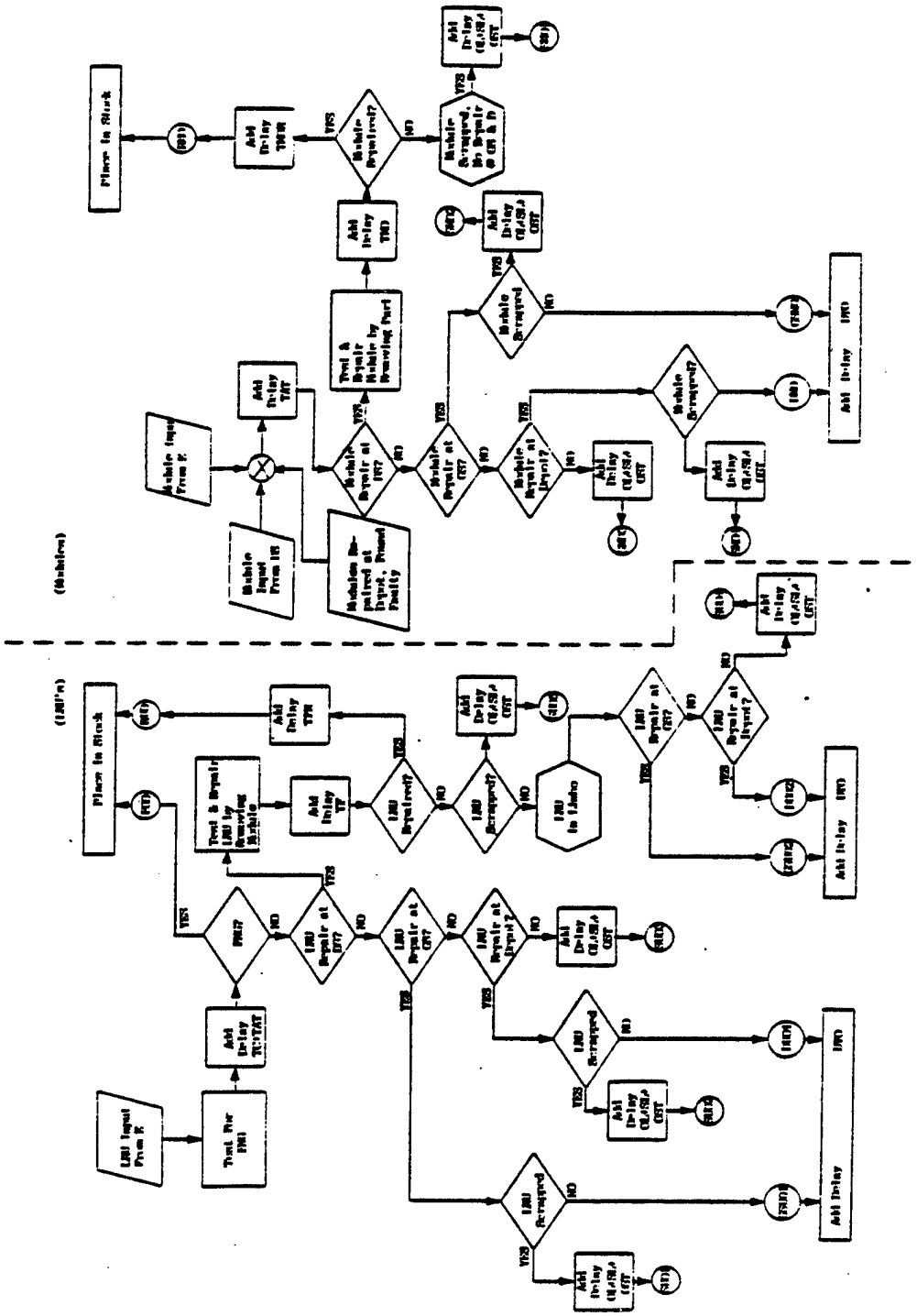


FIGURE 4
GENERAL SUPPORT LEVEL (I) - MAINTENANCE FLOW

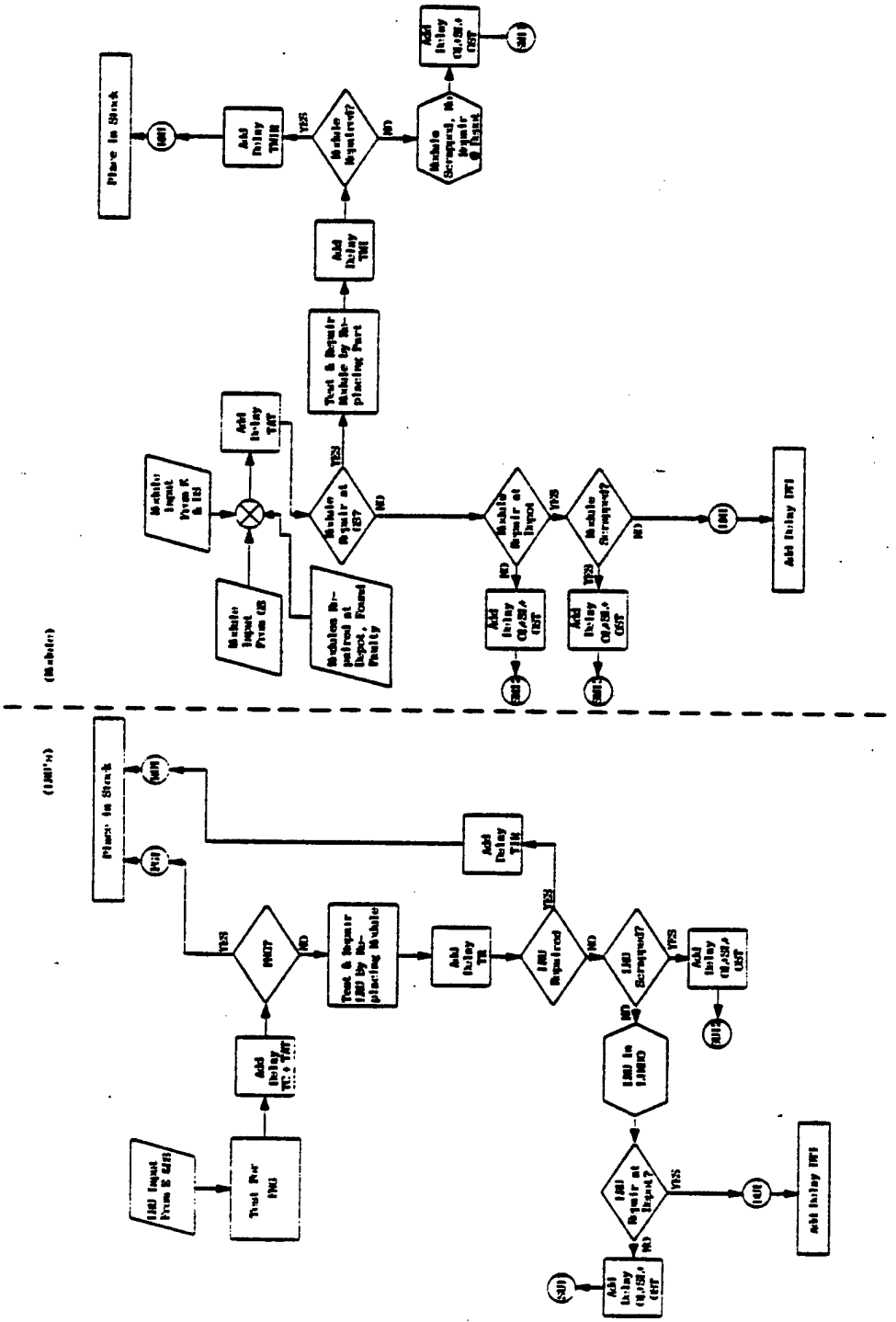
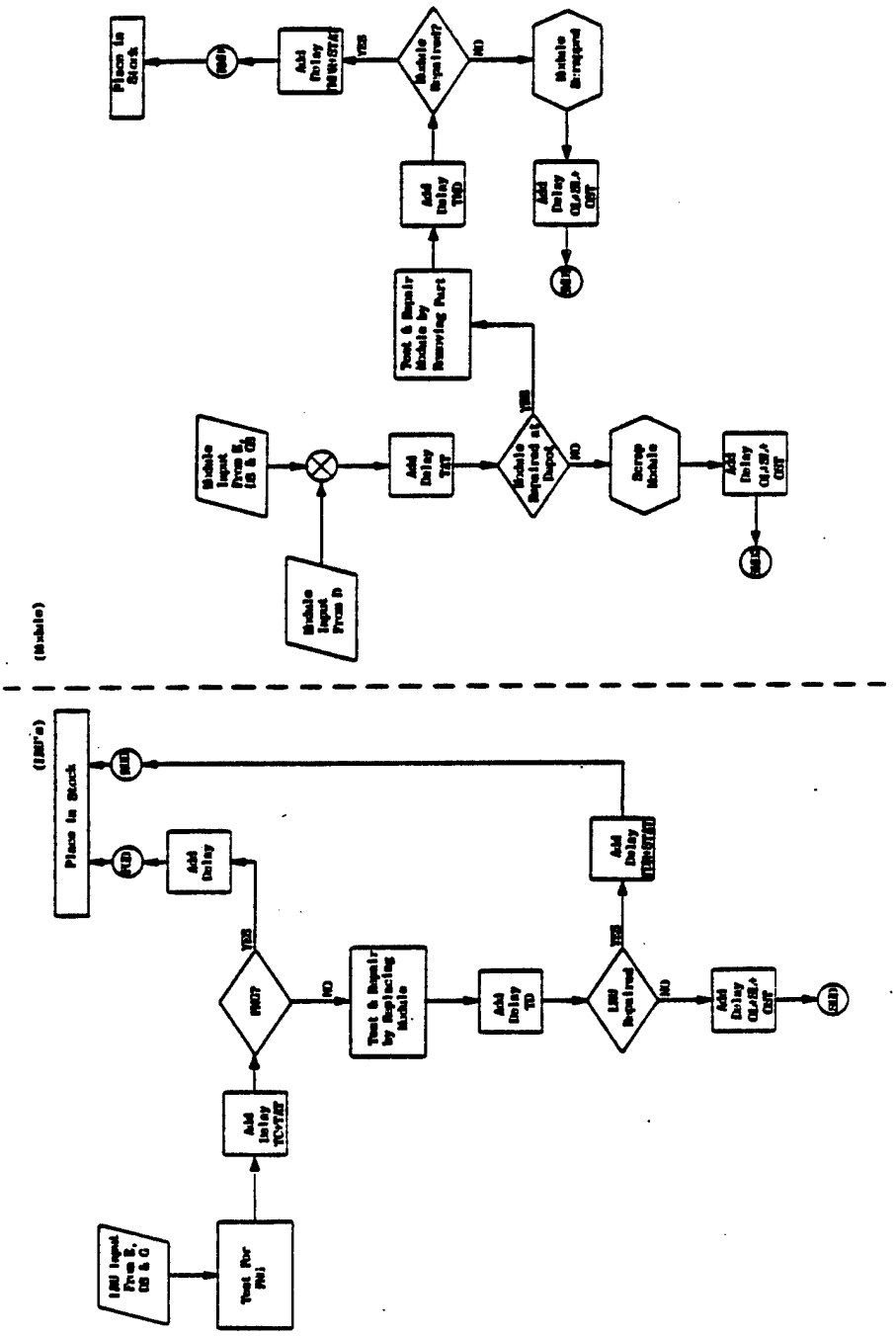


FIGURE 5
DEPOT LEVEL (D) - MAINTENANCE FLOW



SECTION 3 - LOGAM FEATURES (PROVISIONING RULES)

3.2.3.1 Computation of LRUs, Modules, and Parts - In order to fully understand the LOGAM maintenance rule, it is desirable to discuss the LRU flows through the various maintenance levels.

At the equipment level (ED) provisioning is a function of the failure rate (E) and:

1) Mean time to repair (TRC). It is the total corrective maintenance down time in clock hours during the stated period to return the materiel system to an operational state. TRC must be transposed into days for computation of stock. TRC ~~will always have a number~~.

2) Mean time in days spent awaiting a replacement LRU or module (TATE). It is the average length of time in days required to obtain an LRU or module (based upon the maintenance policy) from a support activity. TATE must be calculated based upon the authorized location of replacement LRUs and modules as expressed in the H(I) matrix.

3) Mean downtime in days awaiting maintenance personnel, shop space or transportation at ED (TAT(1)).

4) Mean time in days an LRU or module awaits evacuation to the next maintenance echelon (DTE). This is to account for waiting time to package and ship an item. If a maintenance policy calls for all LRUs to be evacuated, DTE is charged to all LRUs. If a policy calls for a percent of LRUs to be repaired at ED and a percent to be evacuated, DTE is charged to the percent evacuated.

5) Average time in hours to check for a false "no go" (TC). It must be transposed to days for computation of stock. Only LRUs are checked for false "no go's" and only in maintenance concepts that provide for LRU repair/checkout in the Maintenance Policy Matrix. Only policies GC, GI, GJ, and GK have LRU screening capability at ED Level.

6) Mean time in hours to test a LRU at ED level (TE). It must be transposed to days for computation of stock.

7) Mean time in hours to repair an LRU at ED level (TER). It must be transposed to days for computation of stocks.

8) Operating Level, Safety Level and Order and Ship Time (OL+SL+OST) in days. This delay is used to account for stocks to replace those that are scrapped.

Provisioning quantities required to support the maintenance policy are computed considering the flow of LRUs and modules through the selected maintenance concept. As the LRU or module flows through the system provisioning days are computed and accumulated based upon the delays encountered. For example an LRU removed at ED and evacuated to DD would accumulate delays TRC, TAT(1), TATE & DTE while an LRU which was removed, repaired, and returned to stock at ED would accumulate delays TRC, TAT(1), TATE, TC, TE and TER. Thus as the LRUs and modules flow through the evacuate, screen for false "no go", repair, and scrap gates the delays associated with these actions are accumulated and initial provisions of LRUs and modules are computed to account for the time required to process (repair, evacuate, etc.) the unserviceable LRU or module. The following calculations are made to compute LRU provisioning quantities at ED level. (See Figure 2).

Since all LRU removals are charged TRC, TAT(1) and TATE delays, the delays are computed at time of removal rather that at time of disposition (e.g. evacuate, repair, scrap & screen). PDRE is the provisioning days required to account for removal of LRUs at Equipment level.

$$PDRE = TU*(TRC/(24*WER/168)+TAT(1)+TATE)$$

where: TU=(F+FNG+A) is the total number of LRUs removed from the material system.

TRC is divided by 24*WER/168 to translate TRC to days and to allow the work week in hours (WER) of ED level maintenance personnel to be considered.

****

Evacuation of LRUs to higher support levels causes delays depending on how they are evacuated. They can be evacuated immediately because ED level has no LRU repair capability or they can be evacuated after repair was attempted but did not succeed and the LRU was placed in the "in limbo" status. PDEUE1 is the provisioning days for those LRUs for which the G policy allows no repair at ED level. The evacuated flow to OD, DI, and DD are identified as DSUE1, GSUE1, and DUE1.

$$DSUE1=(F+FNG)*(GB+GD+GF+GL+GM+GN+GO+GP)+(F*DPT+FNG)*(GH+GQ)$$
$$GSUE1=(F+FNG)*(GE+GR+GS)$$
$$DUE1=(F*DPT+FNG)*(GG+GT)$$

where: DPT=2.-DAOQL is used to compute the Depot work/load recycled.

Since all policies except GA, GC, GI, GJ, and GK have false "no go" screening capability all the true failures plus false "no go's" must be evacuated to the next level. The term DPT accounts for LRUs that were sent from ED to DD level (policies GG, GH, GQ and GT) that were supposedly fixed but when returned to the organization were found faulty.

The delay is calculated by multiplying the total flow by the handling delay at equipment level (DTE).

PDEUE1=(DSUE1+GSUE1+DUE1)*(DTE)

PDEUE2 is the provisioning days for those LRUs for which the G policy allows repair at ED level. Since not all LRUs are repaired due to the FUE fraction some end up "in limbo" and are evacuated. This percentage is identified as ULE.

DSUE2=F*ULE*(GI)
GSUE2=F*ULE*(GJ)
DUE2=F*ULE*(GK)

The total provisioning days are then calculated by multipling the total flow by the handling delay (DTE), the time to screen the LRU for a false "no go" (TC) and the time to test the LRU (TE).

PDEUE2=(DSUE2+GSUE2+DUE2)*(DTE+(TE+TC)/(24*WER/168))

The PDEUE is the total number of provisioning days at ED level due to evacuated LRUs and is computed by summing the two delays.

PDEUE=PDUE1+PDUE2
PDEUE1 + PDEUE2
****

The scrapping of LRUs generates provisioning days depending on how they were scrapped. They can be scrapped because for whatever reason, the LRU could not be repaired at ED level and the G policy had no higher level authorized (OD, DI, or DD) or because of items that were declared not economically repairable and are then discarded at ED level. Those units which are scrapped due to a G policy with no higher maintenance level are charged only Operating Level, Safety Level, and Order and Ship Time (OL+SL+OST) delays, since these items were not screened and tested before they were scrapped. All other scrap will have the additional charge for FNG screening (TC) and test time (TE).

The various scrap flows are identified as SUE1, SUE2, and SUE3.

SECTION 3 - LOGAM FEATURES (PROVISIONING RULES)

```
SUE1=(F+FNG)*GA
SUE2=F*ULE*GC
SUE3=F*SUE*(GC+GI+GJ+GK)

PDSUE=(SUE1+SUE2+SUE3)*(OL(1)+SL(1)+OST(1))+
      (SUE2+SUE3)*(TC+TE)/(24*WER/168)
```

****

Attritions add additional provisioning days, in that stock must be provided to account for the loss of the LRU. Attritions are those LRUS which are removed from the materiel system at ED level but for whatever reason (lost, stolen, etc.) are lost to the supply system. The provisioning delay due to attritions is similar to scrap

```
PDAU=A*(OL(1)+SL(1)+OST(1))
```

****

The time to test and identify false "no go" LRUs also contributes to provisioning days and is charged a time for testing (TC). At ED level, false "no go" screening capability is available only for concepts GC, GI, GJ, and GK. The flow of false "no go's" which are screened at ED level and found serviceable and returned to stock is identified as FGE.

```
FGE=FNG*(GC+GI+GJ+GK)
```

The number of provisioning days is computed by multiplying the flow and the FNG screening (TC) time.

```
PDSKUE=FGE*TC/(24*WER/168)
```

****

The LRU repair flow at ED level is expressed as TUFTE

```
TUFTE=URE*FUTE
Where; FUTE=F*(GC+GI+GJ+GK)
```

The provisioning days due to repair (PDRUE) are computed by multiplying the flow by delay times. LRUs which have been repaired have gone through false "no go" screening and are charged TC. They also have undergone diagnostic testing and are charged TE, and finally they have been repaired and are charged repair time TER.

```
PDRUE=TUFTE*(TC+TE+TER)/(24*WER/168)
```

****

Since provisioning is based upon the system operating hours, operational availability and the total number of systems deplcyed it is necessary to multiply PDUE by these factors and convert from hours to days. Thus the term OFACT is computed by multiplying the operating time factor (OTF) by 8766 hours per year by the number of installation (ED) and by the inherent availability (AYZIS) and then dividing by 365 days per year.

OFACT=SAVE*8766./365.
Where; SAVE=ED*OTF*AYZIS

Thus the total provisioning quantity for LRU's at ED level is:

PDUE=(PDRE+PDEUE+PDSUE+PDAU+PDSKUE+PDRUE)*OFACT

The remainder of the calculations for LRUs, modules, and parts follow the same logic as shown above and will not be duplicated.

****

3.2.3.3 Distribution of LRUs, Modules, and Parts - The equations used to distribute the required LRUs based upon the stockage matrix (H(I)) are designed to distribute spares to the lowest support level (ED) based upon the H array. For example, assume the H array is H=1,1,1,1, which places stock at each echelon and EDS=6, ODS=2, DIS=1 and DDS=1. Also assume that the required provisioning quantities computed as explained above are as follows:

MAINT. LEVEL	LRU QTY
ED	12.72
OD	7.95
DI	4.77
DD	27.56

In allocating LRUs, a round-up of .5 is assumed. Since the ED LRU requirement is 12.72 and EDS=6 we can have 2 LRUs at each EDS location. The OD LRU requirement is 7.95 and .72 LRUs were left over at EDS due to round off. This added to 7.95 for ODS equals 8.67. The round off is to 8; therefore, each CDS location will have 4 LRUs. The DI LRU distribution is 4.77 but .67 LRUs were left from ED and OD; therefore, the single DIS location will require 5.44 but the round off is to 5. The DDS location will have the remaining 27. The distribution of provisioning is as much as possible, determined by the requirement at each echelon.

In all cases the H array controls stock location. If stock is

not located at an echelon, but work is performed there, the amount of stock that would be computed for that location is added to the next more forward echelon i.e., if H=1,0,1,1, and LRU stock is computed as shown for the sample case. ED=12.72, OD=7.95, DI=4.77, and DD=27.56, stock is located as follows:

ED=20.67, OD=0, DI=4.77, DD=27.56
Rounded to ED=18, OD=0, DI=7, DD=28

The round-off factor that is used has a significant impact on the level of stock distributed. For instance in the above example if 0.999 was used instead of 0.5, the stock would be rounded to ED=24, OD=0, DI=2 and DD=27.

If H=0,0,1,1, stock would be located at the forward most echelon authorized stock. Stock is located as follows:

ED=0, OD=0, DI=25.44, DD=27.56
Rounded to ED=0, OD=0, DI=25, DD=28

In cases where stock is located at a higher echelon the analyst should assure that the mnemonic TATE expresses a reasonable time to obtain the needed LRU or module.

The logic for module and part distribution is also tied to the H array except that parts are not stocked at ED.

3.3 Sensitivity Testing

After a set of individual LRU cases have been run as a baseline case, it is often desirable to be able to rerun the entire data set with selected changes in certain of the input variables. To facilitate this, the program writes a copy of the input data to a memory device during the baseline run. Subsequently, these data may be retrieved, edited, and rerun. These reruns of the input based on selected editing are referred to as sensitivity runs.

3.3.1 Sensitivity Input Array - One of the elements of the input NAMELIST/L/ is an array named SENSY. Values input to this array are used to direct the conduct of sensitivity runs. The array SENSY, stored in common block SENS, has Dimension 266. Entries into these 266 storage locations perform the following functions:

a) Specify the number of input variables whose values are to be edited during the sensitivity runs.

b) Specify the number of times the inputs are to have their values edited. (This specifies the number of sensitivity runs).

c) Specify the rules to be used for the editing of each designated input.

d) Designate the inputs to be altered.

e) Furnish the numeric values to be used by the specified rules in the edition of the designated inputs.

3.3.1.1 First Element of the SENSY Array - In more detail, the first element of SENSY, i.e., SENSY(1), is used to accomplish Function (a) in Section 3.3.1. A positive, real, whole number is entered to state the number of inputs being tested. Within the program, this is called MODE. This program is currently written so that MODE may range from one to twelve inputs. More than twelve inputs results in an error message:

BAD SENSY

followed by a printout of the contents of array SENSY, the sensitivity test is abandoned, and the program resumes as though it were a new start after completing sensitivity testing.

The exact value 0 is used to denote that sensitivity testing is off and the program is running baseline cases. This value exists at program start by initialization in a BLOCK DATA subprogram. Thus, SENSY need not be input to run the baseline case. Similarly, after the completion of all the work of a sensitivity

run, SENSY(1) is reset to zero and no input is needed to resume analysis of baseline cases. In fact, all elements of SENSY are reset to zero. (Input of negative values in SENSY(1) are not detected by the program. The program will run SENSY with unpredictable results. Negative values should not be entered for SENSY(1).

3.3.1.2 Second Element of the SENSY Array - The second element of SENSY, i.e., SENSY(2), is used to carry out Function (b) given in Section 3.3.1. A positive, real, whole number is entered to stipulate the number of sensitivity runs. This is known as NPASS within the program. Due to the limitations of the dimensionality of SENSY, there is a limit to the number of passes that can be made by one loading of SENSY. The number depends on MODE. The limits on NPASS for the twelve possible values of MODE are listed below.

MODE	NPASS LIMIT
1	262
2	130
3	86
4	64
5	50
6	42
7	35
8	31
9	27
10	24
11	22
12	20

The remaining elements of SENSY are furnished as ordered sets of size MODE. Thus, if only one input is being tested, the set size is one; if two, the size is two, etc. up to the limit of twelve per set when MODE is 12.

3.3.1.3 Third Element of the SENSY Array - Function (c) in Section 3.3.1 is the specifications of the edited rules. This is accomplished by furnishing a set of positive, real, whole numbers. There is one rule number in the set for each of the MODE variables to be varied. The permissible rule numbers are as follows:

Rule Number	Effect
1.	Assign
2.	Add
3.	Subtract
4.	Multiply
5.	Divide

If any other value is used, an error message will be written as follows:

ILLEGAL RULE KRULE = X

giving the sequence of the rule. That input will not be altered and the program will continue. Later sets of entries in SENSY contain values to be used with these rules. Thus, for Rule 1, the value furnished is used instead of the value in the baseline data. Rules 2, 3, 4, and 5 take the value using addition, subtraction, multiplication, or division as specified.

Within the program, the set of rules is stored in array NRULE, of Dimension 12. Should Rule 5 ever encounter the value zero in SENSY, the error message

ATTEMPTED DIVIDE ERROR INDEX = X

will be written where X will be the sequence number in the SENSY array. The program will continue using the baseline value for that variable.

Thus, with MODE in SENSY(1), NPASS in SENSY(2), the set of MODE rules are entered in SENSY(3) to SENSY (MODE + 2).

3.3.1.4 Designation of the Variables to be Tested - In the designation of the variables for sensitivity testing, the program is structured to reference them by their numbered positional location in common block INPUT rather than by name. The numbered sequence for addressing LOGAM inputs to be sensitivity tested is given in Table 2. The sequence numbers are also included with the input definitions in section 5.2. (INPUT DEFINITIONS). Thus, to refer to input E, the LRU failure rate, the number to be entered in SENSY is 91. The reference numbers are to be entered as positive real whole numbers. Should a value other than those in the table be entered, an error message will be entered as follows:

ILLEGAL VARIABLE ADDRESSED = M

where M is the illegal number. The program will continue and no variable will be altered for that bad value.

Thus, to carry out Function (d) in Section 3.3.1, an altered set of MODE variable numbers is entered into SENSY (2 + MODE + 1) through SENSY (2 + MODE + MODE). These are stored in the program array NVAR, of Dimension 12.

TABLE 2
SENSITIVITY INPUT LISTING

1	ARA	40	CKPD	79	CUBEU	118	FLM
2	ARAD	41	CKPI	80	CUCE	119	FMD
3	AYZP	42	CKPO	81	CUP	120	FMI
4	CAD	43	CKUD	82	CV	121	FMO
5	CALMAN	44	CKUE	83	DAOQL	122	FN
6	CALPUB	45	CKUI	84	DD	123	FNGF
7	CALSET	46	CKUO	85	DDS	124	FNSP
8	CCAL	47	CLRUPG	86	DI	125	FSA
9	CCALP	48	CMODPG	87	DIS	126	FTI
10	CCALR	49	CMP	88	DTE	127	FTII
11	CCSP	50	CONMAN	89	DTI	128	FTM
12	CCSPP	51	CONTCT	90	DTO	129	FTP
13	CCSPR	52	CPE	91	E	130	FTU
14	CDDI	53	CPI	92	EACAL	131	FUD
15	CDEO	54	CPII	93	EACSP	132	FUE
16	CDFD	55	CPP	94	ED	133	FUI
17	CDID	56	CPUBII	95	EDS	134	FUO
18	CDIO	57	CPUBV	96	EE	135	G(1)-GA
19	CDIST	58	CPV	97	EREI	136	G(2)-GB
20	CDMAN	59	CRI	98	ETE	137	G(3)-GC
21	CDOE	60	CRII	99	ETEI	138	G(4)-GD
22	CDOI	61	CRM	100	ETI	139	G(5)-GE
23	CDPMAN	62	CRP	101	ETII	140	G(6)-GF
24	CDPRMN	63	CRU	102	EVDM	141	G(7)-GG
25	CDRMAN	64	CRV	103	EVDR	142	G(8)-GH
26	CEMAN	65	CSDEP	104	EVDT	143	G(9)-GI
27	CEN	66	CSDSU	105	EVEM	144	G(10)-GJ
28	CEND	67	CSESU	106	EVER	145	G(11)-GK
29	CERMAN	68	CSGSU	107	EVET	146	G(12)-GL
30	CFTD	69	CTCPUB	108	EVIM	147	G(13)-GM
31	CGMAN	70	CTRA	109	EVIR	148	G(14)-GN
32	CGRMAN	71	CTRAD	110	EVIT	149	G(15)-GO
33	CI	72	CTRCAL	111	EVOM	150	G(16)-GP
34	CII	73	CTRI	112	EVOR	151	G(17)-GQ
35	CKIT	74	CTRII	113	EVOT	152	G(18)-GR
36	CKMD	75	CTRSPT	114	FE	153	G(19)-GS
37	CKME	76	CTRV	115	FI	154	G(20)-GT
38	CKMI	77	CUBEM	116	FII	155	H(1)
39	CKMO	78	CUBEP	117	FINT	156	H(2)

TABLE 2 (CONTINUED)
SENSITIVITY INPUT LISTING

157 H(3)	200 QTPD	243 TAYZ(8)	286 TUMD
158 H(4)	201 QTPI	244 TAYZ(9)	287 TUMI
159 HPM	202 QTPO	245 TAYZ(10)	288 TUMO
160 HPP	203 RDD	246 TC	289 WD
161 HPU	204 REO	247 TD	290 WDM
162 IBUG	205 REPEAT	248 TDI	291 WDR
163 IFLAG	206 RF	249 TDMAN	292 WE
164 IMF	207 RID	250 TDMW	293 WEM
165 INHIB	208 ROI	251 TDPMI	294 WER
166 IO	209 SL(1)	252 TDPMII	295 WI
167 IOPER	210 SL(2)	253 TDPRI	296 WIM
168 IS	211 SL(3)	254 TDPRII	297 WIR
169 JTED	212 SL(4)	255 TDR	298 WM
170 NA	213 SMD	256 TDRMAN	299 WMR
171 NB	214 SME	257 TE	300 WMT
172 NU	215 SMF	258 TEMAN	301 WO
173 OD	216 SMI	259 TENMAN	302 WOM
174 ODS	217 SMO	260 TEO	303 WOR
175 OL(1)	218 SPE	261 TER	304 WP
176 OL(2)	219 SPEV	262 TERMAN	305 WTKIT
177 OL(3)	220 SPEVR	263 TF	306 WU
178 OL(4)	221 STAT	264 TFR	307 YAT
179 OST(1)	222 SUD	265 TGMAN	308 YD
180 OST(2)	223 SUE	266 TGRMAN	309 YMWO
181 OST(3)	224 SUI	267 TI	310 YP
182 OST(4)	225 SUO	268 TID	311 YR
183 OTF	226 SVE	269 TIMW	312 YZ
184 P	227 SVR	270 TIO	313 ZFL
185 PMR	228 SVT	271 TIR	314 ZI
186 PP	229 SVV	272 TMD	315 ZM(1)
187 PPR	230 TALMAN	273 TMDD	316 ZM(2)
188 PUR	231 TAT(1)	274 TMDR	317 ZM(3)
189 QMM	232 TAT(2)	275 TMI	318 ZM(4)
190 QMP	233 TAT(3)	276 TMID	319 ZO
191 QMU	234 TAT(4)	277 TMIR	320 ZP(1)
192 QTD	235 TATE	278 TMO	321 ZP(2)
193 QTE	236 TAYZ(1)	279 TMOD	322 ZP(3)
194 QTI	237 TAYZ(2)	280 TMOR	323 ZU(1)
195 QTMD	238 TAYZ(3)	281 TOE	324 ZU(2)
196 QTME	239 TAYZ(4)	282 TOI	325 ZU(3)
197 QTMI	240 TAYZ(5)	283 TOMW	326 ZU(4)
198 QTMO	241 TAYZ(6)	284 TONMAN	
199 QTO	242 TAYZ(7)	285 TRC	

3.3.1.5 Designation of the New Values for the Inputs - The remaining portion of SENSY is used to enter NPASS sets of MODE elements to carry out Function (e) in 3.3.1, i.e., supply the values to be used to alter each variable designated, according to the rule, for each pass. Thus, to recapitulate:

SENSY(1)	MODE	Number of inputs to be tested.
SENSY(2)	NPASS	Number of Runs or Passes.
SENSY(3) to MODE + 2	NRULE	Set of Rules for Editing.
SENSY [(MODE + 3) to (MODE + MODE + 2)]	NVAR	Designation of Input Variables.
SENSY [(MODE + MODE + 3) to (MODE + MODE + MODE + 2)]		First Set of Variables and so forth.

3.3.2 Example of NAMELIST Inputs for Typical Sensitivity Run

If it is desired to investigate the simultaneous variation of failure rate and false "no go" fraction, a typical set of values would be as follows:

MODE = 2.

To run three sets of data:

NPASS = 3.

The input designators, from Table 2

E 91
FNGF 123

In the baseline run, if FNGF was 0.2 for all LRUs and it is desired to run 0., 0.4, and 0.6, for all LRUs, then Rule 1 is used and 0., 0.4, and 0.6 are assigned at the first, second, and third pass, respectively.

For the failure rate E, all LRUs have different values. Rule 1 is not useful as there is no desire to assign the same failure rate to each LRU. More commonly, it is desirable to run multiples of the baseline. Thus, Rule 4 is useful. If no change is desired for the first pass, then the value 1 is used. (This will show the effect of FNGF = 0, without changing E). If for

second and third passes, simultaneously with the doubling and tripling of the false "no go" fraction, it is desired to increase the failure rate by factors of 5 and 10, then values 5 and 10 are used. The data input via NAMELIST/L/ for this sensitivity case will have the appearance as shown in Figure 6.

The program will run each pass specified by SENSY. The baseline data store is "rewound" after each pass. The stored baseline data are not ALTERED. After the last pass, control returns tc the program and execution continues looking for new input data. At this point, another SENSY may be entered and the same saved baseline data will be further sensitivity tested.

If another SENSY is not entered, the new baseline cases may be entered. In such a case, the old saved baseline data are destroyed and the new set is saved. At this point, a control (NU=-4) may be entered to stop the program.

3.3.3 Sensitivity Testing Specification - Included at the end of Section 7.1 is a sensitivity NAMELIST input data set that was run with the baseline USAREUR and CONUS data set. As shown, the input cards for sensitivity testing are placed after the final LRU data set. To use the sensitivity test feature of LOGAM, at least three cards must be punched. The first two are generally used as header cards to identify certain factors pertaining to the particular sensitivity run set. The third (and subsequent cards if required, the exact number depending on the number of variables and passes to run) is the input array in the NAMELIST format as discussed in Section 5.2.

3.3.4 Sensitivity Output - Along with the SENSY array, the control INHIB may be used to suppress the individual LRU printout. If INHIB = 1 is used, only the totals page will be printed. If the control INHIB is not activated, the output printout contains the same number of pages as the baseline results. These printouts, however, will show the results for the new values of the inputs as controlled by the rules contained in the SENSY NAMELIST array.

The printout of a totals page will always be preceded by a printout of the new values of the inputs identified by the designation number given in Table 2. Thus, the new value of the input assigned by activating the sensitivity test feature of LOGAM is always documented. Sample output using SENSY data is discussed in Section 6.7.

FIGURE 6
SAMPLE SENSITIVITY INPUT

```
                RULE     NVAR      PASS1      PASS2    PASS3      UNUSED
SENSY = 2,3,    4,1,     91,123,   1.,0.,     5.,.04,  10.,.06,    254x0
        . .     . .      .    .    .  .       .    .   .     .
        . .     . .      .    .    .  .       .    .   .     .
        . .     . .      .    .    .  .       .    .   .     ...FNGF=.6
        . .     . .      .    .    .  .       .    .   .
        . .     . .      .    .    .  .       .    .   ....E=10xBaseline Value
        . .     . .      .    .    .  .       .    .
        . .     . .      .    .    .  .       .    . ..FNGF=.4
        . .     . .      .    .    .  .       .
        . .     . .      .    .    .  .       ...E=5xBaseline Value
        . .     . .      .    .    .  .
        . .     . .      .    .    .  ....FNGF=0
        . .     . .      .    .    .
        . .     . .      .    .    .....E Unchanged
        . .     . .      .    .
        . .     . .      .    ....FNGF
        . .     . .      .
        . .     . .      ....E
        . .     . .
        . .     . ....Replace FNGF with new Value
        . .     .
        . .     ...Multiply E (Baseline Value) by Input Factor
        . .
        . ....NPASS, 3 Sets of Data
        :
        ...MODE, 2 INPUT Variables per Pass (E and FNGF)
```

3.4 Availability Accumulator

Inherent and operational availabilities can be accumulated at the system and subsystem levels. System availability is the accumulation for all LRUs in a case total and subsystem availability would be for groups of LRUs that make up a case total. The groupings of LRUs are determined by the inputs for TAYZ. When the inputs for TAYZ are such that accumulations are made, the accumulated values are stored into arrays CAYZI and CAYZ. CAYZI and CAYZ are the inherent and operational availability arrays, respectively. The first location of each array will contain the system availability and each subsequent NA locations contain subsystem availabilities. NA is input as the number of accumulations to be made; i.e., the system plus the number of subsystems. The following discussion includes an example for availability accumulations.

TAYZ is an array of dimension ten to provide the capacity for ten availability accumulators (definition for NA specifies how many of the ten accumulators are active). A value must be entered for each of the ten availability accumulators; however, only the first NA of the ten are actually used. For example, if a system consists of eleven LRUs and if that system logically subdivides into functional subsystems, the arrangement of the LRUs in the input data deck should be the first four LRUs that constitute the first subsystem, the next five constitute the second subsystem, and the last two constitute the third subsystem. Then if the user wanted to keep the availability tally for the total system and also for each subsystem, four tallies are required. He would input NA = 4. For TAYZ, he would input the following:

LRU No.		LRU No.	
1	TAYZ = 1., 1., 8*0.,	5	TAYZ = 1., 0., 1., 7*0.,
2		6	
3		7	
4		8	
		9	
		10	TAYZ = 1., 2*0., 7*1.,
		11	

All LRUs would be tallied into the first accumulator, i.e., the first element of the TAYZ array is unity for every LRU. The first four LRUs would be tallied into the second accumulator, i.e., the second element of TAYZ is unity for the first four LRUs and zero for all others. LRUs five through nine would be tallied into the third accumulator, i.e., the third element of TAYZ is unity for these LRUs and not for any others. The last two LRUs will be tallied into the fourth accumulator, i.e., the fourth element of TAYZ is unity for these two and zero for all others.

SECTION 3 - LOGAM FEATURES (Avaiability Accumulator)

Values of TAYZ beyond the fourth element are immaterial because NA = 4. On the case total page, four availabilities will print across the page. The first will be the system availability. The second will be the availability of the first subsystem. The third will be for the second subsystem. The fourth, and last, will be for the third subsystem. Availability outputs are discussed in section 6.5, Maintenance Outputs.

3.5 LOGAM/AR570-2 Maintenance Data Interface

Annual maintenance positions from the AR570-2 MACRIT data base can be interfaced into the LOGAM TOE namelist inputs. The maintenance positions input to LOGAM are sorted by MOS from the MACRIT data base. To obtain this sorted data base one must execute a sort program written by the System Planning Corporation (SPC). The sort program inputs equipment line items. All MOSs associated with the line items are sorted alphanumerically and written to an output tape (8). TAPE8 then becomes the input for LOGAM.

Each record of data on TAPE8 includes the MOS, line item, nomenclature, productivity category, and annual maintenance manhours and positions for three maintenance echelons. The only inputs from the file that will be used in LOGAM are the MOS and the maintenance positions.

The maintenance positions output from the SPC program include the effect of productivity factors on manpower requirements. The SPC program has productivity factors built in which the user selects with a category flag (ICAT). For a detailed description of the SPC program refer to the Support for LOGAM User's manual, dated April 1981. This manual was prepared for the Systems Analysis Division of the U.S. Army Missile Command, Redstone Arsenal.

LOGAM reads the data file sorted by MOS and accumulates the maintenance positions for identical MOS entries. Positions are available for one maintenance facility at each of three echelons; Equipment, Direct Support, and General Support. The accumulations for positions at these three levels are made in Subroutine SUPTMH. For each MOS summary, Subroutine SGAPER is called to equate MACRIT maintenance positions to personnel grades. Comparison of positions with a standard of Grades Authorization (SGA) table is used in determining the quantity of personnel by grade.

One SGA table was selected as the "typical" grade structure for all MOSs. Ordinarily each MOS has it's own SGA table, but to build tables for all possible MOSs would create a data base too massive for LOGAM use. Therefore the SGA table for MOS 35B, an Electrical Instrument Repairer, was selected as "typical". The number of positions per grade for MOS 35B is shown in Table 3.

The MOS 35B table is capable of storing data for thirteen grade levels at ten positions. Presently, there is data stored for only the first five grades, E3 through E7. The first four grades (E3-E6) are considered maintenance personnel and the fifth (E7) is for supervisory personnel. The remaining grades, if input, would be considered supervisory positions. Table 3 has the MOS 35B data for thirteen grades used in the TOE. A blank space in

SECTION 3 - LOGAM FEATURES (AR570-2 MACRIT Data)

TABLE 3
Standard of Grade Authorization for MOS 35B

	GRADE	1	2	3	4	5	6	7	8	9	10
	E3		1	1	1	2	2	2	3	4	4
Maintenance	E4	1	1	2	2	2	2	2	2	2	3
Personnel	E5				1	1	1	2	2	2	2
	E6						1	1	1	1	1
	E7					1	1	1	1	1	2
	E8										
Supervisory	E9										
Personnel	W0										
	02										
	03										
	04										
	05										
	06										

the table indicates the absence of a position.

The number of personnel per grade is found by searching the column coinciding with the positions accumulated for an MOS. The number of personnel in each grade should sum to the column number. There is one exception, since E7 is a supervisory position it is not included in the maintenance positions. For an example, the number of positions from the MACRIT data is 7.1 for an MOS accumulation. The personnel by grade will then be taken from column 8 since positions are rounded to the next higher integer. The number of maintenance personnel by grade is three E3s, two E4s, two E5s, and one E6 which totals to the eight positions. There is also one E7 in a supervisory position for the eight maintenance positions.

The logic for determining the quantity of personnel by grade when the number of positions exceeds ten is included in subroutine SGAPER. The following equation illustrates the procedure for computing quantities by grade for positions greater than 10.

$$NG_i = \frac{NP-1}{10} * SGA(10,i) + SGA(NP - \frac{NP-1}{10} * 10 ,i)$$

Where,

NG_i = the number of positions for grade level i.

NP = The number of maintenance positions.

SGA = Table of maintenance positions by grade.

i = The grade for which positions are to be computed.

Essentially the algorithm multiplies the contents of the tenth position of SGA by the total position's (NP) integer multiple of 10 and adds the contents of the position respresented by the difference of the total position and the integer multiple of 10.

After all MOSs on the data file have been processed, the personnel quantities are interfaced with the TOE namelist input data. This interface is performed in subroutine TOEPER. Before the maintenance support personnel can be integrated into the TOE inputs the user must first set up the personnel fields in the namelist inputs. He does this by specifying a 1 in the first column of a ten column field. Each TOE line item has ten input fields associated with it. To differentiate between the line items for storing maintenance support personnel and any other personnel related line item, the second field of the line item is set to zero. The second field is the number of personnel for the line item where the line item represents a grade level.

Subroutine TOEPER will store the personnel quantity for the grade represented by this line item into the second input field. The grade level for a personnel line item is stored in the fourth field where the third field denotes that the grade is either for enlisted or officer personnel. With the exception of the annual cost for a person in grade, these are the only inputs required. The annual cost is entered in field position ten. The quantity of personnel and type of charge flags (maintenance, overhead) are set by the program. For an explanation of the TOE namelist inputs refer to section 5.2 of this manual.

The maintenance positions accumulated from the sorted MACRIT data file are for one facility at each of three maintenance echelons. Before storing the positions into the TOE inputs (T(2,J)) they are multiplied by the number of facilities at a maintenance level. The three levels for which positions are stored are Equipment, Direct Support, and General Support. The positions at the three levels are multiplied by ED, OD, and DI, respectively. These positions are then summed and stored into the TOE data array for the grade represented by line item T(2,J), where J is the line item number.

Once the maintenance support positions have been incorporated into the TOE data, the original logic of evaluating O&S costs is followed. The maintenance support personnel has simply become a part of the TOE data base and by inputting IOPER=1 in NAMELIST/L/ the TOE data is executed.

3.6 TOE Operation and Support

Operation and support (O&S) costs based on a TOE structure are evaluated in Subroutine OPER and OPER's supporting routines. TOE costs are added to the LOGAM maintenance costs and output in a format consistent with Department of Army Pamphlet No 11-4. TOE costs can be evaluated in the LOGAM program for any number of theaters. The user inputs the TOE data (T array) with any of the individual LRU cases of a theater and sets the option flag IOPER=1. The cost totals (case total) for a theater are output with the control flag NU < -1. TOE costs will only be evaluated and printed when a case total is requested and IOPER=1.

3.6.1 TOE Operation and Support Inputs

TOE data is input with the "T" array of NAMELIST/L/. T is an array of 2000 words in length which accomodates input values for 200 TOE line items. There is a maximum of 10 values that can be input for each line item. The first value of each line item must be a number in the range of 1 to 8. This value instructs OPER as to which of 7 subroutines is to be used in evaluating the cost for the line item. A value of 8 will terminate TOE processing and return to LOGAM to print a System Operation and Support case total.

In some cases it is important that the numerical value of the first line item entry be input in ascending order. When computing personnel related costs in Subroutine TWO, the personnel quantities must have been previously accumulated in Subroutine ONE. Another instance of this anomaly is in Subroutine SIX where the accumulation of instrumentation costs must have been made previously in Subroutine FIVE. Even though the evaluation of all line items are not dependent on computations of previous line items, it is good practice to always input the line item types in ascending order.

The TOE data are normally input with the "T" appearing only once followed by 10 values per card such as:

```
T=1.,0.,5.,1.,0.,0.,0.,0.,0.,60000.,
```

Each card could be input using subscripts, i.e., "T(1)=","T(11)=", etc. By using subscripts, data for additional theaters could be input by changing only the line item affected. For example if the only change from the first theater is in line item 11, then "T(11)=" with the 10 associated values are the only necessary inputs. When new line items are to be inserted in the list, the inserted item could be input with the appropriate subscript and the remaining line items from the previous theater repeated.

SECTION 3 - LOGAM FEATURES (TOE O&S)

There is one instance where the values of the TOE inputs are modified within the LOGAM program. This occurs while interfacing AR570-2 MACRIT maintenance data with the TOE data. The contents of the personnel line items (T(1,J)=1 where the 1 denotes personnel) are modified. T(2,J)=0 and IMT>0 would have to be input before "T" would be modified. Refer to section 3.5, "LOGAM/AR570-2 Maintenance Data Interface", for a description of "T" elements that are modified. If the AR570-2 maintenance data for the second theater is the same as the first then it's not necessary to process the MACRIT data file again. Set IMT=0 and the personnel data of the "T" array will remain the same as in the previous theater.

The "T" array is double dimensioned in the TOE processor subroutines but single dimensioned when being input. Subroutine FINLE that reads the NAMELIST inputs has the capability of reading only single subscripts hence a value for "T(4,2)=3" would be input as "T(42)=3" in NAMELIST/L/.

There are 7 subroutines in addition to the driver routine (OPER) used to process TOE data. Each subroutine name (ONE, TWO, etc.) corresponds to the first legitimate value (1 thru 8) of a 10 element field. The second value of the field (where applicable) determines which area of the subroutine to use. The control values and their corresponding subroutine names are shown in Table 4.

Table 5 includes a list of the applicable input fields and the mnemonics for the 10 element fields. The mnemonics are for reference only, they are not input names within NAMELIST/L/. An explanation for each of the applicable fields can be found in Table 6. The absence of a mnemonic in one of the 10 element fields indicates that this field is not used in the TOE processor routines. The asterisks (*) in Table 5 for input type 6 indicate that there are instrumentation costs input here but there are no nmemonics assigned. There can be up to a maximum of 9 instrumentation costs input. Subroutine FIVE simply sums these costs.

TABLE 4

DESCRIPTION OF POSSIBLE LINE ITEMS

FIELD VALUES		DATA EVALUATED
First	Second	
1	N/A	Personnel data from the TOE
2	1	Personnel cost multipliers, list 1
2	2	Personnel cost multipliers, list 2
2	3	Personnel cost multipliers, list 3
3	N/A	Equipment/Fuel usage data
4	N/A	Ammunition usage data
5	N/A	Instrumentation Missiles
6	1	Arty/Ord inputs
6	2	Follow on training inputs
7	N/A	Signals the end of the inputs to the post-processor and the beginning of computation.
8	N/A	Program stop

TABLE 5
FIELD INPUTS AND MNEMONICS

FIELD NOS.	I	II	III	IV	V	VI	VII	VIII	IX	X
TYPE OF INPUT										
PERSONNEL COUNT	1.	QTY	OFF/EM	GRADE#	OH	DED.	CREW	SUPT	MAINT	$/YR
PERS MULT A	2.	1.	EPCSR	EPCSC	CAOD	ARCEP	CRCEP	AROEP	CROEP	AOIC
PERS MULT B	2.	2.	ARCOO	CRCOO	AROO	CROO	TPPFO	TPPFE	ACQMUO	ACQMUE
PER MULT C	2.	3.	OPCSR	OPCSC	AMSC					
FUEL USE	3.	N/A	LIN#	AULIN	RFU	CF	FOL	N/A	N/A	QLIN
AMMUNITION	4.	CATAM	CAIAM	CARSUA	CAAPLA					
INSTRUMENT MISSILES	5.	*	*	*	*	*	*	*	*	*
ARTY/ORD	6.	1.	CTEAO	CAAPLO	CGRSAO	CRUFAO	CCSFAO			FPYAO
FOT	6.	2.	CTEFOT	CAAPLF	CGRSFO	CERUFO	CCSFOT	COPFOT		FPY/FOT
TOTAL CONTROL	7.									
STOP PROGRAM	8.									

TABLE 6

EXPLANATION OF MNEMONICS AND USE

MNEMONIC NAME	INPUT MEANING	FIELD ENTERED	FIELD I/II IDENTIFICATION
AMSC	Average annual $/person medical support	V	2/3
ACQMUE	Average annual $/enlisted man for quarters	X	2/2
ACQMUO	$/Yr./Officer or warrant/ quarters	IX	2/2
AOIC	$/Yr. other indirect cost	X	2/1
ARCEP	Attrition rate/Yr./enlisted crew	VI	2/1
ARCOO	Attrition rate/Yr./officers crew	III	2/2
AROEP	Attrition rate/Yr./Organization enlisted	VIII	2/1
AROO	Attrition rate/Yr./Organization and overhead officers	V	2/2
AULIN	Hours/Yr. energy using line item is used. Entire card refers to a specific item of equipment.	IV	3/
CAAPLA	$/Yr. for A.P.L. data takers/BN. assoc. with firings. (Ammo.)	V	4/
CAAPLF	$/Yr. for APL data takers/BN. assoc. with follow-on training firings	IV	6/2
CAAPLO	$/Yr. for APL data takers/BN. assoc. with firings (ARTY/ORD firing).	IV	6/1
CAIAM	$/Yr. associated with ammo firings for unit training	III	4/
CARSUA	$/Yr. for range support assoc. with ammo firings	IV	4/

SECTION 3 - LOGAM FEATURES (TOE O&S)

CATAM	$/Yr. for ammunition transport	II	4/
CCSFAO	$/Yr. for contractor support for ARTY/ORD firings	VII	6/1
CCSFOT	$/Yr. for contractor support for follow-on training firings	VII	6/2
CERUFO	$/Yr. use of eastern range for follow-on test firings	VI	6/2
CF	Cost of fuel in $/gal. Appears in same line (card or group of 10 as other inputs related to same device or vehicle)	VI	3/
CGRSAO	Cost ($) per ARTY/ORD firing for range support.	V	6/1
CGRSFO	Cost ($) per FOT firing for range support.	V	6/2
COPFOT	Cost ($) per FOT firing for overseas preparation of missile	VIII	6/2
CRCEP	Replacement cost for a crew enlisted men including training.	VII	2/1
CRCOO	Replacement cost per crew officer/warrant including training.	IV	2/2
CROEP	Replacement cost per organizational enlisted men including training.	IX	2/1
CROO	Replacement cost per organizational officer/WO including training	VI	2/2
CRUFAO	Cost of range use per ARTY/ORD firing.	VI	6/1
CTEAO	Cost of transport per ARTY/ORD firing.	III	6/1
CTEFOT	Transportation cost per FOT firing	II	6/2
EPCSC	Permanent change of station	IV	2/1

	change of station (times/yr.)		
CREW	Indication of assignment of individual represented by the line (card) to the crew. A "1" value means "crew", a "0" means other assignment.	VII	1/
DED	A "1" means individual(s) is dedicated, a "0" means not dedicated.	VI	1/
FOL	Fractional increase over fuel use to allow for oil and lube.	VII	3/
FPYAO	No. of ARTY/ORD firings/yr. for this organization.	X	6/1
FPYFOT	No. of FOT firings/yr. for this organization.	X	6/2
MAINT	A "1" indicates individual is assigned to maintenance function, a "0" indicates otherwise.	IX	1/
$/Yr.	Total pay and allowances for 1 person represented by this line.	X	1/
SUPT	A "1" indicates individual is assigned to the support.	VIII	1/
QTY	The number of people represented by this line.	II	1/
OFF/EM	A "0" indicates line represents an officer or warrant officer, a "1" represents enlisted personnel.	III	1/
GRADE	A numerical (real number) representation of the grade of the people represented by the line. (.5 represents warrant officer).	IV	1/
OPCSC	Permanent change of station cost per officer/WO.	IV	2/3
OPCSR	Rate (no. per yr.) of change of	III	2/3

	station for officers/WO		
QLIN	The quantity of the TOE line item using the fuel.	X	3/
RFU	Rate (gal. per hr.) of usage of the fuel by each of the devices using the fuel and represented by the entry.	V	3/
*	This entry (entries) provide a means of specifying the instrumentation costs incurred in firing a missile. Nine entries are possible. They will be added by the program. The line represents a type of missile.	II thru X	5/
LIN#	This provides an opportunity to enter a number identifying the line item of the TOE.	III	3/
OH	A "1" indicates that the people identified by the entry are to be considered as overhead. A "0" indicates otherwise.	V	1/
CAOD	$/Yr. for other direct costs of the organization.	V	2/1
TPPFE	Transients, patients and prisoners factor for enlisted men. Used to increase cost on basis of pay and allowances. A fraction	VIII	2/2
TPPFO	Transients, patients and prisoners factor for officers.	VII	2/2

3.6.2 TOE Operation and Support Cost

TOE costs are evaluated after LOGAM maintenance costs are totaled. These costs are totaled with the LOGAM outputs and printed using the DA PAM 11-4 format. The following discussion includes only those elements of the 11-4 format that are evaluated by the TOE processor. The mnemonics for the followiong CERs are described in Table 6.

3.6.2.1 3.011 Crew Pay and Allowances (PA)

Crew pay and allowances include base pay and allowances according to grade for military personnel whose primary function is to operate the weapon system being costed. For the purpose of LOGAM, personnel will be combined in groups identified by grade and their pay allowances. Personnel data are input for line items where T(1,J)=1 (J is the line item). Number of personnel, grades, type of personnel (crew), and pay and allowances are the inputs.

The CER for crew pay and allowances is:

PA=($/YR)*CREW*YR*QTY

PA is evaluated and accumulated into POUT(1) in Subroutine ONE when T(1,J)=1.

3.6.2.2 3.013 INDIRECT PAY AND ALLOWANCES

This cost element includes base pay and allowances for theater personnel who are not crew or maintenance personnel such as batallion/company/division commander or other overhead/dedicated support personnel. This cost is the product of the number of applicable personnel times their pay and allowances over the operational life (years) of the system being costed. The inputs for this cost element are similar to the inputs for crew (3.011) except that the dedicated/overhead support flags are set instead of the crew flag.

The CER for indirect pay and allowances is:

PAI=($/YR)*NCREW*YR*QTY

PAI is evaluated and accumulated into POUT(2) in Subroutine ONE when T(1,J)=1.

3.6.2.3 3.014 Permanent Change of Station (PCS)

PCS relates to the cost of replacement personnel to and from overseas theaters and within CONUS. LOGAM recognizes that change of station factors are different for enlisted personnel and officers and are not necessarily sensitive to grade level. This then reduces to two cost

estimating relationships (CERs):

```
PSCE=(QEPD+QEPM)*EPCSR*EPCSC*YR
PSCO=QO*OPCSR*OCPCS*YR
PCS=PCSE+PSCO
```

PCSO

PCSO

These CERs compute the PCS for enlisted (PCSE) and officer (PSCO) personnel. QEPM is the number of field level maintenance test and repair personnel computd by the LOGAM maintenance analysis. QEPM is totaled in Subroutine TWO from the PERS array that was evaluated in Subroutine TEMWF. The remaining parameters are inputs to the TOE data array (T).

Subroutine TWO is called to compute PCS when the user inputs T(1,J)=2 and T(2,J)=1. The cost for PCS is accumulated into POUT(3).

3.6.2.4 3.022 Petroleum, Oil and Lubricants (POL)

POL cost includes the system costs associated with the consumption of fuel and lubricants for a TOE line item (LIN). The CER for each line item that uses POL is of the following form:

```
CPOL=QLIN*AULIN*RFU*CF*FOL*YR
```

Subroutine THREE is called to compute POL costs when the user inputs T(1,J)=3. CPOL accumulations are made into POUT(4) for all POL line items input.

3.6.2.5 3.023 Unit Training Ammunition and Missile

Unit training, ammunition and missiles, includes the cost of ammunition and missiles consumed by the system being costed during unit training. Excluded is the cost of ammunition consumed durng small arms qualification. The costs for this element are computed and accumulated for all line items of this type in Subroutine FOUR and SIX.

When dealing with ammunition, a CER of the following form applies:

```
CAMMO=(CATAM+CAIAM+CARSUA+CAAPLA)*YR
```

Ammunition costs are evaluated in Subroutine FOUR when the user inputs T(1,J)=4. This cost is accumulated with the missile training costs which are evaluated in Subroutine SIX.

In general when dealing with missiles, there are two types of missile firings that are costed:

ARTY-ORD firings - CARORD
Follow-on Operational Test FOT firings - CFOT

and these result in two CERs as follows:

```
CARORD=(CATAO+CAMIAO+CARSUO+CACSAO+CAAPLO)*YR
CFOT=(CATFOT+CAPFOT+CAIFOT+CARSUF+CACSFO+CAAPLF)*YR
```

ARTY-ORD costs (CARORD) are evaluated in Subroutine SIX when the user inputs T(1,J)=6 and T(2,J)=1, where J is the line item number. The parameters for the CARORD CER that are not input directly are given below.

```
CATAO=CTEAO*FPYAO
CAMIAO=SIGIN*FPYAO
CARSUO=(CGRSAO+CRUFAO)*FPYAO
CACSAO=CCSFAO*FPYAO
```

SIGIN is the accumulation of instrumentation costs that were evaluated in Subroutine FIVE. The other parameters used in these equations are described in Table 5.

Follow-on-training costs are also computed in Subroutine SIX but the user inputs T(2,J)=2 to execute the code for this cost. The parameters for the FOT costs (CFOT) that are not input directly are listed below. These costs are accumulated with the ammunition and ARTY-ORD costs into POUT(5).

```
CATFOT=CTEFOT*FPYFOT
CAPFOT=COPFOT*FPYFOT
CAIFOT=SIGIN*FPYFOT
CARSUF=(CGRSFO+CERUFO)*FPYFOT
CACSFO=CCSFOT*FPYFOT
```

3.6.2.6 3.052 Other Direct Support

This element is a flexible category which can be defined to include any direct operating and support cost not included elsewhere. An example is civilian contractor maintenance for electric power for the system.

An equation of the following form will be added:

```
ODIR=CAOD*YR
```

The TOE portion (ODIR) of this cost element is evaluated in Subroutine TWO when the user inputs T(1,J)=2 and T(2,J)=1. The cost (SDA(10)) to reorder, store and administer a line item in the supply system is added to ODIR in Subroutine SEVEN. SDA(10) is the case accumulation of U17 which is evaluated in Subroutine LGCST. The costs for Other Direct Support are accumulated into POUT(6).

3.6.2.7 3.061 Personnel Replacement

This element includes the cost of training replacements including pay and allowances for trainees and instructors. Personnel replacement also includes recruiting costs for enlisted personnel, costs of in-processing and initial outfitting, and separation costs.

The differences between personnel replacement costs for enlisted personnel and officers is recognized. The difference between crew enlisted and other enlisted personnel is also taken into consideration.

The equation for enlisted personnel replacement cost is as follows:

CDEPRC=CEPRC+ODEPRC

where,

CEPRC=CEP*ARCEP*CRCEP*YR
ODEPRC=OEPLC*AROEP*CROEP*YR

In the previous 2 equations CEP is the number of crew enlisted personnel and OEPLC is the number of organizational level personnel less crew. These 2 values are accumulated in Subroutine ONE as ENLM(1) and ENLM(2), respectively.

CDEPRC is evaluated in Subroutine TWO when T(2,J)=1 is input. The program variable name for CDEPRC is POUT(7). CDEPRC will be accumulated with replacement cost for officers which is evaluated using the following equation:

RCO=CORC+OORC

where,

CORC=OCO*ARCOO*CRCOO*YR
OORC=OOLC*AROO*CROO*YR

OCO and OOLC are the number of operational crew officers and the number of organizational/overhead officers/WO less crew. These 2 values are accumulated in Subroutine ONE as O(1) and O(2), respectively.

RCO is evaluated in Subroutine TWO when T(2,J)=2 is input. RCO is accumulated into POUT(6) with the enlisted personnel costs (CDEPRC) and the maintenance enlisted personnel costs SDA(8) from LOGAM.

SDA(8) is the case accumulations of cost for training test and repair personnel. This cost is evaluated in Subroutine LGCST for each individual LRU case as U12. SDA(8) is summed with the TOE personnel replacement costs in Subroutine SEVEN.

3.6.2.8 3.062 Transients, Patients, Prisoners Costs (TPP)

This element includes the pay and allowances for personnel added to the Army strength over and above the table of organization and equipment/table of distribution and allowances (TOW/TCA) spaces. This arises because on permanent change of station between units a soldier is accounted for as a transient. Similar accounting provides for long-term hospital cases and for prisoners committed to the Retraining Brigade or the Displinary Barracks. Included in this cost are the following types of personnel:

Annual cost for dedicated organizational personnel.
Annual cost for maintenance personnel.
Annual cost for crew personnel.

The total cost for TPP is the sum of these

CTPP=(CTPPD+CTPPM+CTPPC)*YR

where,

CTPPD=(PADO*TPPFO)+(PADOEM*TPPFE)
CTPPM=(PADOM*TPPFO)+(PADMEM+CMPT/YR)*TPPFE
CTPPC=(PADOC*TPPFO)+(PACEM*TPPFE)

CTTP is evaluated and accumulated into POUT(8) in Subroutine TWO when T(2,J)=3 is input.

3.6.2.9 3.063 Quarters, Maintenance and Utilities

This element includes the cost of maintenance and utilities for personnel living in Government owned quarters (family quarters, bachelor officers quarters/bachelor enlisted quarters (BOQ/BEO), and barracks). The CER for this cost element is of the following form:

CQMU=CQMUO+CQMUE

where,

CQMUO=QO*ACQMUO*YR
CQMUE=(TNEM+CMPT/YR)*ACQMUE*YR

CMQU is evaluated and accumulated into POUT(9) in Subroutine TWO when T(2,J)=3 is input.

3.6.2.10 3.064 Medical Support Cost

This element includes the variable cost of medical and dental support rendered to military personnel. The CER for this cost element is of

the following form:

SCM=(QO+QEPD+QEPM)*AMSC*YR

QO and QEPD are the number of officers and non-maintenance enlisted personnel in the TOE organization. These personnel totals are accumulated in Subroutine ONE. QEPM is the number of field level test and repair personnel evaluated in the maintenance analysis (Subroutine TEMWF) of LOGAM.

SCM is evaluated and accumulated into POUT(10) in Subroutine TWO when T(2,J)=3 is input.

3.6.2.11 3.065 Other Indirect Cost

This element is a flexible category which can be defined to include any indirect operating and support costs not included elsewhere. Such costs will differ from system to system. This element includes the cost of general supplies to the force units which exist solely because of the system being costed. This element also includes any identifiable transportation cost (other than to and from Depot maintenance), such as special transportation of repair parts, secondary items, POL and ammunition. The cost of Program Offices or Product Improvement Offices if they exist is also included here. The cost of ammunition for small arms qualification is included. An equation of the following form is included to accommodate the operational OIC cost element:

OIC=AOIC*YR

OIC is evaluated and accumulated into POUT(11) in Subroutine TWO when T(2,J)=1 is input.

The output for this cost will include Other Indirect costs SDA(9) evaluated in the maintenance cost section of LOGAM. SDA(9) is the accumulated costs of housing test equipment (CFT) and shipping of test equipment to the field (CSHTF).

The LOGAM maintenance Other Indirect cost SDA(10) is summed with the TOE Other Indirect cost in Subroutine SEVEN.

SECTION 4

DETAILED PROGRAM DESCRIPTION

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LOGAM)

4.1 PROGRAM LOGAM

The main routine for the LOGAM program is also named LOGAM. Sometimes when a reference is made to "LOGAM" it is a reference to the entire program and at other times the reference is to the main subroutine. In most cases the reference is to LOGAM the main routine when used in section 4. The discussion for section 4.1 is the line by line or group of lines description for the main subroutine.

LOGAM is the executive routine that calls the program subroutines to input data, perform logistics computations and print results. Each major area of the routine's logic is described in subsections A thru I where each subsection is a logical division of the routine. A brief description of each subsection follows.

Section A. Assignment of variables to computer memory.

Section B. Intialization of internal variables and one-time inputs.

Section C. Reinitialization of program variables and case inputs.

Section D. Modification of case input values using sensitivity inputs.

Section E. Basic logistics flow, pipeline quantities and inherent availability.

Section F. Initial provisioning.

Section G. Computes spares, maintenance demands, costs and prints individual case outputs.

Section H. Resets work demand accumulators and certain cost factors.

Section I. Computes TOE costs and prints case and grand totals.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LOGAM)

A. This section of the LOGAM program assigns computer memory locations to input variables and to those variables that are evaluated in the program. These assignments are made by using labled COMMON, DIMENSION, and EQUIVALENCE statements.

1. PROGRAM LOGAM (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
TAPE1,TAPE17,TAPE18,TAPE8=/200)
This is the program statement for the main routine. The files used in the program are defined by this statement. TAPE5 is the input file from which all data inputs are read. TAPE6 is the output file from which all results are printed. TAPE17 and TAPE18 are files for storing summarized LRU results. TAPE1 is used to store the individual LRU data inputs and TAPE8 contains the annual maintenance manhours and support positions.

2. COMMON/POUT/POUT(13),AMULT
COMMON/OFF/OFF(6)
COMMON/ENLM/ENLM(7)
These common blocks are used in Subroutine OPER and the subroutines called by OPER to compute and print system support results. They are also used in the BLOCK DATA routine BLKDAT to initialize the variable lists.

3. COMMON/HEADER/DATE(5),TEXT(72),ANLYSIS(5),UNITIS(5),
REMARK(8),COSTIS(9),IPAGE
This common block contains the information to be printed as the header on each page of output.

4. COMMON/DAPAM/SDA(16)
This common block is used in Subroutine EIGHT to print System Maintenance and Operating Support costs.

5. COMMON/MNTFL/DSUE1,GSUE1,DUE1,DSUE2,GSUE2,DUE2,SUE1,
SUE2,SUE3,GSUO1,DUO1,GSUO2,DUO2,SUO1,SUO2,
SUO3,SUO4,SUO5,GSMO,SMO1,DMO,DMI,SMI1,SMI2,
SMI3,SMDI,SMD2,SUD1,DUI,SUI1,SUI2,FGE,FGO,
FGI,FGD,TUFTE,TUFTO,TUFTI,TUFTD,TUFEC,TUFOC,
TUFIC,TUFDC,TSME,TSMO,TSMI,TSMD,TMFO,TMFI,
TMFD,TMFEO,TMFOI,TMFID,TSPD,TSPI,TSPO,TSU,
TSLF,TSLD,F,FNG,A,TU,DPT
This common block contains the list of variables for the maintenance flow of LRUs, modules and parts through the test and repair facilities at each echelon. These variables are evaluated in Subroutine MFLOW and are used in Subroutines BASIC and MPROV to compute provisioning quantities.

6. COMMON/FRACT/SUEC,SUOC,SUIC,SUDC,SMEC,SMOC,SMIC,SMDC,
FUEC,FUOC,FUIC,FUDC,FMOC,FMIC,FMDC,
USE,URE,ULE,USO,URO,ULO,USI,URI,ULI,USD,URD,
MSO,MRO,MSI,MRI,MSD,MRD

```
REAL MSO,MRO,MSI,MRI,MSD,MRD
```

This common block contains the list of variables for the scrap and repair fractions that are evaluated in Subroutine RSFRCT. These variables are used in subroutine MFLOW to compute maintenance flows. The "REAL" statement types the MSO thru MRD variables as floating points (real).

7.
```
COMMON/INPUT/ARA,ARAD,AYZP,CAD,CALMAN,CALPUB,CALSET,
      CCAL,CCALP,CCALR,CCSP,CCSPP,CCSPR,CDDI,CDEO,
      CDFD,CDID,CDIO,CDIST,CDMAN,CDOE,CDOI,CDPMAN,
      CDPRMN,CDRMAN,CEMAN,CEN,CEND,CERMAN,
      CFTD,CGMAN,CGRMAN,CI,CII,CKIT,CKMD,CKME,CKMI,
      CKMO,CKPD,CKPI,CKPO,CKUD,CKUE,CKUI,CKUO,CLRUPG,
      CMODPG,CMP,CONMAN,CONTCT,CPE,CPI,CPII,CPP,
      CPUBII,CPUBV,CPV,CRI,CRII,CRM,CRP,CRU,CRV,
      CSDEP,CSDSU,CSESU,CSGSU,CTCPUB,CTRA,CTRAD,
      CTRCAL,CTRI,CTRII,CTRSPT,CTRV,CUBEM,CUBEP,CUBEU,
      CUCE,CUP,CV,DAOQL,DD,DDS,DI,DIS,DTE,DTI,DTO,
      E,EACAL,EACSP,ED,EDS,EE,EREI,ETE,ETEI,ETI,
      ETII,EVDM,EVDR,EVDT,EVEM,EVER,EVET,EVIM,
      EVIR,EVIT,EVOM,EVOR,EVOT,FE,FI,FII,FINT,
      FLM FMD,FMI,FMO,FN,FNGF,FNSP,FSA,FTI,
      FTII,FTM,FTP,FTU,FUD,FUE,FUI,FUO,G,H,
      HPM,HPP,HPU,IBG,IFLAG,IMF,INHIB,IO,
      IOPER,IS,JTED,NA,NB,NU,OD,ODS,OL,OST,
      OTF,P,PMR,PP,PPR,PUR,QMM,QMP,QMU,QTD,
      QTE,QTI,QTMD,QTME,QTMI,QTMO,QTO,QTPD,
      QTPI,QTPO,RDD,REO,REPEAT,RF,RID,ROI,SL,
      SMD,SME,SMF,SMI,SMO,SPE,SPEV,SPEVR,STAT,
      SUD,SUE,SUI,SUO,SVE,SVR,SVT,SVV,TALMAN,
      TAT,TATE,TAYZ,TC,TD,TDI,TDMAN,TDMW,TDPMI,
      TDPMII TDPRI,TDPRII,TDR,TDRMAN,TE,TEMAN,
      TENMAN,TEO,TER,TERMAN,TF,TFR,TGMAN,TGRMAN,TI,
      TID,TIMW,TIO,TIR,TMD,TMDD,TMDR,TMI,TMID,TMIR,
      TMO,TMOD,TMOR,TOE,TOI,TOMW,TONMAN,TRC,TUMD,
      TUMI,TUMO,WD,WDM,WDR,WE,WEM,WER,WI,WIM,WIR,
      WM,WMR,WMT,WO,WOM,WOR,WP,WTKIT,WU,YAT,YD,
      YMWO,YP,YR,YZ,ZFL,ZI,ZM,ZO,ZP,ZU,SENSY,T
```

This common block contains the list of input variables found in NAMELIST/L/. A description for these variables can be found in Section 5.2.

8.
```
COMMON/PROVQ/DELU,PDUE,PDUO,PDUI,PDUD,PQTU,PDME,PDMO,
      PDMI,PDMD,PQTM,PDPO,PDPI,PDPD,PQTP,PQTE,
      PQTO,PQTI,PQTD,PQT,QT,QTM,QTP
```

This common block contains the provisioning quantities for LRUS, modules, and parts. Variables PDUE through PQTP are the undistributed quantities that are computed as a function of the maintenance flows. Variables PQTE through PQT are the undistributed LRU provisions that are modified in Subroutine AVAIL when enforcement of an

operational availability is requested. QT,QTM and QTP are the totals after distributing the provisions equally over stock points at each echelon.

9. COMMON/REPSP/QB,QC,QCD,RU,RUD,QBM,QCM,QCMD,QQCM,
RM,RMD,RQMN,QBP,QCP,QCPD,QQCP,RP,PRD,RQPN
This common block contains the reorder quantities, consumed material and end of program residuals. The values for these variables are evaluated in Subroutine REORDR.

10. COMMON/MMHWF/AEY,AERY,AOY,AORY,AIY,AIRY,ADY,ADRY,
SAEY,SAERY,SAOY,SAORY,SAIY,SAIRY,SADY,SADRY,
CAEY,CAERY,CAOY,CAORY,CAIY,CAIRY,CADY,CADRY,
ESU,ESUM,ESUR,ESUY,ESURY,DSU,DSUM,DSUR,DSUY,
DSURY,GSU,GSUM, GSUR,GSUY,GSURY,DEP,DEPM,DEPR,
DEPY,DEPRY, AME,SAME,DSIM,ESI,ESME,DEPAIE,
DEPAIM,DEPAIY,DEPAR,DEPARY
This common block contains the manhours, manpower, and test equipment demands for maintenance at all echelons. These variables are computed in Subroutine TEMWF.

11. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
This common block contains the tape or disc IO file names. The values for these files are initialized in the Block Data routine.

12. COMMON/LRU/LRU,NRU,ILRU,NLRU,ILRU2,NLRU2
This common block contains LRU case counters. These variables are all input or set in the main program.

13. COMMON/SUMLRU/TLRU(15),NDLRU,ICN,ISET,IATE,IA7,IA8
This common block contains the information for summarizing LRU cases. TLRU and NDLRU are inputs of the 8th "one-time" data card. TLRU is a 60 character description of the summarization process and NDLRU is the number of distinct LRUs to be summarized. The remaining parameters are LRU counters and disc file units used in Subroutine SUMRY to perform the summarization.

14. COMMON/BAS/QUE,QUO,QUI,QUD,QUDH,QFE,QFO,QFI
QYE,QYO,QYI,QYD,QYF,QME,QMO,QMI,QMD,QMDH
QPO,QPI,QPD,QPDH,QFME,QFMO,QFMI,QFMD
This common block contains the scrap and repair quantities tied up in the pipelines. These variables are evaluated in Subroutine BASIC. When Subroutine BASIC is called from LGCST, these variables will represent shipping cost factors.

15. COMMON/COSTS/CET,CFT,CIVT,CMPT,CQTE,CQTO,CQTI,CQTD,
CQTME,CQTMO,CQTMI,CQTMD,CQTPO,CQTPI,CQTPD,

CQTT,CQTMT,CQTPT,CRUT,CRMT,CROT,CSAT,CSHT,
CTST,CWHT,DELTA,ECMPT,EPVGCT,GCT,PDELTA,
PVGCT,

This common block contains the individual LRU case costs that are evaluated in Subroutine LGCST. These are the variables that are output in Subroutine IDVOP.

16. COMMON/ZERO/CCET,CCTS,CCTSR,CCF,CCM,CCMF,CCMD,
CCMFD,CTRF,CTRDEP,CTR,CIV,CIVREC,CRT,CWH,
CSA,CSAREC,CSH,CGT,CTREC,PCD,CQTU,PCP,CQTM,
PCR,CQTP,PCS, CQTUMP,PCGT,SEMPT,SEPC,SPCR
SEPV,SDEL,SPDEL,CAYZ(10),CAYZI(10)

This common block contains the cost elements accumulated for each LRU into a case total. The elements in the common block are computed in Subroutine LGCST with the exception of CAYZ, CAYZI. The system availability arrays are evaluated in LOGAM.

17. COMMON/SENS/NRULE(12),NVAR(12),MODE,KPASS,NPASS

This common block contains the list of variables that are used to perform sensitivity analyses. SENS is referenced in Subroutine SENSIT to print a page header when the output page is for a sensitivity analysis.

18. COMMON/PERSN/PERS(5,2),PERL(5,2),HPD(5,2),
WPD(5,2),WPDL(5,2)

This common block contains case concept accumulated totals for test and repair personnel requirements. These parameters are evaluated and output in Subroutine PERSON with the exception of HPD which is accumulated in Subroutine TEMWF.

19. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,
FMWO,OR,AYZIS,AYZOS

This common block contains a group of miscellaneous variables that are used in various subroutines. AAIE and FMWO are computed in TEMWF. AAIE is set to 1 when Type II test equipment is used at the Depot. FMWO is the frequency of modification work orders. EDEE, YR8, and SAVE are set in LOGAM. EDEE is the number of LRUs installed at all equipment locations. YR8 is the number of hours in the O&S phase. SAVE is used to convert units computed as a function of clock time to units that are a function of the operational time of the system. STI and STII are set in Subroutine SUMRY to the values for Type I and Type II posting flags to print summarized totals if the posting flags are ever turned on. QUA is the initial production quantity computed in Subroutine REORDR for LRUs. OR, AYZIS and AYZOS are outputs of AVAIL. OR is the over-ride factor (fractional part of AYZP) for computing operational availability. AYZIS and AYZOS are the computed

inherent and operational availabilities, respectively.

20. DIMENSION SAVM(16),SAV(326)
DIMENSION CUM(55),SUM(35)
DIMENSION OSAV(326)
DIMENSION SDAM(16)
These statements set the dimensions for variables used internal to the main routine (LOGAM). Reference the next 3 statements for a description of SAVM, SAV and CUM. SUM and SDAM are used to accumulate grand total costs. OSAV is used to save an LRU input case that can be later used to reset the /INPUT/ common block.

21. EQUIVALENCE (CCET,CUM(1))
This statement equivalences the case total accumulator array CUM to the variable list of common block /ZERO/.

22. EQUIVALENCE (ARA,SAV(1))
This statement equivalences the variable list of common block /INPUT/ to the working array SAV.

23. EQUIVALENCE (SAEY,SAVM(1))
This statement equivalences the working array SAVM to 16 variables in COMMON/MMHWF/ beginning with SAEY.

B. This section of LOGAM initializes grand total accumulator arrays, sets up namelist debugging lists and reads data that will be common to all LRU data cases.

1. DATA SUM/35*0./
DATA SDAM/16*0./
These statements initialize accumulator arrays for grand totals and maintenance support costs.

2. NAMELIST/BUG/
This NAMELIST name with its variable list is used for output purposes only as a debugging device. The variable list is not included here.

3.
```
      READ (K003FX,1000) TEXT,ANLYIS,DATE,COSTIS,AMULT
1000  FORMAT (18A4/18A4/18A4/18A4/5A4/5A4/9A4,
          5X,F10.5)
```
These statements read and set the format for the first set of data cards to be input. Seven cards are input with this format. Four cards for TEXT, one card each for ANALSIS and DATE, and one card containing COSTIS and AMULT. AMULT is the cost conversion factor.

4.
```
      READ (K003FX,8004)TLRU,IRUN,NDLRU
8004  FORMAT (15A4, 2I10)
```
These statements read and set the format for the second set of card inputs. Sixty (60) alphanumeric characters are input for TLRU and an integer value each for IRUN and NDLRU. IRUN determines if subroutine FINLE is used to preprocess namelist inputs for errors only or to process each case and execute. IRUN=0 preprocesses all the data cases without execution. IRUN=1 will process each data case for errors and execute the case even if errors are found. NDLRU are the distinct number of LRU cases for which a summarization is to be performed.

C. This section of LOGAM resets program flags, cumulator arrays and reads an LRU data case.

1. 9 ASSIGN 1 tO KAD
This statement is the beginning statement for a completely new set of data. The value of one is given to KAD. KAD is checked on in the sensitivity section. When all sensitivity functions are completed KAD will be given the value 9 and a return made to this statement.

2.
```
LRU=0
ILRU=1
ILRU2=1
```
These statements reinitialize LRU counters. ILRU is the first LRU position in a group of LRUs that are shared for maintenance computations of Type I test equipment. ILRU2 is the same as ILRU except it is used for Type II equipment.

3. 1 IF(IS.EQ.0)GO TO 64
A return is made to this statement after each individual LRU case is completed. IS is zero on the first pass only. This statement by-passes setting the input array SAV on the initial pass through the program. On the first pass SAV contains the default values set in the block data routine.

4.
```
   IF(IS.NE.1)GO TO 52
   DO 53 I=1,326
53 SAV(I)=OSAV(I)
```
These statements recall LRU inputs from a previous read when IS=1. IS must be input with the previous data case to enter this logic. OSAV is the first data case unless the user inputs NB=0 on a subsequent case.

5. 64 IS=3
This statement controls the setting of accumulated cost values and resetting availability, workload, and case total accumulators. IS=3 is set to prevent resetting accumulators until IS=1 is input at the end of a case (concept).

6.
```
    DO 631 I=1,35
631 CUM (I)=0.
```
These statements reset the area of common block/ZERO/ where accumulated values over a concept are additive. This includes the case total costs CCET through SPDEL.

7.
```
    DO 581 I=1,16
    SAVM(I)=0
581 SDA(I)=0
```

These statements reset the maintenance support cost array before accumulating data for a new case total.

8.
```
    DO 641 I=36,55
641 CUM(I)=1.
```
These statements reset the area of common block /ZERO/ where accumulated values over a concept are multiplicative. This includes the CAYZ and CAYZI arrays.

9.
```
    DO 651 I=1,5
    DO 651 J=1,2
651 HPD(I,J)=0.
```
These statements reset the test and repair manhour accumulator arrays for a new case total.

10.
```
52 CONTINUE
IF(IS.EQ.2)GO TO 50
   DO 51 I=1,20
51 G(I)=0.
50 CONTINUE
```
These statements will reset the maintenance policies for the next LRU unless IS=2 is input.

11.
```
65 CONTINUE
```
This statement is the end of the logic for initializing and resetting totals and accumulator arrays.

12.
```
QTE=0.
QT0=0.
QTI=0.
QTD=0.
```
These statements initialize the initial provisions for LRUs at the Organizational, Direct Support, General Support, and Depot supply facilities, respectively.

13.
```
QTME=0.
QTM0=0.
QTMI=0.
QTMD=0.
```
These statements initialize the initial provisions for modules at the Organizational, Direct Support, General Support and Depot supply facilities, respectively.

14.
```
QTPO=0.
QTPI=0.
QTPD=0.
```
These statements initialize the initial provisions for parts at the Direct Support, General Support and Depot supply facilities, respectively.

15.
```
    READ (K003FX,1101) UNITIS
```

```
     IF(EOF(K003FX))3,10
1101 FORMAT (5A4)
  10 CONTINUE
```

These statements read the 9th input card and checks for an end of file on the card input device. UNITIS is used as an output descriptor of the LRU class and number for the next LRU to be processed. If an end of file is encountered instead of UNITIS data, a transfer is made to statement 3 for a program stop. Otherwise, transfer is to statement 10 for additional inputs.

16.

```
     READ (K003FX,1001) REMARK
     IF (EOF(K003FX))3,13
1001 FORMAT(12A6)
     13 CONTINUE
```

These statements input the 10th card of data containing 72 characters of information. REMARK is used to describe the next LRU scheduled for input. If an end-of-file is encountered instead of REMARK then transfer is made to statement 3 for a program stop. Otherwise, transfer is made to statement 13 for continuation of LRU inputs.

17.

```
CALL FINLE(IRUN,K003FX,K004FX,IEND)
IF(IEND.EQ.1)STOP
```

This statement calls Subroutine FINLE to read LRU data from the NAMELIST/L/ input statements. IRUN=0 will process the data for errors only. IRUN=1 will process a data case and return to the next statement and execute the case even if errors are found. If an end-of-file is read from the input file (K003FX) program execution is terminated.

18.

```
IF(NU.EQ.-4)STOP
```

This statement is the normal way to terminate execution of the program.

19.

```
   IF(NB .NE. 0)GO TO 54
   NB=1
   DO 55 I=1,326
55 OSAV(I)=SAV(I)
54 CONTINUE
```

These statements save the data of COMMON/INPUT/ (as modified by the last NAMELIST/L/ input) for possible recall on a later pass. SAV and common block INPUT share the same locations in central memory. NB is set equal to 1 to prevent resetting of OSAV on subsequent LRU cases unless the user wishes to reset the array by inputting NB=0 in a namelist /L/ data case.

20.

```
IF(SENSY(1).NE. 0.)GO TO 11
```

This statement transfers control to the sensitivity

section of the program. SENSY data is read with the NAMELIST/L/ input list. SENSY(1) is the number of variables in the input list that will undergo modifications for a sensitivity run. SENSY data should be input only after all LRUs in a concept have been processed.

21.
```
IF(QTE.GE..5)H(1)=1.
IF(QTO.GE..5)H(2)=1.
IF(QTI.GE..5)H(3)=1.
IF(QTD.GE..5)H(4)=1.
IF(H(4).LT..5)HPU=0.
```
These statements check the NAMELIST/L/ input values for predetermined quantities of LRU stock at the Equipment, Direct, General and Depot facilities. If quantities are input (>.5) at a facility, then the stock authorization flag is stored in the appropriate H location. Discretionary procurement holding time for LRUs (HPU) is set to zero when there is (not) LRU stock at the Depot.

22.
```
    DO 150 I = 1,4
    IF (H(I).GE.0.5)GO TO 151
150 CONTINUE
    H(4)=1.
```
These statements check the stock authorization array (H) to determine if stock is authorized for at least one stock level. If a stock level was not authorized than the program sets authorization at the Depot.

23.
```
151 GTOT=0.
    DO 152 I = 1,20
152 GTOT=GTOT+G(I)
```
These statements sum the maintenance policy fractions (G). The sum of maintenance policies input must sum to 1. GTOT is used to check if maintenance policies sum to unity.

24.
```
IF (GTOT.GT..9999.AND.GTOT.LT.1.0001)GO TO 157
```
This statement checks for maintenance policies that do not sum to 1 (100% of maintenance). If the total is 1.0 then logic is transferred to statement 157, otherwise the following statements are executed.

25.
```
    WRITE (K004FX,153)
153 FORMAT(1H1,93H,***MAINTENANCE POLICIES
    DO NOT SUM TO 100 PERCENT POLICIES MODIFIED
    AS A RATIO OF TOTAL***//
    1X,6HPOLICY, 3X,5HINPUT, 3X,8HMODIFIED)
    DO 154 I = 1,20
    BG=G(I)
    G(I)=BG*(1.0/D(GTOT))
```

```
154 WRITE (K004FX,155)I,BG,G(I)
155 FORMAT(3X,I2,5X,F5.2,5X,F5.2)
    WRITE (K004FX,156)GTOT
156 FORMAT(10X,5H-----/10X,F5.2)
```

These statements are executed when the G policies do not sum to 1.0. A message informing the user of the error is printed and the input policies are prorated as a function of the input total (GTOT). The modified G policies that now sum to 1.0 are printed along with policies input (BG). The total (GTOT) of the G policies that are input is also printed.

26. `IF(GTOT.EQ.0.0) G(20)=1.0`
`IF(GTOT.EQ.0.0) WRITE(K004FX,1003)`
These statements set the mainentance policy GT=1.0 if the user fails to input policies and writes a message instructing the user to this effect.

27. `157 CONTINUE`
This is the statement transferred to when the maintenance (G) policies sum to 1.0.

28. `IF(LRU.EQ.0)REWIND ND`
`LRU=LRU+1`
`NLRU=LRU`
`NLRU2=LRU`
`NRU=0`
These statements rewind the individual LRU data file on the first pass of a LRU concept. Each LRU data record in a concept will later be written to file ND. LRU is then incremented for each NAMELIST/L/ data record input. LRU is reset to zero for a sensitivity run. NLRU is the last LRU case accumulated when a Type I maintenance data is shared between LRUs. NLRU2 is similar to NLRU except it is used with Type II equipment.

29. `WRITE(ND)SAV,UNITIS,REMARK`
This statement writes the NAMELIST inputs (SAV) and LRU descriptions to data file ND. ND is the file read in the sensitivity section of this program to retrieve the data for each individual LRU case. The ND file is also used to print formatted namelist in subroutine INPOP.

30. `IF(IO.EQ.0)GO TO 12`
`CALL INPOP`
These statements are used to print the NAMELIST inputs in a formatted form. Subroutine INPOP is called to print the NAMELIST inputs when IO is input to a non-zero value.

31. `GO TO 12`

This statement skips to the section of the program that begins the logistics computation.

D. This section of LOGAM is the sensitivity section. After all LRUs in a concept are processed, inputs to the SENSY array can alter selected parameter values. Modifying the original LRU values with the SENSY inputs is performed in this section.

1. 11 CONTINUE
   This is the transfer statement after a sensitivity request is made. Transfer to this statement is described in line item C-20 above.

2. IF(LRU.EQ.0)LRU=NRU
   This statement is required only if sensitivity cases are stacked (more than one case). When a sensitivity case is completed, program transfer is made to statement number 1 where LRU is reset to zero before reading the next data case. If another sensitivity case is input then NRU will still have the value for the number of LRUs processed in the last sensitivity case; whereby LRU can be properly reset.

3. IF(LRU.EQ.0) GO TO 91
   This statement transfers logic to a diagnostic statement when LRU=0. The only time this should happen is when SENSY(1) is input with the first LRU data case and has a value greater than zero.

4. MODE=SENSY(1)+.1
   This statement sets the number of variables in NAMELIST/L/ that are to be modified. The ".1" is to solve round-off problems when converting from floating point to integer.

5. NPASS=SENSY(2)+.1
   This statement sets the number of variations for a selected set of variables (NVAR) to be modified. Essentially this is the number of passes that will be made through the LRU data base of file ND. Each pass modifies LRU data values according to the rules (NRULE) which are explained later.

6. IF ((MODE.GT.12).OR.((2+(MODE*(NPASS+2))).GT.266))
   GO TO 19
   This statement checks for the number of values that can be input to the SENSY array without overflowing the dimensioned value (266) of the array. If overflow occurs transfer is made to a diagnostic statement (19).

7. DO 6 I=1,MODE
   NRULE(I)=SENSY(I+2)+.1
   MODEI2=MODE+2+I
   6 NVAR(I)=SENSY(MODEI2)+.1
   These statements set the rules (NRULE) for variable

modification and the variable's position (NVAR) in the labeled common block/INPUT/. The modification rules are explained later in line item H-21.

8. IF (INHIB.EQ.1) CALL PAGE
This statement positions the printer paper to the top of a page and prints header information when individual LRU printout is inhibited.

9. KPASS=0
This statement initializes the pass counter.

10. 7 KPASS=KPASS+1
This statement counts the number of passes through an LRU data set. Each pass is a variation to the NVAR set of inputs. When all the LRUs in a data set have been processed, return is made to this statement unless all input variations have been exhausted (KPASS=NPASS).

11. REWIND ND
This statement rewinds the LRU data file.

12. NRU=0
ILRU=1
ILRU2=1
These statements initialize a counter (NRU) for the number of LRUs analysed in a sensitivity pass and the initial LRU accumulated when sharing Type I (ILRU) and Type II (ILRU2) maintenance data between LRU cases.

13. 8 NRU=NRU+1
NLRU=NRU
NLRU2=NRU
This statement counts the number of LRUs that have been modified and processed. Return will be made to this statement after each LRU is processed until NRU=LRU, where LRU is the number of individual LRU data cases analysed in the baseline case. NLRU is set here as the last case number for a group of LRU's whose Type I maintenance data is stored and posted in the individual LRU case printouts. NLRU2 is similar to NLRU except it is used when Type II equipment is posted.

14.
```
    IF (IS.NE.1)GO TO 58
    IS=3
    DO 611 I=1,35
611 CUM(I)=0.
    DO 582 I=1,16
    SAVM(I)=0.
582 SDA (I)=0.
    DO 621 I=36,55
621 CUM (I)=1.
```

```
    DO 661 I=1,5
    DO 661 J=1,2
661 HPD(I,J)=0.
 58 CONTINUE
```

These statements reinitialize the cumulative data arrays. Their use and descriptions are the same as described in line items C-4. through C-9.

15. IF(NRU.LT.LRU) ASSIGN 8 TO KAD
This statement assigns the value 8 to the logic transfer variable KAD until all LRUs of the baseline case have been processed. KAD=8 sends control to statement number 8 after a sensitivity pass through the baseline case is completed.

16. IF (NRU.EQ.LRU) ASSIGN 7 TO KAD
This statement assigns the value 7 to the logic transfer variable KAD when the last LRU in a data set is processed. KAD=7 sends control to statement number 7 for the next sensitivity pass of the LRUs.

17. IF (NRU.EQ.LRU.AND.KPASS.EQ.NPASS) ASSIGN 9 TO KAD
This statement assigns the value 9 to the logic transfer variable KAD when all sensitivity passes for all LRUs have been exhausted. KAD=9 sends control to statement number 9 to reinitialize program variables and input a new data set.

18. READ (ND) SAV, UNITIS, REMARK
This statement reads the NAMELIST inputs for the current LRU case to be modified for a sensitivity run. The LRU units and remarks information are also read from the file.

19.
```
   I=0
80 I=I+1
```

These statements initialize and set the counter for selecting the modification rule and the position of the variable (within the /INPUT/ list) to be modified.

20.
```
KRULE=NRULE(I)
IF(KRULE.LT.1.OR.KRULE.GT.5)GO TO 300
```

These statements set the rule number for modifying a variable and checks if it is an acceptable number. There are only 5 possible rules; replacement, addition, subtraction, multiplication and division. If any other rule number is input a transfer to a diagnostic statement is made.

21. M=NVAR(I)
This statement selects the position within the COMMON/INPUT/ variable list for the Ith variable to be

modified.

22. INDEX=I+2+((KPASS+1)*MODE)
This statement computes the position within the SENSY array that contains the value used to modify the LRU data variable represented by location M.

23. IF(M.LT.1.OR.M.GT.326)GO TO 200
This statement transfers to a diagnostic statement if the variables position is not within the first 326 positions of COMMON/INPUT/.

24. VALUE=SENSY(INDEX)
This statement stores the modification value for the Ith MODE and RULE.

25. GO TO (81,82,83,84,85),KRULE
This statement sends control to the program logic associated with the rule currently in effect (KRULE)

26.
```
81 SAV(M)=VALUE
   GO TO 86
82 SAV(M)=SAV(M)+VALUE
   GO TO 86
83 SAV(M)=SAV(M)-VALUE
   GO TO 86
84 SAV(M)=SAV(M)*VALUE
   GO TO 86
85 IF(VALUE.EQ.0)GO TO 100
   SAV(M)=SAV(M)/VALUE
```
These statements modify the inputs (SAV) for an individual LRU baseline case according to the value of KRULE. The rule for modification is; KRULE=1, replace the old value with new VALUE; KRULE =2, add old value to VALUE; KRULE=3, subtract old value and VALUE; KRULE=4, multiply old value by VALUE; KRULE=5, divide old value by VALUE. If VALUE=0 and KRULE=5 a transfer is made to a diagnostic statement.

27. 86 IF(I.LT. MODE)GO TO 80
This statement checks to determine if all the sensitivity modes have had their values modified.

28. IF(INHIB.EQ.1)CALL SENSIT (SAV)
This statement calls Subroutine SENSIT to print a description of the sensitivity run when the individual LRU print flag is turned off.

29. IF(NRU.LT.LRU)GO TO 12
IF(KPASS.LT.NPASS)GO TO 12
These statements transfer control to the logistics computation section of this program until all

sensitivity passes for all individual LRU cases are processed.

30.
```
   DO 90 J=1,INDEX
90 SENSY(J)=0.
   GO TO 12
```
These statements reinitialize the SENSY array when the last pass of the last LRU is reached. Transfer of control is then made to the logistics section to compute the last sensitivity pass.

31.
```
100 WRITE (K004FX,102)INDEX
102 FORMAT(1X,40HATTEMPTED DIVISION BY
          ZERO, ERROR INDEX=,I5)
    GO TO 86
```
These statements print a diagnostic message when the value input to SENSY for a division rule (KRULE=5) is zero and then sends control to check for the presence of another variable to modify.

32.
```
200 WRITE (K004FX,202) M
202 FORMAT (1X,26HILLEGAL VARIABLE ADDRESSED,I4)
    GO TO 86
```
These statements print a diagnostic message when the variable selected for modification is not one of the permitted ones; i.e., variables that occupy locations 1 through 326 of COMMON/INPUT/. Control is then sent to statement 86 to test for the presence of another variable to modify.

33.
```
300 WRITE (K004FX,302) KRULE
302 FORMAT (1X,19HILLEGAL RULE KRULE=I5)
    GO TO 86
```
These statements print a diagnostic message when a modification rule is not one of the permitted ones; i.e., KRULE is not 1 through 5. Control is sent to statement 86 to test for the presence of another variable to modify.

E. This section of LOGAM begins the basic logistics flow on a per LRU per hour basis. This section contains equations and operations in which the demands that each LRU type makes on the support system and the costs consequently incurred are calculated.

1. 12 CONTINUE
This statement is the starting control point for logistics computations for all LRU types.

2. CALL RSFRCT
This statement calls Subroutine RSFRCT to compute the fraction of LRUs and modules that are either repaired or scrapped or in "limbo". The outputs of this subroutine are used in Subroutine MFLOW in the computation of maintenance flows.

3. CALL MFLOW
This statement calls Subroutine MFLOW to compute maintenance flows. The outputs of this routine are used in Subroutines BASIC and MPROV to compute provisioning.

4. REOT=24.*REO+TOE
ROIT=24.*(ROI+TIO)
RIDT=24.*(RID+TDI)
These statements compute the hours of supply for condemned modules at the equipment, and condemned modules and parts at Direct Support and General Support levels, respectively.

5. TEOT=TEO+TOE
This statement computes the two-way pipe time in hours for the Equipment/Direct Support pipe.

6. TOIT=24.*(TOI+TIO)
This statement computes the two-way pipe time in hours for the Direct Support/General Support pipe.

7. TIDT=24.*(TID+TDI)
This statement computes the two-way pipe time in hours for the General Support/Depot pipe.

8. HF=(24.*HPU)+(168.*FTU)
HM=(24.*HPM)+(168.*FTM)
HP=(24.*HPP)+(168.*FTP)
These statements compute delay times in hours from procurement start until delivery of LRUs, modules and parts after factory restart.

9. CALL BASIC
(168.*FTM,168.*FTP,168.*FTU,HF,HM,HP,

```
     24.*RDD,REOT,ROIT,RIDT,TUMD,
     TUMI,TUMO,TRC,TEOT,TOIT,TIDT,1)
```

This statement calls Subroutine BASIC to compute the quantity of LRUs, modules and parts "down" or tied up (in the pipelines) per hour per LRU location.

10. `CALL AVAIL (1,AYZIS,OR,SAVE)`
This statement calls Subroutine AVAIL to compute inherent availability.

11.
```
    DO 933 I=1,NA
933 IF(TAYZ(I).EQ.1)CAYZI(I)=CAYZI(I)*AYZIS**REPEAT
```
This DO loop tallies inherent availabilities for subsystems of LRUs. The system availability is tallied in the first location and each subsystem's tally will be recorded in each subsequent location. The LRU subsystem structure is determined by the location of the TAYZ inputs within the LRU data deck (NAMELIST/L/). CAYZI is used to print out subsystems availabilities on the case total output page.

12. `SAVE=ED*OTF*AYZIS`
This statement computes the "real time" operating factor for all equipment installations (ED) as a function of inherent availability (AYZIS) and the operating time (OTF) factor.

13.
```
QUE=QUE*SAVE
QUO=QUO*SAVE
QUI=QUI*SAVE
QUD=(QUD+QUDH)*SAVE
```
These statements take into account the "real time" operating factor based on inherent availability to re-compute scrap LRU quantities tied up in the pipes; Equipment/Direct Support, Direct Support/General Support, General Support/Depot and Depot/Factory, respectively. The Depot/Factory quantities are modified with the quantity (QUDH) contributed to procurement holding time.

14. `QU=H(1)*QUE+H(2)*QUO+H(3)*QUI+H(4)*QUD`
This statement computes the total quantity of LRU scrap tied up in the pipes, dependent upon the presence of supply at any of the support locations. If either H(1), H(2), H(3) or H(4) is zero then there is no authorized stock at that location.

15.
```
QFE=QFE*SAVE
QFO=QFO*SAVE
QFI=QFI*SAVE
```
These statements take into account the "real time" operating factor based on inherent availability to

redefine the quantity of LRUs tied up in float repairs for the pipes; Equipment/Direct Support, Direct Support/General Support and General Support/Depot, respectively.

16. QME=QME*SAVE
QM0=QMO*SAVE
QMI=QMI*SAVE
QMD=(QMD+QMDH)*SAVE
These statements take into account the "real time" operating factor based on inherent availability to redefine quantities of modules tied up in scrap for the pipes; Equipment/Direct Support, Direct Support/General Support, General Support/Depot and Depot/Factory, respectively. The Depot/Factory quantities are modified by the quantity of modules (QMDH) tied up in procurement holding time.

17. QM=QMO+QMI+QMD+QME
This statement totals the quantity of module scrap tied up in all pipes.

18. QFME=QFME*SAVE
QFMO=QFMO*SAVE
QFMI=QFMI*SAVE
QFMD=QFMD*SAVE
These statements take into account the "real time" operating factor based on inherent availability to redefine the quantity of modules in float repairs for the pipes; Equipment/Direct Support, Direct Support/General Support, General Support/Depot and Depot/Factory, respectively.

19. QPO=QPO*SAVE
QPI=QPI*SAVE
QPD=(QPD+QPDH)*SAVE
These statements take into account the "real time" operating factor based on inherent availability to redefine the quantity of parts tied up in scrap for the pipes; Direct Support/General Support General Support/Depot, and Depot/Factory, respectively. The Depot/Factory quantities are modified by the quantity of parts (QPDH) tied up in procurement holding time.

20. QP=QPO+QPI+QPD
This statement totals the quantity of parts tied up in scrap for all pipes.

21. IF(IBG.EQ.1) WRITE (6,BUG)
This statement prints the list of variables in NAMELIST/BUG/ when debugging of a data case is required.

F. This section of LOGAM computes the initial provisioning for LRUs, modules, and parts. There are three methods of determining initial stock. The user may predetermine the supplies by inputting the stock through NAMELIST/L/, use LOGAM supply rules or LOGAM maintenance rules. When using either the supply or maintenance rules any computed value for stock can be over-ridden by inputting the stock value greater than or equal to 1.

The stock option flag AYZP is used to select the method of determining initial provisions. A negative value for AYZP selects the predetermined option. The program under this option will use the stock input, except in the event there are not an equal number of provisions for each stock point. The program will distribute the stock equally over all stock points and round the value to a whole number according to the "Z" rounding factors. This distribution and rounding process is also performed on the quantities computed with both the supply and maintenance rules.

With AYZP input less than one but greater or equal to zero, the supply rules are selected to compute initial stock. The supply rules use the repair and scrap pipeline quantities evaluated in Subroutine BASIC to compute the initial stock. Subroutine IOL is called with the BASIC quantities where they are modified by a safety stock coefficient. If AYZP contains a fractional part, the fractional part will be used in Subroutine AVAIL to force an operational availability toward the fractional value. AVAIL is programmed to ignore any fractional input less than .5 (50% availability).

LOGAM maintenance rules are selected when AYZP is greater than or equal to one. The fractional part of AYZP is used to drive operational availability as was discussed above. The maintenance flows evaluated in Subroutine MFLOW are used in Subroutine MPROV to compute the initial provisions. The computed quantities from MPROV are distributed and rounded as they are for the other provisioning options.

Subroutine REPLC is called after the provisioning options have been executed to replace a computed value with any predetermined value input by the user. The computed values with any predetermined replacements are distributed and rounded in Subroutine STKDS. If the predetermined stock option is selected (AYZP<0.) then all the computed values will be replaced by the input values.

1. If (AYZP) 580, 712, 715
   This statement transfers logic to either the predetermined value logic (statement 580), to the LOGAM Supply Rule logic (statement 712) or to statement 715 to determine if the option is to force availability

on Supply Rule stock or if its for the Mainentance Rule logic. The provisioning logic to be used depends on the value input for AYZP. AYZP<0 selects the predetermined value logic, AYZP>0 (but less than 1) selects the LOGAM Supply Rule. AYZP≥1 selects the LOGAM Maintenance Rule.

2. 715 IF (AYZP-1.) 712,713,713
This statement is used to check for the option to force operational availability when using the LOGAM Supply Rule. If the value of AYZP is less than one, then the option is to use Supply Rules (statement 712) and to force operational availablity to the fractional part of AYZP.

3. 712 CONTINUE
This is the statement transfered to when LOGAM Supply Rules are requested (0≤AYZP<1).

4. CALL IOL (EDS,CKUE,QUE,QFE,PDUE)
CALL IOL (ODS,CKUO,QUO,QFO,PDUO)
CALL IOL (DIS,CKUI,QUI,QFI,PDUI)
CALL IOL (DDS,CKUD,QUD,0.,PDUD)
These statements use the LOGAM supply rules to compute the initial LRU provisions at the Equipment, Direct Support, General Support and Depot levels, respectively. EDS is the number of supply points,(CKUE is a safety stock coefficient, and QUE and QFE are scrap and repair quantities tied up in the supply pipelines. PDUE is the computed provisioning quantity.

5. CALL IOL (EDS,CKME,QME,QFME,PDME)
CALL IOL (ODS,CKMO.QMO,QFMO,PDMO)
CALL IOL (DIS,CKMI,QMI,QFMI,PDMI)
CALL IOL (DDS,CKMD,QMD,QFMD,PDMD)
These statements compute initial module provisions at the Equipment, Direct Support, General Support, and Depot levels, respectively. The arguments in these calls are similar to the one's above for LRUs.

6. CALL IOL (ODS,CKPO,QPO,0.,PDPO)
CALL IOL (DIS,CKPI,QPI,0.,PDPI)
CALL IOL (DDS,CKPD,QPD,0.,PDPD)
These statements will compute initial parts provisions at Direct Support, General Support and Depot, respectively. There is no part stock at the Equipment level. The arguments in these statements are similar to the arguments for LRUs and modules.

7. GO TO 580
This statement ends the SUPPLY RULE logic and transfers around the LOGAM Maintenance Rules.

8. 713 CONTINUE
This statement is the transfer point when maintenance provisioning rules are selected (AYZP≥1).

9. CALL MPROV
This statement calls Subroutine MPROV to compute initial provisions using the LOGAM Maintenance Rules. This statement is executed if the input value for AYZP is greater than or equal to one. MPROV uses the maintenance flows evaluated in Subroutine MFLOW and the time charges input by the user to perform its evaluations. The computed LRU, module and parts stock from MPROV will be distributed in Subroutine STKDS such that there are equal provisions at each stock point of an echelon. The outputs from MPROV are contained in common block /PROVQ/.

10. 580 CONTINUE
A transfer is made to this statement once provisions are either predetermined, computed by supply rules, or evaluated with the maintenance rules.

11. CALL REPLC
This statement calls Subroutine REPLC to replace stock values computed by either the supply rule or the maintenance rule with any stock values that the user inputs. Any stock value input greater than or equal to one will replace the computed value. If the predetermined option (AYZP<0) is selected, all of the preset values (Q=0) and input values become the computed values. The values output from this subroutine are the ones that will be distributed and rounded in Subroutine STKDS unless the fractional part (OR) of AYZP is used to enforce an operational availability. Operational availability is enforced in Subroutine AVAIL by modifying the LRU stock. The modified LRU stock values will become the ones destributed in STKDS.

12. IF (IBG.EQ.1) WRITE (6,BUG)
This statement will print the variables in NAMELIST/BUG/ when debugging of a data case is requested.

13. CALL AVAIL (2,AYZOS,OR,SAVE)
This statement calls the availability routine to compute operational availability (AYZOS). When the desired availability is predetermined (OR), the LRU provisions will be modified to achieve this goal.

14. DO 943 I=1,NA

943 IF(TAYZ(I).EQ.1.) CAYZ(I)=CAYZ(I)*AYZOS**REPEAT
These statements tally operational availabilities for subsystems of LRUs. The system availability is stored in the first location of CAYZ. LRU subsystem availabilities are determined by the location of the TAYZ inputs within the data deck. CAYZI is used to print subsystem availabilities on the case total output page.

15. CALL STKDS
This statement calls Subroutine STKDS to distribute the computed initial provision requirements such that there are an equal number of stock at each supply point of an echelon. The distributed stock is rounded as a function of the "Z" roundoff factors. The default value for these factors (Z) are set in the BLOCK DATA routine as 0.9999.

G. This section of LOGAM redefines scrap rates based on operational availability to give real time rates for all deployed equipments and computes the replenishment spares, maintenance demands and costs for an LRU case.

1. SAVE=ED*OTF*AYZOS
   This statement computes a conversion factor as a function of operational availability (AYZOS). The factors are used to convert scrap rates from "clock time" values per installation to "real time" (OTF) values for total installations (ED).

2. SUT=SAVE*TSU
   SUTD=SAVE*TSLD
   These statements use the conversion factor SAVE to compute "real time" scrap rates for LRUs at all support levels and Depot support, respectively.

3. SMTD=SAVE*TSMD
   SMT=SAVE*(TSME+TSMO+TSMI+TSMD)
   These statements use the conversion factor SAVE to compute "real time" scrap rates for modules at Depot support and total support, respectively.

4. SPTD=SAVE*TSPD
   SPT=SAVE*(TSPO+TSPI+TSPD)
   These statements use the conversion factor SAVE to compute "real time" scrap rates for parts at Depot support and all support levels, respectively.

5. YR8=8766.*YR
   This statement converts the years of operation and maintenance into clock hours (YR8).

6. EDEE=ED*EE
   This statement computes the total number of LRUs installed. ED is the number of installations and EE is the number of LRUs per installation.

7. YPF=8766.*YP-168.*FTU
   IF(YPF.LE.168.*FTU)YPF=168.*FTU
   IF(YPF.LE.0.0)YPF=2000.
   These statements compute the production phase interval during the acquisition phase. YPF is in hours and is used in Subroutine REORDR to compute the maximum production rate for replenishment spares based on the same production rate for stock during the acquisition phase. If a production rate for replenishment spares is input, the production rate determined from YPF will not be used. These statements set at a minimum the factory start-up time (FTU) as the production phase or arbitrarily sets the interval to 2000 hours if the

production phase computed is zero.

8. QUA=EDEE+QT
This statement computes the initial production quantity for LRUs. This includes those installed at all equipment locations (EDEE) and the initial provisions (QT).

9. CALL REORDR (QUA,QT,QU,QTD,SUT,SUTD,PUR,YPF,YR8,QMU, QB,RQN,QC,QCD,RU,RUD,QQC)
This statement calls REORDR to compute lifecycle reorders and scraps for LRUs. The parameters on the first line are input to the routine and the second line are the outputs. A brief description of the variables is given below. The variables occupying the same positions in the argument list for the next two statements are the same as here except they are for modules and parts instead of LRUs.

QUA - Initial LRU production quantity during procurement phase.
QT - Initial LRU provisions at all echelons.
QU - Quantity of LRUs in scrap pipeline.
QTD - Initial LRU provisions at Depot supply points.
SUT - Real time LRU scrap rate for all maintenance echelons.
SUTD- Real time LRU scrap rate at the Depot maintenance level.
PUR - LRU production rate.
YPF - Length of the production cycle in hours.
YR8 - The number of hours in the O&M phase.
QMU - Input value for the LRU minimum reorder lot size.
QB - Computed value for LRU reorder lot size.
RQN - The number of LRU reorder transactions during lifecycle.
QC - The total LRU reorder during lifecycle.
QCD - The LRU reorder for Depot supply points.
RU - LRU residuals at end of life cycle for all echelons.
RUD - LRU residuals at the Depot.
QQC - LRUs consumed at all echelons for the lifecycle.

10. QMA=QUA+QTM
This statement computes the initial production quantity for modules. This includes a Module for each initial LRU buy (QUA) plus the initial stock provisions.

11. CALL REORDR(QMA,QTM,QM,QTMD,SMT,SMTD,PMR,YPF,YR8,QMM, QBM,RQMN,QCM,QCMD,RM,RMD,QQCM)

This statement computes values for reordering parts during program lifecycle. The description of the variable list here are the same as in statement 9 except they are for modules.

12. QPA=QMA+QTP
This statement computes the initial production quantity for parts. This includes one part for each module buy plus the initial provisions.

13. CALL REORDR(QPA,QTP,QP,QTPD,SPT,SPTD,PPR,YPF,YR8,QMP, QBP,RQPN,QCP,QCPD,RP,RPD,QQCP)
This statement computes values for reordering parts during program lifecycle. The description of the variable list here are the same as in statement 9 except they are for parts.

14. CALL TEMWF
This statement calls Subroutine TEMWF to compute the demands for test equipment and maintenance positions to service failures on deployed equipments.

15. CALL LGCST
This statement calls Subroutine LGCST to compute LOGAM costs. Costs for individual LRU cases are evaluated here and accumulated into case totals.

16. IF (IFLAG.GT.0) GO TO 8027
IFLAG is used to skip the summarization of selected LRU cases. IFLAG greater than zero skips the individual LRU case summarys. The individual LRU cases that are summed is controlled by the value of NDLRU. NDLRU is input on the eighth data card.

17. CALL SUMRY(1)
This statement calls Subroutine SUMRY to summarize selected LRU cases. This subroutine is called when the value of IFLAG=0. The value of IFLAG has a default value of zero. Unless the user turns the flag off with a value of one in NAMELIST/L/, this logic will be entered. The method of summarizing LRU cases is the addition of outputs for individual cases 1 and NDLRU+1,2 and NDLRU+2, etc. The outputs from this subroutine are printed using the individual LRU output formats of Subroutine IDVOP. These outputs are printed after the grand totals are printed (NU=-3).

18. 8027 CONTINUE
This statement is the transfer point when the option for requesting summarized output is not selected. When IFLAG=1 is input, the summarization process will be skipped.

19. IF (INHIB.EQ.1) GO TO 70
This statement controls the print of an individual LRU output page. INHIB=1 inhibits the output of individual LRUs by skipping the following statement.

20. CALL IDVOP(1)
This statement calls Subroutine IDVOP to print the outputs for an individual LRU case. This routine is called if INHIB=0. The default value for INHIB is zero and unless the user sets INHIB=1 in NAMELIST/L/, this subroutine will be called.

21. 70 CONTINUE
This statement is the transfer point in the program when individual LRU outputs are not selected (INHIB=1).

22. ILRU=ILRU+ETI*(NLRU-ILRU+1)
ILRU2=ILRU2+ETII*(NLRU2-ILRU2+1)
These statements compute the LRU input case number that starts the next accumulation of maintenance data for Type I and Type II test equipment, respectively.

H. This section of LOGAM reinitializes program variables for the next LRU data case. The work demand accumulators are reset to zero to begin sharing data between LRU cases once work demands are posted out for previous LRU cases. Work demands are accumulated until the posting flags, ETE, ETI and ETII are input equal to one. The next case after these flags are set will begin a new accumulation. The posting flags can be set individually, they do not all have to be turned on at the same time. Some cost factors are set to zero after the first LRU data case is executed. These values will remain zero for the case concept unless the user inputs values for each individual case. The first individual LRU case in additional case concepts will have the original input value restored.

1. SAME=SAME*(1.-ETE)
   This statement resets the accumulator for the manhours to fault isolate remove, test and replace LRUs at one installtion when the work demand flag for Type V test equipment is on (ETE=1.).

2. SETD=((1.-ETI)*(1.-AAIE))+(AAIE*(1.-ETII))
   SADY=SADY*SETD
   SADRY=SADRY*SETD
   These statements reset the accumulators for test and repair manhours at a Depot facility, unless the work demand flags ETI and ETII are off (equal to zero).

3. SAEY=SAEY*(1.-ETI)
   SAERY=SAERY*(1.-ETI)
   These statements reset the accumulators for test and repair manhours at an Organizational maintenance location when the work demand flag for Type I test equipment is on (ETI=1).

4. SAIY=SAIY*(1.-ETI)
   SAIRY=SAIRY*(1.-ETI)
   These statements reset the accumulators for test and repair manhours at a Direct Support location when the work demand flag for Type I test equipment is on (ETI=1.)

5. SAOY=SAOY*(1.-ETI)
   SAORY=SAORY*(1.-ETI)
   These statements reset the accumulators for test and repair manhours at a General Support location when the work demand flag for Type I test equipment is on (ETI=1.)

6. CPUBII=0.
   CPUBV=0.
   CALPUB=0.
   CTCPUB=0.

These statements reset the non-recurring cost factors for technical data of Type II, Type V, Type III, and Type IV test equipment, respectively.

7. EACAL=0.
EACSP=0.
These statements reset the flags that control posting out of one time costs for Type III and Type IV test equipment and manpower.

8. CI=0.
CII=0.
CV=0.
CCAL=0.
CCSP=0.
These statements reset the non-recurring cost factors for developing Type I, Type II, Type V, Type III, and Type IV test equipment, respectively.

9. CEND=0.
CPE=0.
These statements reset the non-recurring cost factors for LRU development and production, respectively.

10. IF(NU.GE.0) GO TO KAD (1,7,8,9)
The processing of an individual LRU case terminates with this statement. If the value of NU is input less than zero in the last LRU case, the next section of the program is entered to output case totals, grand totals and operation and support costs. A value of NU greater than zero will either return to the logic for inputting a new data case (KAD=1 or 9) or to the sensitivity section (KAD=7 or 8) and modify a previous LRU case.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LOGAM)

I. This section of LOGAM contains the logic for accumulating and outputting LOGAM costs, TO&E costs and personnel requirements for case concept totals and grand totals. When the value NU=-1 is input, only the case totals for personnel requirements, LOGAM costs and System Maintenance Support costs are printed. The case total accumulators will not be reinitialized unless the user inputs IS=1 in the NAMELIST inputs. A value of NU=-2 will print the same case totals as above and in addition will print a grand total page for LOGAM costs and System Maintenance Support costs. System Operations and Support costs can be evaluated at this point if the user has input the TOE data (T) and set IOPER=1 in NAMELIST/L/. After printing outputs for NU=-2, the flag IS=1 is set automatically so that the case total accumulators will be reinitialized. A value of NU=-3 will give the same results as NU=-2 except that the grand total accumulators are reinitialized and outputs for summarized LRU cases can be printed; if IFLAG=0. The default value for IFLAG is zero, therefore this option will be executed unless the user sets IFLAG=1 in NAMELIST/L/. In conjunction with IFLAG the user must input a value for NDLRU for the summarization process to work. NDLRU is input on the eighth data card. Refer to section 5 of this document for a description of the NDLRU input.

1. IPRNT=0
   This flag controls the print of case and grand total outputs. With IPRNT=0 case total titles are printed with the output pages. When IPRNT=1 grand total titles are printed and a transfer is made out of the output logic.

2. CALL PERSON
   This statement calls Subroutine PERSON to compute and print the case totals for test and repair personnel. The personnel totals are computed from the maintenance work demands (hours) accumulated in Subroutine TEMWF.

3. WRITE(K004FX,1022)
   WRITE (K004FX,1023)(CAYZ(I),I=1,NA)
   WRITE (K004FX,1024)(CAYZI(I),I=1,NA)
   These statements print operational and inherent availabilities, respectively. The first location of each array contains the system availability while the remaining NA locations contain LRU subsystem availabilities. This print will be found where case totals are output for test and repair personnel.

4. 69 CONTINUE
   This is the return statement for printing grand totals. The output statements are the same for case totals and grand totals. The first pass (IPRNT=0) through will print the case total, if grand totals are requested (NU= -2 or -3) a jump back to this statement is made

with IPRNT=1 to print the grand totals.

5. CALL TTLOP(IPRNT)
This statement calls Subroutine TTLOP to print LOGAM costs for either case totals or grand totals. When NU is less than zero, case totals are printed with the value of IPRNT=0. IPRNT determines if the case total page header is printed or if its the grand total page header. If NU<-1, a transfer is made to statement 69 just prior to this statement to print grand totals. IPRNT=1 for grand totals is set in this section after all case total pages are printed.

6. CALL EIGHT (0,IPRNT)
This statement calls Subroutine EIGHT to print System Maintenance Support cost case totals or grand totals. The first subroutine argument (0) is a flag to print LOGAM Maintenance Support costs rather than Operations Support costs. IPRNT prints either a case total (IPRNT=0) page header or a grand total (IPRNT=1) page header.

7. IF(NU.EQ.-1) GO TO KAD(1,7,8,9)
This statement transfers logic to execute the next individual LRU case if the case total output flag (NU=-1) is on. Statements 1 and 9 for KAD will read a new LRU data case. Statements 7 and 8 enter the logic to modify an existing LRU case for a sensitivity analysis. With NU<-1, Operation and Support costs (TOE) and grand totals are executed.

8. IF(IOPER.NE.1) GO TO 101
This statement will skip the logic to compute System Operation and Support costs using the TOE data unless IOPER=1 is set in NAMELIST/L/.

9. CALL OPER
This statement calls Subroutine OPER to compute Operations and Support costs from the TOE data. This subroutine can only be referenced after a case total has been completed and a grand total accumulation requested (NU=-2 or -3). Unless IOPER=1 Subroutine OPER will not be called under any circumstances. Using a case total as a theater of operation, Support costs for any number of theaters can be evaluated by inputting TOE (T array) data with the last individual LRU case in the case concept.

10. IOPER=0
This statement prevents Subroutine OPER being called until the user resets the flag in NAMELIST/L/.

11. CALL EIGHT(1,IPRNT)
This statement calls Subroutine EIGHT to print System Operation and Support costs. These costs are the sum of the LOGAM System Maintenance Support case totals and the Support costs evaluated in Subroutine OPER.

12. 101 CONTINUE
This statement is the transfer point for bypassing the computation of TOE Support costs.

13. IF(NU.EQ.-2.AND.IPRNT.EQ.1) GO TO KAD(1,7,8,9)
IF(NU.EQ.-3.AND.IPRNT.EQ.1) GO TO 75
These statements control the setting of grand total output arrays and the printing of summarized LRU output cases. With IPRNT=0, the grand total arrays are stored for printing. After grand totals are printed IPRNT=1 is set to prevent entering the grand total logic again. The difference in using either NU=-2 or NU=-3 is that a value of -3 will go to statement 75 and reinitialize the grand total accumulator arrays and print the summarized LRU case outputs.

14. IPRNT=1
This statement sets the print flag that prevents the re-entering of the grand total output logic until a new grand total is requested.

15. IS=1
This statement sets the flag that allows the reinitialization of the case total accumulators. This flag is set only if the value of NU is less than -1.

16. DO 66 I=1,35
66 SUM(I) = SUM(I) + CUM(I)
These statements accumulate case totals for later output as grand totals. The data for each individual LRU case was accumulated into CUM. When a case total (a group of LRU cases) is completed, the case totals are printed and accumulated into SUM with other case totals. SUM (grand total) is printed whenever the value of NU is a -2 or -3. CUM is equivalenced to the first 35 variables in COMMON BLOCK/ZERO/. When NU=-3, the grand total array (SUM) will be reinitialized to zero.

17. DO 584 I=1,16
584 SDAM(I) = SDAM(I) + SDA(I)
These statements accumulate grand totals for maintenance support costs and are output in the same manner as discussed in the previous statement. SDA contains the case totals that are computed in Subroutine LGCST.

18.
```
   DO 68 I=1,35
68 CUM(I)=SUM(I)
```
These statements store the grand total outputs SUM into the case total array CUM for printing. The grand total outputs use the same print statements as the case totals. The grand totals are for the cost variables listed in COMMON/ZERO/.

19.
```
   DO 585 I=1,16
585 SDA(I)=SDAM(I)
```
These statements store the maintenance support grand total outputs SDAM into the case total array SDA for printing. The grand total outputs uses the same print Subroutine (EIGHT) as did the case total output.

20. GO TO 69
This statement transfers logic back through the same code that was used to print case totals, but this time for grand totals.

21. 75 CONTINUE
This statement and the following logic is entered only if grand total outputs have been requested (NU=-3) and printed (IPRNT=1).

22. IF(IFLAG.GT.0) GO TO 8011
This statement is used to bypass the print of summarized LRU case outputs.

23. CALL IDVOP(2)
This statement calls Subroutine IDVOP to print summarized LRU case outputs. The output printed in this subroutine are evaluated in Subroutine SUMRY. IDVOP is the same subroutine that prints the individual LRU case outputs.

24. CALL SUMRY(2)
This statement calls Subroutine SUMRY to reinitialize the summarized LRU case accumulator array. SUMRY is also used to evaluate the summarized data.

25. 8011 CONTINUE
This statement is the transfer point for bypassing the summarized LRU print.

26.
```
   DO 67 I = 1,35
67 SUM(I)=0.
   DO 586 I = 1, 16
586 SDAM(I) = 0.
```
These statements reset the grand total accumulators. Only when NU=-3 will this logic be entered. SUM is the

accumulator array for the grand totals printed in Subroutine TTLOP while SDAM is the accumulator array for grand total maintenance support costs printed in Subroutine EIGHT.

27. GO TO KAD(1,7,8,9)
This statement transfers program logic to the appropriate section of code that is dictated by the value assigned to KAD.

28. 19 WRITE (K004FX,21) SENSY
21 FORMAT (1X,9HBAD SENSY/34(8E14.7/))
GO TO 9
91 WRITE (K004FX,92)
92 FORMAT (1X,18HBAD SENSY SEQUENCE)
GO TO 9
These statements print diagnostics when errors are encountered in sensitivity input data. After each diagnostic, a logic transfer is made to statement 9 to reinitialize accumulators and to read another data case. The transfer to either statement 19 or statement 91 is from the sensitivity section.

29. 3 STOP
END
These statements terminate program execution. The only way to get to this logic is by reading on end-of-file on the input data file. An end-of-file occurs when all input data has been read. The normal way to terminate execution of the program is to input NU=-4 in the last data case.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (FUNCTIONS)

4.2 LOGAM Function Routines

There are three function routines that are described in this section. A brief description of each is the following:

1. BLKDAT — Block data routine that initializes the input variables and some internally used variables. This routine initializes the variables only once, this is at load time of the program.

2. AB — Function used to return a numeric zero when the input argument is less than or equal to zero; otherwise, a numeric one is returned.

3. D — Function used to return a numeric one if the input argument is less than or equal to zero. Used in several statements of the program to prevent a division by zero.

4.2.1 BLOCK DATA - BLKDAT

This routine initializes the variables of COMMON/INPUT/ and other variables that require their values to be preset. A BLOCK DATA routine causes the referenced variables to assume the assigned values before the first executable statement is entered. This routine can not be called from another routine and the values are initialized only once per program start. Any values input by NAMELIST/L/ will override the values set by this routine.

A. The following section of code contains the assignment of program variables to computer memory. For those variables initialized in COMMON/INPUT/ a group of "SAV" arrays are equivalenced to groups of variable names as a means of breaking up the initialization (DATA statements) into logical groups.

1. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5)
   REMARK(18),COSTIS(9),IPAGE
   This statement contains the list of information printed as a header for a page of output. With the exception of IPAGE all other information is input on the first 8 data records (cards). IPAGE is initialized in this routine.

2. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This statement contains all of the tape and/or disc files used by LOGAM.

3. COMMON/LRU/LRU,NRU,ILRU,NLRU,ILRU2,NLRU2
   This statement contains the LRU case counters. "LRU" counts the individual LRU cases input and "NRU" counts the cases for a sensitivity run. NRU is the only one of these variables that is initialized in this routine. ILRU and NLRU are the initial and last case number for a group of LRUs whose Type I test equipment is posted. ILRU2 and NLRU2 are similar to ILRU and NLRU except they are for Type II test equipment.

4. COMMON/SUMLRU/TLRU(15),NDLRU,ICN,ISET,IATE,IA7,IA8
   This statement contains the variables used for summarizing and printing LRU case totals. TLRU and NDLRU are a 60 character description of the summarization and the number of LRU cases to be summarized, respectively. These 2 variables are input on the 8th input record read by LOGAM but the remaining variables of this list are initialized in this routine.

5. COMMON/POUT/POUT(13),AMULT
   This statement contains the output array for accumulating System Support costs from LOGAM and the TOE. These outputs are printed in the DA PAM 11-4

format by Subroutine EIGHT. AMULT is the cost unit conversion factor.

6. COMMON/OFF/O(6)
COMMON/ENLM/ENLM(7)
These statements contain the arrays for accumulating personnel and pay and allowances for officers and enlisted men. These accumulations are in the TOE evaluation section (Subroutine OPER) of the program.

7. COMMON/INPUT/ARA,ARAD,AYZP,CAD,CALMAN,CALPUB,CALSET
CCAL,CCALP,CCALR,CCSP,CCSPP,CCSPR,CCDDI,CDEO,
CDFD,CDID,CDIO,CDIST,CDMAN,CDOE,CDOI,CDPMAN,
CDPRMN,CDRMAN,CEMAN,CEN,CEND,CERMAN,
CFTD,CGMAN,CGRMAN,CI,CII,CKIT,CKMD,CKME,CKMI,
CKMO,CKPD,CKPI,CKPO,CKUD,CKUE,CKUI,CKUO,CLRUPG,
CMODPG,CMP,CONMAN,CONTCT,CPE,CPI,CPII,CPP,
CPUBII,CPUBV,CPV,CRI,CRII,CRM,CRP,CRU,CRV,
CSDEP,CSDSU,CSESU,CSGSU,CTCPUB,CTRA,CTRAD,
CTRCAL,CTRI,CTRII,CTRSPT,CTRV,CUBEM,CUBEP,CUBEU,
CUCE,CUP,CV,DAOQL,DD,DDS,DI,DIS,DTE,DTI,DTO,
E,EACAL,EACSP,ED,EDS,EE,EREI,ETE,ETEI,ETI,
ETII,EVDM,EVDR,EVDT,EVEM,EVER,EVET,EVIM,
EVIR,EVIT,EVOM,EVOR,EVOT,FE,FI,FII,FINT,
FLM,FMD,FMI,FMO,FN,FNGF,FNSP,FSA,FTI,
FTII,FTM,FTP,FTU,FUD,FUE,FUI,FUO,G,H,HPM,
HPP,HPU,IBG,IFLAG,IMF,INHIB,IO,IOPER,
IS,JTED,NA,NB,NU,OD,ODS,OL,OST,OTF,P,PMR,
PP,PPR,PUR,QMM,QMP,QMU,QTD,QTE,QTI,QTMD,QTME,
QTMI,QTMO,QTO,QTPD,QTPI,QTPO,RDD,REO,REPEAT,
RF,RID,ROI,SL,SMD,SME,SMF,SMI,SMO,SPE,SPEV,
SPEVR,STAT,SUD,SUE,SUI,SUO,SVE,SVR,SVT,SVV,
TALMAN,TAT,TATE,TAYZ,TC,TD,TDI,TDMAN,TDMW,TDPMI,
TDPMII,TDPRI,TDPRII,TDR,TDRMAN,TE,TEMAN,TENMAN,
TEO,TER,TERMAN,TF,TFR,TGMAN,TGRMAN,TI,TID,TIMW,
TIO,TIR,TMD,TMDD,TMDR,TMI,TMID,TMIR,TMO,TMOD,
TMOR,TOE,TOI,TOMW,TONMAN,TRC,TUMD,TUMI,TUMO,WD,
WDM,WDR,WE,WEM,WER,WI,WIM,WIR,WM,WMR,WMT,WO,WOM,
WOR,WP,WTKIT,WU,YAT,YD,YMWO,YP,YR,YZ,ZFL,ZI,ZM,
ZO,ZP,ZU,SENSY,T
This statement contains the list of all variables that can be input through NAMELIST/L/. To initialize these variables, "SAV" arrays are equivalenced to certain groups of variables in this list. The "SAV" arrays are described in the following statements.

8. DIMENSION SAVA(134),SAVG(27),ISAV(11),SAVO(63)
SAVT(53),SAVW(38)
This statement dimensions "SAV" arrays to correspond to groups of variable positions in COMMON/INPUT/. The values for each of these groups will be initialized in a DATA statement by equivalencing the arrays to the

COMMON/INPUT/ list.

9. EQUIVALENCE(SAVA,ARA),(SAVG,G),(ISAV,IBG)
EQUIVALENCE(SAVO,OD),(SAVT,TAYZ),(SAVW,WD)
These statements equivalence the "SAV" arrays to elements of the COMMON/INPUT/ list. SAVA shares the same computer memory addresses as the first 134 locations of COMMON/INPUT/. The variables affected when storing data into SAVA are ARA through FUO. The next variable in the /INPUT/ list is "G" which is the beginning location for SAVG. SAVG and the remaining arrays share storage with /INPUT/ similarily to SAVA including SAVW which includes variables through ZU. The last 2 elements of/INPUT/ (SENSY and T) are initialized under their COMMON/INPUT/ name.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (BLOCK DATA)

B. This section of BLOCK DATA initializes the input and program variables using FORTRAN DATA statements.

1. DATA NRU/0/
   This statement initializes the sensitivity case LRU counter to zero.

2. DATA ICN,ISET,IATE,IA7,IA8/5*0/
   This statement initializes the summarized LRU case counters and flags to zero. These variables are used in Subroutine SUMRY.

3. DATA O/6*0.1
   DATA ENLM/7*0./
   This statement initializes the arrays that accumulate TOE personnel and Pay and Allowances for officers and enlisted personnel.

4. DATA POUT/13*0./
   This statement initializes the array that accumulates System Support costs for LOGAM and the TOE. These outputs are printed in the DA PAM 11-4 format by Subroutine EIGHT.

5. DATA AMULT/1.0/
   This statement initializes the cost conversion factor.

6. DATA IPAGE/0/
   This statement initializes the output page counter.

7. DATA K001FX,K002FX,K003FX,K004FX,ND,K8/17,18,5,6,1,8/
   This statement initializes the working files (K001FX,K002FX) for storing summarized LRU cases to 17 and 18. The input file (K003FX) and the print file (K004FX) are set to 5 and 6, respectively. The working file (ND) for storing individual LRU cases for a later recall is set to 1. K8 which is initialized as 8 is the AR570-2 MACRIT data file.

8. DATA SAVA/2*0.,1.,79*0.,4*1.,7*0.,20*1.,
   5*0.,3*1.,2*0.,1.,6*0.,4*1./
   This statement initializes 134 values of COMMON/INPUT/ beginning with variable ARA and ending with FUO.

9. DATA SAVG/21*0.,3*1.,3*0./
   This statement initializes 27 values of COMMON/INPUT/ beginning with variable G and ending with HPU.

10. DATA ISAV/0,1,5*0,2*1,2*0/
    This statement initializes the interger variables of COMMON/INPUT/ beginning with IBG and ending with NU.

11. DATA SAVO/10*0.,2*1.,0.,1.,2*0.,3*1.,13*0.,1.,.9,
12*0.,2*1.,15*0./
This statement initializes 63 values of COMMON/INPUT/ beginning with variable OD and ending with TATE.

12. DATA SAVT/10*1.,43*0./
This statement initializes 53 values of COMMON/INPUT/ beginning with variable TAYZ and ending with TUMO.

13. DATA SAVW/9*168.,0.,2*48.,3*168.,6*0.,1.,10.,0.,
.99999,0.,4*.99999,0.,7*.99999/
This statement initializes 38 values of COMMON/INPUT/ beginning with variable WD and ending with ZU.

14. DATA SENSY/266*0./
This statement sets all locations of the sensitivity array to zero. SENSY is next to the last variable in COMMON/INPUT/.

15. DATA T/2000*0./
This statement initializes the last variable in the COMMON/INPUT/ list. T is the TOE input array for computing System Operation and Support costs.

16. END
This statement terminates the initialization process.

4.2.2 FUNCTION AB

This function is referenced from Subroutines AVAIL, LGCST and ONE. Its only purpose is to return a value that is either a zero or a one. The input argument X of Function AB is checked for a value that is less than or equal to zero. If X has such a value then a value of zero is returned for AB; otherwise AB is equal to one. The function AB is used in many statements in the LOGAM program as a multiplier where only positive non-zero values contribute to the value of the statement.

The code for this function is very simple and will not be paraphrased here but the code is included below.

```
FUNCTION AB(X)
AB=1.
IF(X.LE.0.)AB=0
RETURN
END
```

4.2.3 FUNCTION D

This function is referenced from several LOGAM Subroutines. It's purpose is to return a positive non zero divisor to the referencing statement. When the input argument(X) is positive and non zero, the value returned as function D is the value of X. A zero or negative value of X will result in a value of D=1.0 being returned. This function prevents a division by zero in the referencing statements.

The code for this function is rather simple and will not be paraphrased here but the code is included below.

```
FUNCTION D(X)
D=X
IF(X.GT.0)RETURN
D=1.0
WRITE(6,10)X
10 FORMAT(1X,34H****FUNCTION D CALLED WITH VALUE=,F10.2//)
RETURN
END
```

SECTION 4 - DETAILED PROGRAM DESCRIPTION (AVAILABILITY)

4.3 LOGAM Availability Routines

The following three subroutines that are described in this section are used to compute inherent and operational availabilities.

1. AVAIL Computes the availabilities and under certain conditions can modify the initial provisions.

2. DEF Called from AVAIL to compute back order quantities using a poisson distribution.

3. SPOL Called from AVAIL to modify availability when redundant (built-ins spares) equipments are used.

4.3.1 SUBROUTINE AVAIL

This subroutines computes both inherent and operational availability. AVAIL is called by the main program (LOGAM) with the value IA=1 to compute inherent availability and IA=2 to compute operational availability. Inherent availability is computed as a function of the quantity of material systems and the supply quantity required to satisfy the time for replacing a LRU. Supply quantities are modified by the backorder quantities to compute operational availability. Scheduled maintenance downtime is also considered when computing operational availability where it is not for inherent availability. Availability is modified in this subroutine to account for built-in spares (FN).

The user has the capability to drive operational availability to a pre-determined value. The fractional portion (OR) of the input for AYZP is used as the operational availability to strive for. Any value less than .5 will not be considered. Supply quantities are either added to or substracted from the quantities of the previous pass through the operational availability evaluation until either the availability is met, all supplies are exhausted or the addition of supplies is having a minimal effect on availability.

AVAIL uses a power of two function for doubling or halving the quantities until the iteration for the desired availability is completed. This method is much faster than the previous method of adding one unit at a time. It also allows the user to reduce the availability by subtracting supplys. The criteria adopted here for which supply level to add or subtract supplies is the following. When supplies are added in order to increase availability, the supplies are always added to the organizational level. When supplies are subtracted, the units at the Depot are the first exhausted and then to GS, DS, and organization if necessary.

A. The first section of code includes the subroutine name and the assignment of variables in computer memory.

1. SUBROUTINE AVAIL (IA,AYZ,OR,SAVE)
   This is the entry point for this subroutine. There are two references from the main routine (LOGAM) to AVAIL. The first is with the value IA=1 for inherent availability and the second is with IA=2 for operational availability. AYZ is the availability value returned to LOGAM as either AYZIS (inherent) or AYZOS (operational). OR is the fractional part of AYZP that represents the desired operational availability. Anything less than 0.5 is not considered. SAVE is used to convert backorders to real time operation quantities

and is used only in the evaluation of operational availability.

2. COMMON/INPUT/
COMMON/PROVQ
COMMON/BAS/
These common blocks are described in LOGAM.

B. The next section of code is the initialization for AVAIL.

1. DELU=0.
This statement resets the quantity of units that will be either added or subtracted from the supply quantities when a predetermined operational availability is desired.

2. OTFN=OTF
Sets the operating time factor to the input value for computation of inherent availability. OTFN is recomputed for use in the operational availability evaluation.

3. OR=0.
Resets the factor for a predetermined operational availability. OR is the fractional part of AYZP.

4. PQTE=PDUE
PQTO=PDUO
PQTI=PDUI
PQTD=PDUD
PQT=PQTU
These statements initialize the quantity of units required to meet the desired operational availability to the supply quantities computed as function of the maintenance policies. The quantities are for Organization, Direct Support, General Support, Depot, and Total.

C. This section of AVAIL begins the preliminary computations for evaluating inherent and operational availability. Pipeline supply quantities are summed here for later use in computing back order quantities and availability.

1. IF(IA.EQ.2) GO TO 10
This statement causes a transfer to statement 10 when operational availability is requested from the main program (LOGAM).

2. QYU=0.
```
QYZ=QYE*H(1)+(QYO*H(2)*(1.-H(1)))+(QYI*H(3)
    *(1.-H(1))*(1.-H(2)))+(QYD*H(4)*(1.-H(1)
```

```
    *(1.-H(2)*(1.-H(3)))+(1.-H(1))*(1.-H)(2))
    *(1-H(3))*(1.-H(4)*QYF
GO TO 40
```

These statements begin the initialization for inherent availability. Since this subroutine uses the same statement to compute inherent and operational availability the backorder quantity, QYU, is set to zero for inherent availiability. QYZ is used for both inherent and operational availiability computation; where QYZ is the pipeline LRUs to the nearest authorized supply location (H). For example, if neither Equipment, DS, GS, nor Depot are authorized supply points, then the quantity is QYF. QYF is the quantity determined in BASIC by including the Depot Factory pipe.

3.
```
10 OR=AMOD(ABS(AYZP),1.)
```

This statement begins the operational availability evaluation. OR is the fractional part of AYZP that the user inputs for the desired operational availability. OR acts as an override factor to drive the LRU initial stock to a level where the desired availability is met.

4.
```
RQU=(QUE+QFE)*H(1)+(QUO+QFO)*AB(H(1)+H(2))+
      (QUI+QFI)*AB(H(1)+H(2)+H(3))+
      QUD*AB(H(1)+H(2)+H(3)+H(4))
RQM=QMO+QFMO+QMI+QFMI+QMD+QFMD+QME+QFME
RQP=QPO+QPI+QPD
```

These statements compute the expected value demand for supply by summing pipeline scrap and repair float quantities for LRUs, modules and parts, respectively.

5.
```
OTFN=OTF*(1+SMF)
```

The fraction of real time for equipment in operation (OTF) is modified by scheduled maintenance fraction (SMF).

6.
```
QYMP=DEF(RQM+DEF(RQP,PQTP,1.),PQTM,1.)
```

This statement computes the Back Order Quantity (BOQ) for unfilled items from the warehouse. The BOQs are computed by function DEF which is based on a poisson distribution of demand per resupply interval. The statement is a nest of two calls to DEF. The first call calculates (DEF(RQP,PQTP,1.)) how the shortage of parts will affect the demand for modules. The shortage of modules increases the demand for LRUs on the next call (DEF(RQM+,PQTM,1.)). QYMP is the shortage of modules and will be used later to determine the increase in demand for LRUs because of the module shortage.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (AVAIL)

D. This section of AVAIL initializes factors and flags that are used to iterate on a desired operational availability (OR). These factors are used when the fractional part of AYZP is greater then 0.5. An OR $\leq$ .5 assumes that the desired availability is the computed availability (AYZ).

1. N=0
   This sets the exponent for the power of two algorithm used to compute the increment (decrement) to the LRU initial provisions. The first pass (N=0) will modify the LRU provisions (PQT) by one unit.

2. ISKP=0
   This flag is used to skip the resetting of the sign factors (SGN,SGNN) after the first pass of the operational availability iterations. The sign factors are either both negative or positive on the first iteration but SGNN can change on subsequent iterations.

3. ISGN=0
   This statement initializes the flag used to decrement the power of two exponent (N) when the computed availability either becomes less or greater than the desired availability. The numeric sign of SGN and SGNN must differ before ISGN becomes set.

4. SGN=1
   SGNN=1
   These statements initialize the numeric signs to increment the provisioning quantities (PQT) until computed availability (AYZ) exceeds desired availability (OR). These variables will both become negative if on the first iteration OR is greater than AYZ where PQT must be decremented.

E. This section of AVAIL computes the availabilities. Both inherent amd operational availability are computed here. When inherent availability is evaluated a transfer is made to statement 40 after QYU is initialized to zero and OTFN is set to the value of OTF. The effect of built-in spares (FN) on availability is also evaluated in this section.

1. 30 QYU=DEF(RQU+QYMP,PQT,1.)/SAVE
   This statement is used only in the evaluation of operational availability. A poisson distribution of demand per supply cycle is used to determine how the shortage of modules (QYMP) increases the demand for LRUs. QYMP was evaluated in statement C-8 above. PQT is the LRU provisioning quantity and RQU is the expected demand for pipeline supply of LRUs. PQT was evaluated in statement C-4. Dividing the results by

SAVE converts to the shortage of LRUs at a single equipment location.

2. 40 AYZ=EE/(EE+OTFN*(QYZ+QYU))
This statement computes availability at a single organizational site based on the quantity of installed equipments (EE) and the demand for supplies (QYZ,QYU). When this statement is used in the inherent availability computation QYU=0 and the operational up time (OTFN) is the input value for OTF.

3. AYZ=SPOL(AYZ,FN,EE)
This statement modifies the operational availability in subroutine SPOL to account for built-in spares.

F. The next section of AVAIL is used to iterate to a desired operational availability (OR). The LRU provisioning quantities (PQT) are modified until either the availability is reached, or PQT becomes zero, or addition of LRUs have a minimal effect on the back order quantity (QYU). The increment (decrement) to PQT is doubled on each iteration and added to the quantity computed on the previous iteration. When the addition (subtraction) of LRUs drives AYZ past the desired availabilty, the delta function (DELU) will be halved from the previous iteration and subtracted (added) from PQT. This procedure continues until the DELU value become equal to one unit (availability is reached) or until one of the other conditions mentioned above is met. SGN and SGNN are the controls that either increment or decrement the provisioning quantity.

The logic contained here allows for the removal of LRU provisions at all supply levels beginning with Depot storage first and ending at the organizational level. When stock is added it is included with the organizational level supply only.

1. IF (OR.LE.0.5) GO TO 130
This statement transfers logic out of AVAIL if the desired operational availability is not greater than fifty percent.

2. IF (ABS(AYZ-OR).LE.0.001) GO TO 100
This statement terminates the iteration for OR if the operational availability computed is within one tenth of a percent of the desired value.

3. IF (PQT .LE. 0.0 .AND. AYZ .GT. OR) GO TO 100
This statement terminates the iteration for OR if the computed availability exceeds the desired availability and the LRU supply is exhausted.

4. IF (ISKP .EQ. 0) GO TO 45

This statement allows for the resetting of the sign functions (SGN,SGNN) on the first iteration. The sign functions are initialized as positive but are reset negative if the computed availability (AYZ) is greater than desired availability (OR).

5. IF (AYZ .LT. OR .AND. QYU .LE. 1.E-10) GO TO 100
This statement terminates the iteration if the addition of LRUs is having a minimal effect on the back order quantities. This statement prevents the addition of LRU stock without an appreciable increase in operational availability.

6.
```
   IF (ISKP .EQ. 1) GO TO 50
45 ISKP=1
   IF (AYZ .LE. OR) GO TO 90
   SGN=-1
   SGNN=-1
   GO TO 90
```
These statements control the resetting of the sign functions (SGN, SGNN). These functions as a pair can only be set on the first iteration. They are set positive initially but can be reset negative if provisioning quantities (PQT) are to be decremented, i.e., AYZ is greater than OR.

7.
```
50 SGNN=1
   IF (AYZ.GT.OR) SGNN=-1
```
These statements are entered only after the first iteration has been completed. They set the sign for the delta LRU (DELU) computation. If computed availability (AYZ) is greater than desired availability (OR) then LRUs must be subtracted from the supply (SGNN=-1), otherwise SGNN is positive.

8. IF (SGN.NE.SGNN) ISGN=1
This statement sets the flag ISGN when the direction of change (+,-) is different from the initial direction (first iteration). SGN is set initially but SGNN can change on subsequent passes depending on whether LRUs are to be added or subtracted in order to iterate on OR. When SGN and SGNN first become different, the logic to halve the DELU value instead of doubling begins.

9.
```
70 IF(ISGN.NE.1) GO TO 80
   N=N-1
   IF(N.LT.0) GO tO 100
   GO TO 90
```
These statements set the power of two exponent for the halving process when the signs of SGN and SGNN become different (ISGN=1). The closest iteration to the

desired availability that is allowed in the subroutine is when the absolute value of DELU becomes one. Since 2 to the Nth power equals 1 when N=0, the iteration process is terminated if N becomes negative.

10. 80 N=N+1
This statement computes the power of two exponent for doubling the quantities to be added (subtracted) to the provisioning quantities. This statement will be executed as long as the iteration process is in the same direction as the initial iteration (SGN=SGNN). For example, on the first three iterations AYZ is less than OR but on the fourth iteration AYZ becomes greater than OR. In this case LRUs must be removed from the previous (3rd iteration) PQT to iterate on OR. The logic described in section F-9 above would then be executed in order to halve the DELU from the third iteration where the sign of SGNN determines if the DELU is added or subtracted from the previous PQT.

11. 90 DELU=SGNN*(2.0**N)
PQT=PQT+DELU
These statements compute the LRU supplies for iterating to a desired operational availability (OR). The DELU algorithm will double the supply increments until the availability is reached or exceeded (either in a negative or positive direction). If OR is exceeded then the halving logic along with the sign function (SGNN) allows iterating to OR within one unit (DELU=$\pm$1). The PQT from the previous iteration is modified by the value of the delta unit function. PQT is set initially to the computed value for initial LRU provisions at all supply levels.

12. IF (PQT.GT.0.0) GO TO 30
PQT=0.0
DELU=0.0
GO TO 30
These statements transfer logic to statement 30 where back order quantities are computed just prior to computing operational availability. If the supply quantity (PQT) becomes less than one because of the DELU value, PQT and DELU are set to zero for one last pass through the availability computations.

13. 100 DELU=PQT-PQTU
This statement is executed after the last iteration is made through the availability computations. DELU here is the total LRUs added or subtracted from the initial provisions (PQTU) to meet the desired availability.

14. IF (DELU.LT.0.0) GO TO 110

```
PQTE=PDUE+DELU
GO TO 120
```

These statements will add any additional LRUs (DELU) to the initial provisions at the organizational level.

15.
```
110 PQTD=PDUD+DELU
    IF (PQTD.LT.0.0) PQTI=PDUI+PQTD
    IF (PQTI.LT.0.0) PQTO=PDUO+PQTI
    IF (PQTO.LT.0.0) PQTE=PDUE+PQTO
```

These statements compute the LRU provisions required at a supply level, by beginning at the Depot first, for the condition where LRUs must be removed to meet the requested operational availability (OR). These statements are executed only if DELU is negative. If there are more LRUs to be removed from a supply level than there are provisions, the additional removals will come from the next level down to the equipment level if necessary.

16.
```
120 CONTINUE
    IF (PQTE.LT.0.0) PQTE=0.0
    IF (PQTO.LT.0.0) PQTO=0.0
    IF (PQTI.LT.0.0) PQTI=0.0
    IF (PQTD.LT.0.0) PQTD=0.0
```

These statements set the LRU supplies to zero at any level where there was more stock to be removed than was provisioned for.

17.
```
PQT=PQTE+PQTO+PQTI+PQTD
```

This statement sums the LRU provisions from each supply level for the total LRUs required to meet the requested operational availability.

18.
```
RETURN
END
```

These statements terminate execution of subroutine AVAIL and returns control to the calling program, LOGAM.

4.3.2 FUNCTION DEF

This function is referenced from AVAIL to compute a Back Order Quantity (BOQ) for unfilled orders of items at the warehouse. The unfilled orders result from an inadequate supply for meeting the demand experienced during any given interval between resupplies. This function is based on a Poisson distribution of demand per resupply interval. The function is called with the expected value stock demand and the stock on hand; whereby, DEF will compute the average number of unfilled orders.

A. The first statement is the function name and it's arguments.

1. FUNCTION DEF(QUF,QT,XD)
   The BOQ value computed by this function is returned to the referencing statement as DEF. The arguments are described as:

   QUF - The expected value demand (quantity) per resupply interval.
   QT - The quantity of stock on hand.
   XD - A multiplication and division factor. In this program XD is always used as 1.0.

B. The next section of code initializes values and checks the input quantities to determine if its necessary to proceed through the Poisson distribution.

1. DEF=0.
   IF(QUF.LE.0.)RETURN
   These statements will return a value of zero for the BOQs if there is not a demand for stock.

2. N=(QT/XD)+.1
   This statement prevents a round off error when converting the quantity of stock on hand to on integer value. XD is equal to 1.0.

3. DEF=QUF
   IF(N.EQ.0)RETURN
   These statements will return a BOQ value equal to the stock demands when there is no stock on hand.

4. X=QUF/XD
   RK=1.
   These statements initialize the expected stock demand and a counter for the number of times the expected stock demand will be halved. If the stock demand is greater than 26, both the stock demand and the stock on

hand are halved until the value for stock demand is not greater than 26. Each time the stock is halved, RK will be doubled. This logic occurs later in this routine. The reasoning is discussed in the next section.

C. The next section of code presets values that will be used in the distribution. A computation is made here to determine if the stocked values are sufficient; if so, a return is made with the value zero for BOQs.

1. 5 IF(X.GT.26.)GO TO 3
   This statement will not permit entering the distribution code until the adjusted stock demand (X) is 26 or less. The logic at statement 3 (line item E-1) will halve both the stock demand and stock on hand until X is 26 or less. The reasoning for this is that the distribution T=EXP(-X) would be insignificant for greater values of X.

2. DEF=0.
   IF(N.GT.(2+INT(X+(6.*SQRT(X))))RETURN
   These statements will return a BOQ value of zero if the adjusted stock quantity is greater than the adjusted distributed value for stock demand.

3. M=N-1
   T=EXP(-X)
   RN=N
   Y=X-RN+(T*RN)
   I=0
   These statements initialize the values that will be used in the next section of code where the BOQ will be computed using the Poisson distribution.

D. The next section of code uses a Poisson distribution to compute the BOQs.

1. 6 I=I+1
     RI=I
     T=T*(X/RI)
     Y=Y+(T*(RN-RI))
     IF(I.LT.M)GO TO 6
   These statements will compute the BOQ (Y) for a segment of the stock on hand quantities (RN). RN is the floating point value for stock on hand (N) where N was computed in B-2 above. The value for Y on entering this logic was for the Nth quantity. This logic will

loop until the contribution for each decremented value of N is computed and summed with the previous value of Y.

2. DEF=Y*XD*RK
This statement multiplies the BOQ value (Y) computed above with the number (RK) of times the stock on hand quantity (QT) was subdivided before entering the Poisson distribution logic. XD is input equal to 1.0. DEF is now the BOQ for the stock demands (QUF) input.

3. IF(DEF.LT.0) DEF=0.
RETURN
These statements return to the calling program and DEF will be reset to zero if the computed DEF is a negative value.

E. The following code is used to subdivide the input quantities. A transfer is made here from the statement described in line item C-1 above. The quantities will continue to be segmented (subdivided) until the segmented value of stock demand (X) is less than or equal to 26.

1. 3 X=X/2.
RK=RK+RK
N=N/2
These statements subdivide the expected stock demand (X) and the stock on hand (N). The number of times the values are segmented is accumulated in RK.

2. DEF=QUF
IF(N.EQ.0)RETURN
When the stock on hand segment becomes zero before the expected demand becomes acceptable (26), the BOQ is set to the expected demand that was input (QUF) and a return is made to the calling program.

3. GO TO 5
This statement transfers logic to statement 5 (C-1 above) where another check is made to determine if the stock on hand segment (N) is now less or equal to 26.

4. END
This statement terminates execution of Function DEF.

4.3.3 FUNCTION SPOL

This function is used to modify equipment availability when there are either built in spares or equipment redundancies. SPOL is referenced from subroutine AVAIL after the computation for either inherent or operational availability is made. The availability returned to AVAIL is stored in SPOL.

A. This section of SPOL includes the function name and it's arguments.

1. FUNCTION SPOL(AYZE,FN,EE)
   This statement transmits the input values to the function through the argument list. The value computed in this function and returned to LOGAM will be stored in SPOL. The input arguments are:

   AYZE — Equipment availability (either inherent or operational as computed in AVAIL).

   FN — Data input for the number of identical LRUs in a materiel system.

   EE — Data input for the number of materiel systems at an Equipment installation.

B. The next section of SPOL includes the logic for modifying the input availability.

1. SPOL=1.
   IF(FN.GE.EE)RETURN
   These statements will return an availability factor of 100 percent if the number of identical LRUs is equal or greater than the number of materiel systems.

2. SPOL=AYZE
   IF(EE.EQ.1.)RETURN
   These statements will return the input availability factor when there is only one materiel system at an installation.

3. SPOL=SPOL**EE
   IF(FN.LT..5)RETURN
   This statement will compute a composite availability as a product of each materiel system and return this value if there are not any identical LRUs in the system.

4. T=SPOL

This statement saves the composite availability factor for the FN=1 case. This factor is used later to compute the contribution that each additional FN has on the equipment availability.

5. U=1.-AYZE

This statement computes the compliment of the input availability. This value is used in an equation later that will compute the contribution that the FN term makes to the availability.

6.
```
  I=0
1 I=I+1
  RI=I
  T=(EE-RI)*U*T/(AYZE*RI)
  SPOL=SPOL+T
  IF(RI.GE.FN)RETURN
  GO TO 1
```

These statements will loop through the availability computations once for each FN. When the number of loops (RI) becomes equal to the number of identical LRUs (FN) a return is made to AVAIL with the composite availability SPOL. On each pass through the loop, the contribution (T) that each identical LRU (RI) makes on the input availability is computed and added to the previous computed composite availability (SPOL).

7. END

This statement terminates execution of the function.

4.4 LOGAM Maintenance Flow

The subroutines of this section compute the scrap/repair fractions and the flow of material through the scrap/repair pipelines. The 2 routines described in this section are:

RSFRCT	Computes the scrap, repair and in-limbo fractions.
MFLOW	Computes the pipeline flow of material for scrap and repair using the outputs of RSFRCT and the failure rates.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (RSFRCT)

4.4.1 SUBROUTINE RSFRCT

RSFRCT is called from LOGAM to compute repair and scrap fractions for LRUs and modules. For those units that are neither repaired nor scrapped, an "in-limbo" fraction is computed. The fractions output from this subroutine are used in Subroutine MFLOW to compute the maintenance flows.

A. This section of RSFRCT assign the program variables to computer memory.

1. COMMON/INPUT/
   This common block contains the list of variables input by NAMELIST/L/. The variables from this list that are used in RSFRCT are the LRU scrap fractions SUE,SUO,SUI,SUD; module scrap fracitons SME,SMO,SMI,SMD; LRU repair fractions FUE,FUO,FUI,FUD; and module repair fractions FMO,FMI,FMD. For a complete list of variables in this common block refer to section A of LOGAM.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAT(4),
   TAYZ(10),ZM(4),ZP(3),ZU(4),SENSY(266),
   T(2000)
   This statement contains the diminsioned variables of COMMON/INPUT/.

3. COMMON/FRACT/SUEC,SUOC,SUIC,SUDC,SMEC,SMOC,SMIC,SMDC,
   FUEC,FUOC,FUIC,FUDC,FMOC,FMIC,FMDC,
   USE,URE,ULE,USO,URO,ULO,USI,URI,USD,URD,
   MSO,MRO,MSI,MRI,MSD,MRD
   REAL MSO,MRO,MSI,MRI,MSD,MRD
   These statements include the repair, scrap and "in-limbo" fractions that are output from RSFRCT. The "REAL" statement assigns the variables with integer names to floating point names.

B. This section of RSFRCT uses the input fractions to compute the fraction of LRUs and modules that are either repaired, scrapped or are "in-limbo" at each maintenance echelon.

1. SUEC=1.-SUE
   SUOC=1.-SUO
   SUIC=1.-SUI
   SUDC=1.-SUD
   These statements compute the fraction of LRUs not scrapped at the Equipment, Direct Support, General Support and Depot levels, respectively.

2. SMEC=1.-SME

SMOC=1.-SMO
SMIC=1.-SMI
SMDC=1.-SMD
These statements compute the fraction of modules not scrapped at the Equipment, Direct Support, General Support and Depot levels, respectively.

3. FUEC=1.-FUE
FUOC=1.-FUO
FUIC=1.-FUI
FUDC=1.-FUD
These statements compute the fraction of LRUs not repaired at the Equipment, Direct Support, General Support and Depot levels, respectively.

4. FMOC=1.-FMO
FMIC=1.-FMI
FMDC=1.-FMD
These statements compute the fraction of modules not repaired at Direct Support, General Support and Depot levels, respectively. (Modules are not repaired at equipment level).

5. USE=SUE+SUEC*FUEC
URE=SUEC*FUE
ULE=SUEC*FUEC
USO=SUO+SUOC*FUOC
URO=SUOC*FUO
ULO=SUOC*FUOC
USI=SUI+SUIC*FUIC
URI=SUIC*FUI
ULI=SUIC*FUIC
These statements compute the fraction of LRUs scrapped, the fraction of LRUs repaired and the fraction of LRUs in limbo at the Equipment, Direct Support and General Support levels, respectively.

6. USD=SUD+SUDC*FUDC
URD=SUDC*FUD
These statements compute the fraction of LRUs scrapped and repaired at Depot, respectively.

7. MSO=SMO+SMOC*FMOC
MRO=SMOC*FMO
MSI=SMI+SMIC*FMIC
MRI=SMIC*FMI
MSD=SMD+SMDC*FMDC
MRD=SMDC*FMD
These statements compute the fraction of modules scrapped and the fraction of modules repaired at Direct Support, General Support and Depot, respectively.

8. RETURN
END
These statements terminate execution of RSFRCT and returns logic to LOGAM.

4.4.2 SUBROUTINE MFLOW

MFLOW uses failure rate, maintenance policies, and scrap and repair fractions to evaluate the hourly flow of material through the maintenance facilities. Maintenance flows are computed for LRUs, modules and parts at four maintenance echelons; i.e., Organizational, Direct Support, General Support and Depot. MFLOW is called from LOGAM and uses the scrap and repair fractions evaluated in Subroutine RSFRCT.

A. This section of MFLOW contains the assignment of program variables to computer memory. The lists for the common blocks in this section can be found in the main program.

1. COMMON/INPUT/
   This common block includes the entire list of variables input through NAMELIST/L/. Refer to the main program (LOGAM) description for a list of these variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),
   ZU(4),SENSY(266),T(2000)
   This statement is a list of all the dimensioned variables in common block INPUT.

3. EQUIVALENCE(G(1),GA),(G(2),GB),(G(3),GC),(G(4),GD),
   (G(5),GE),(G(6),GF),(G(7),GG),(G(8),GH),
   (G(9),GI),(G(10),GJ),(G(11),GK),(G(12),GL),
   (G(13),GM),(G(14),GN),(G(15),GO),(G(16),GP),
   (G(17),GQ),(G(18),GR),(G(19),GS),(G(20),GT)
   This statement equivalences the maintenance policy array "G" to the variable names that are used in the statements of this subroutine.

4. COMMON/FRACT/
   This common block includes the scrap and repair fractions that were evaluated in Subroutine RSFRCT.

5. REAL MSO,MRO,MSI,MRI,MSD,MRD
   This statement assigns these variables in common block FRACT to real (floating point) variables.

6. COMMON/MNTFL/DSUE1,GSUE1,DUE1,DSUE2,GSUE2,DUE2
   SUE1,SUE2,SUE3,GSUO1,DUO1,GSUO2,DUO2
   SUO1,SUO2,SUO3,SUO4,SUO5,GSMO
   SMO1,DMO,DMI,SMI1,SMI2,SMI3,
   SMD1,SMD2,SUD1,DUI,SUI1,SUI2,
   FGE,FGO,FGI,FGD,TUFTE,TUFTD,TUFTI,
   TUFTD,TUFEC,TUFOC,TUFIC,TUFDC,
   TSME,TSMO,TSMI,TSMD,TMFO,TMFI,TMFD,

TMFEO,TMFOI,TMFID,TSPD,TSPI,TSPO,
TSU,TSLF,TSLD,F,FNG,A,TU,DPT

This common block contains the list of maintenance flows that are evaluated in this subroutine (MFLOW). Each of these variables will be described in the following sections of this subroutine's write-up.

B. The next section of MFLOW computes total equipment failures, false no-gos and attrition at one equipment installation.

1. F=E*EE
   This statement multiplies the failure rate (E) with the number of equipments (EE) at an installation to get total failure rate.

2. FNG=FNGF*F
   This statement multiplies the total failure rate (F) with the false "no-gos" fraction to get the total false "no-go" rate.

3. A=YAT*EE/(8766.*OTF)
   This statement converts the yearly attrition (YAT) rate for one equipment to a total hourly attrition rate based on the operating hours (OTF) of the system.

4. TU=A+F+FNG
   This statement computes total LRU removals per hour by summing attrition, failures, and false removals.

5. DPT=2.-DAOQL
   This statement computes a workload factor for Depot repairs as a result of faulty repairs being returned.

C. This section of MFLOW computes the maintenance flow for LRUs at the organizational level.

1. DSUE1=(F+FNG)*(GB+GD+GF+GL+GM+GN+GO+GP)+
   (F*DPT+FNG)*(GH+GQ)
   GSUE1=(F+FNG)*(GE+GR+GS)
   DUE1 =(F*DPT+FNG)*(GG+GT)
   These statements compute an hourly flow of LRUs from the Organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to evacuate failures and false "no go's" to a higher maintenance level.

2. DSUE2=F*ULE*GI
   GSUE2=F*ULE*GJ
   DUE2 =F*ULE*GK

These statements compute the hourly flow rate for "in limbo" LRUs from the organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to repair at ED, but were not repaired because of the FUE fraction; therefore, resulting in an evacutation to a higher maintenance level.

3. FUTE=F*(GC+GI+GJ+GK)
This statement computes the hourly flow of LRUs through the test facility at the organizational level.

4. SUE1=(F+FNG)*GA
SUE2=F*ULE*GC
SUE3=SUE*FUTE
These statements compute hourly scrap rates for LRUs at the Organizational level. These scrap rates are computed when there is not a repair maintenance policy, when "in limbo" LRUs do not have a higher level of maintenance, and for those LRUs that are scrapped after going through the test facility, respectively.

5. FGE=FNG*(GC+GI+GJ+GK)
This statement computes the hourly flow of false "no go's" which are screened at Organizational level, found serviceable and returned to stock.

6 TUFTE=URE*FUTE
This statement computes the hourly flow of LRUs through the test facility at the Organizational level that are deemed repairable.

7. TUFEC=TUFTE+FGE+SUE3+SUE2+SUE1+DUE2+GSUE2+DSUE2+
DSUE1+GSUE1+DUE1
This statement computes the hourly flow of LRUs that go through the checkout facility at the Organizational level. These are all the flows at ED that will be charged TC.

D. This section of MFLOW computes the maintenance flow for LRUs at the Direct Support level.

1. GSUO1=F*SUOC*(GF+GO+GP)
DUO1=F*SUOC*(GH+GQ)*DPT
These statements compute the hourly flow of LRUs that are evacuated from the Direct Support level to General Support and Depot, respectively. These evacuations are for LRUs with maintenance policies to screen false "no go's" at Direct Support and send the non-scrapped fraction of failures to a higher echelon of

maintenance.

2. GSUO2=F*ULO*GM
DUO2 =F*ULO*GN
These statements compute the hourly flow rate for "in limbo" LRUs from Direct Support to General Support and Depot, respectively. These flow rates are for the units that have a maintenance policy to repair at OD but were not repaired because of the FUO fraction; therefore, resulting in an evacuation to a higher maintenance level.

3. SUO1=F*SUO*(GF+GO+GP)
SUO2=F*SUO*(GH+GQ)*DPT
These statements compute the hourly scrap rate for LRUs at the Direct Support level. The scrap rates here are for those maintenance policies that screen false "no go's" at Direct Support and evacuate LRUs for repair to General Support and Depot, respectively.

4. SUO3=F*GB
This statement computes the hourly scrap rate for LRUs using the maintenance policy to detect false "no go's" at Direct Support and discard all failures.

5. SUO4=F*ULE*ULO*(GD+GL)
This statement computes the hourly scrap rate at Direct Support for that fraction of "in limbo" LRUs evacuated from ED and are still having an "in limbo" status at Direct Support.

6. FUTF=F*(GD+GL+GM+GN+ULE*GI)
This statement computes the hourly flow of LRUs through the test facility at Direct Support.

7. SUO5=SUO*FUTF
This statement computes an hourly scrap rate for LRUs at Direct Support. The scrap rate computed here is for those LRUs that flow through the test facility and are declared not repairable.

8. FGO=FNG*(GB+GD+GF+GH+GL+GM+GN+GO+GP+GQ)
This statement computes the hourly flow of false "no go's" screened at Direct Support, found serviceable, and returned to stock.

9. TUFTO=URO*FUTF
This statement computes the hourly flow of LRUs through the test facility at Direct Support that was deemed repairable.

10. TUFOC=TUFTO+FGO+SUO5+SUO4+SUO3+SUO2+SUO1+DUO2+GSUO2

+DUO1+GSUO1
This statement computes the hourly flow of LRUs through the checkout facility at Direct Support. These are all the flows that will be charged TC at OD.

E. This section of MFLOW computes the maintenance flow for LRUs at the General Support level.

1. DUI=F*ULI*SUOC*GP+F*ULI*GS
This statement computes the hourly flow rate for "in limbo" LRUs from General Support to Depot. This flow is for units that are neither repaired nor scrapped at GS.

2. SUI1=F*ULI*(GE+GR+ULE*GJ+ULO*GM+SUOC*(GO+GF))
This statement computes the hourly scrap rate of LRUs at General Support for "in limbo" units evacuated from ED and DS and are still "in limbo" status at General Support. These LRUs are scrapped because there is not a higher maintenance level for evacuation.

3. FUTI=F*(GE+GR+GS+ULE*GJ+ULO*GM+SUOC*(GO+GF+GP))
This statement computes the hourly flow of LRUs through the test facility at General Support.

4. SUI2=SUI*FUTI
This statement computes the hourly scrap rate for LRUs at General Support. The scrap rate here is for those LRUs that flow through the test facility and are declared not repairable.

5. FGI=FNG*(GE+GR+GS)
This statement computes the hourly flow of false "no go's" screened at General Support, found servicable, and returned to stock.

6. TUFTI=URI*FUTI
This statement computes the hourly flow of LRUs through the test facility at General Support that are deemed repairable.

7. TUFIC=TUFTI+FGI+SUI2+SUI1+DUI
This statement computes the hourly flow of LRUs through the checkout facility at General Support. These are all the flows at DI that will be charged TC.

F. This section of MFLOW computes the maintenance flow for LRUs at the Depot level.

1. FUTD=F*((GG+GT)*DPT+ULE*GK+ULO*GN+ULI*GS+SUOC*

DPT*(GH+GQ)+ULI*SUOC*GP)
This statement computes the hourly flow of LRUs through the test facility at Depot.

2. SUD1=FUTD*USD
This statement computes the hourly scrap rate for LRUs at the Depot. The scrap rate computed here is for those LRUs that flow through the test facility at Depot and are declared not repairable.

3. FGD=FNG*(GG+GT)*DPT
This statement computes the hourly flow of false "no go's" screened at Depot, found servicable, and returned to stock.

4. TUFTD=FUTD*URD
This statement computes the hourly flow of LRUs through the Depot test facility that are deemed repairable.

5. TUFDC=TUFTD+FGD+SUD1
This statement computes the hourly flow of LRUs through the checkout facility at Depot. These are all the flows at DD that will be charged TC.

G. This section of MFLOW computes the maintenance flow for modules at the Organizational level.

1. DSME=F*URE*SMEC*GI
GSME=F*URE*SMEC*GJ
DME =F*URE*SMEC*GK*DPT
These statements compute hourly flow of modules from the Organizational level to Direct Support, General Support, and Depot, respectively. These flow rates are computed when there is a maintenance policy to evacuate failures to a higher maintenance level.

2. SME1=F*SME*URE*(GI+GJ+GK)
This statement computes a scrap rate at the Organizational level for those modules tested at ED and are declared not repairable.

3. SME2=F*URE*GC
This statement computes a scrap rate for modules at the Organizational level when the Maintenance policy is to repair an LRU and discard the module.

4. TSME=SME1+SME2
This statement computes the total hourly scrap rate at the Organizational level.

5. TMFEO=DSME+GSME+DME
This statement computes the total flow of modules evacuated from the Organization level to higher echelons of maintenance.

H. This section of MFLOW computes the maintenance flow for modules at the Direct Support level.

1. GSMO=F*URO*SMOC*GM
DMO =F*URO*SMOC*GN*DPT
These statements compute the hourly module flow from Direct Support to General Support and Depot, respectively, for maintenance policies that evacuate failures to a higher maintenance level.

2. SMO1=F*URO*(MSO*GL+ULE*MSO*GI)+F*URE*SMEC*MSO*GI
SMO2=F*URO*SMO*GM
SMO3=F*URO*GD
SMO4=F*URO*SMO*GN
These statements compute the hourly module scrap rates at Direct Support. The scrap rates computed here are for maintenance policies that repair at DS, evacuate modules for repair at GS, discard modules at DS, and evacuate modules for repair at Depot, respectively.

3. TSMO=SMO1+SMO2+SMO3+SMO4
This statement computes the total module scrap rate for modules at Direct Support.

4. TMFO=F*MRO*(URO*GL+URO*ULE*GI+URE*SMEC*GI)
This statement computes the hourly flow of modules through the repair facility at Direct Support.

5. TMFOI=GSME+DME+GSMO+DMO
This statement adds the hourly flow of modules from Direct Support to higher maintenance echelons to the flow originating at the Organizational level.

I. This section of MFLOW computes the maintenance flow for modules at the General Support level.

1. DMI=F*URI*(SMIC*GS+SUOC*SMIC*GP)*DPT
This statement computes the hourly flow of modules from General Support when the maintenance policies are to repair at the Depot level.

2. SMI1=F*MSI*(URE*SMEC*GJ+URO*SMOC*GM+URI*
(GR+SUOC*GO+ULE*GJ+ULO*GM))
SMI2=F*URI*(GE+SUOC*GF)
SMI3=F*URI*(SMI*GS+SUOC*SMI*GP)

These statements compute the hourly scrap rates for modules at General Support. The rates computed here are for maintenance policies that repair at GS, discard at GS, and evacuate for repair at Depot, respectively.

3. TSMI=SMI1+SMI2+SMI3
This statement computes the total module scrap rate at General Support.

4. TMFI=F*MRI*(URI*(ULO*GM+ULE*GJ+GR+SUOC*GO)+
URE*SMEC*GJ)
This statement computes the hourly flow of modules through the repair facility at General Support.

J. This section of MFLOW computes the maintenance flow for modules at the Depot level.

1. TMFID=DME+DMO+DMI
This statement computes the total module flow from the lower echelons to the repair facility at Depot.

2. FSMD1=F*URD*(DPT*GT+ULE*GK+ULO*GN+ULI*GS+
SUOC*DPT*GQ+ULI*SUOC*GP)
FSMD2=F*(URE*SMEC*GK+URO*SMOC*GN+URI*SMIC*GS+
URI*SUOC*SMIC*GP)*DPT
These statements compute the hourly flow of modules through the test facility at Depot. FSMD1 is for the maintenance policies that repair modules at Depot with LRU repair at Depot for the "in limbo" units from lower echelons. FSMD2 is for maintenance policies where LRUs are repaired at lower echelons and the modules are evacuated to the Depot.

3. SMD1=MSD*(FSMD1+FSMD2)
SMD2=F*URD*(DPT*GG+SUOC*DPT*GH)
These statements compute the hourly scrap rate for modules at Depot. SMD1 is the scrap rate for the modules that go through the test facility. SMD2 includes the scrap for the "discard" maintenance policy and the scrap after screening for false "no go's".

4. TSMD=SMD1+SMD2
This statement computes the total scrap rate for modules at the Depot level.

5. TMFD=MRD*(FSMD1+FSMD2)
This statement computes the hourly flow of modules through the Depot repair facility.

K. This section of MFLOW computes scrap rates at field and Depot

facilities and the demand for parts.

1. TSLF=A+SUE1+SUE2+SUE3+SUO1+SUO2+SUO3+SUO4+
   SUO5+SUI1+SUI2
   This statement sums the LRU hourly scrap rates of all Field level facilities.

2. TSLD=SUD1
   This statement sets the LRU hourly scrap rate at the Depot level.

3. TSU=TSLF+TSLD
   This statement computes the total hourly scrap rate for LRUs.

4. TSPO=TMFO
   TSPI=TMFI
   TSPD=TMFD
   These statements set the parts demand flow at Direct Support, General Support and Depot levels, respectively. Since each repaired module requires a new part, the parts demand and module demand are the same.

5. RETURN
   END
   These statements end Subroutine MFLOW and return control to the main program.

4.5 LOGAM Provisioning Routines

The subroutines described in this section are used to compute initial and replenishment spares. A brief description of the subroutines of this section is:

BASIC	Computes the quantities tied up in the scrap/repair pipelines. Its outputs are used to compute inherent availability and initial spares when using the LOGAM supply rules. BASIC is also called from Subroutine LGCST to compute shipping factors.
IOL	Computes initial spares using the LOGAM "supply rule". IOL modifies the BASIC outputs with a safety stock factor.
MPROV	Computes initial spares using the LOGAM "maintenance rule".
REPLC	This routine is used to replace any computed initial provisions with predetermined provisions. Predetermined provisions are those input through NAMELIST.
STKDS	This routine distributes initial spares such that there is an equal quantity at each supply point of a supply echelon.
REORDR	Computes replenishment spares, consumed material and residuals.

4.5.1 SUBROUTINE BASIC

BASIC computes shipping costs and quantities tied up in pipelines. In general, BASIC uses the rate of flow of LRUs, modules, and parts through the pipes to determine shipping costs and quantities. The flow rates are multiplied by one way shipping costs to find the rate of cost to ship replacements and by two way costs to find the cost to ship repairables. Quantities tied up in the pipelines are computed by multiplying the pipe and delay times with the flow rates. Material flow rates are evaluated in Subroutine MFLOW and passed to BASIC through the COMMON/MNTFL/ parameters.

Shipping costs and quantities tied up in the pipelines are computed using the same statements in BASIC but are made with two separate calls. BASIC is called from LOGAM to compute pipeline quantities and from LGCST to compute shipping factors.

A. The first section of code includes the subroutine name and the arguments passed from LOGAM. The remainder of this section assigns computer memeory locations for the variables used in BASIC.

1. SUBROUTINE BASIC (FIXM,FIXP,FIXU,BHF,BHM,BHP,REQD,
   EALR,DALR,GALR,DEPMOD,GMOD,DMOD,TRCE,
   ASLE,ASLO,ASLI,EFACT)
   This statement is the entry point for the subroutines which includes the list of input parameters. These paramenters along with the maintenance flows will be used to either determine shipping cost rates or quantities tied up in pipe flow. The description for these paramenters can be found in the statements where they are used.

2. COMMON/MNTFL/(list of parameters not included)
   This statement contains the maintenance flows evaluated in Subroutine MFLOW. Refer to MFLOW for a list and description of these parameters.

3. COMMON/BAS/QUE,QUO,QUI,QUD,QUDH,QFE,QFO,QFI,QYE,
   QYO,QYI,QYD,QYF,QME,QMO,QMI,QMD,QMDH,
   QPO,QPI,PQD,QPDH,QFME,QFMO,QFMI,QFMD
   This COMMON statement includes a list of the parameters that are computed in BASIC. When BASIC is called from LOGAM these parameters will represent pipeline quantities. When called from LGCST they represent the shipping cost factors.

4. COMMON /INPUT/ (list of parameters not included)
   This COMMON statement includes the list of variables

input by NAMELIST/L/. Refer to LOGAM for a list of these variables.

5. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
TAT(4),TAYZ(10),ZM(4),ZP(3),
ZU(4),SENSY(266),T(2000)
This statement contains the list of all dimensioned variables of COMMON/INPUT/.

B. This section of BASIC computes either the shipping costs or the demands for repaired modules. The value computed will be for two way (round trip) flow between maintenance levels.

1. QFME=TMFEO*ASLE
This statement computes a value for repairable modules that are evacuated and returned in the Equipment/Direct Support pipeline. TMFEO is the repairable evacuation flow rate from the Organizational level to higher maintenance facilities. TMFEO is the module flow rate (per hour), ASLE is the round trip cost per item (SHTEO) for shipping and round trip pipe length in hours (TEOT) for module demand.

2. QFMO=TMFO*DMOD+TMFOI*ASLO
This statement computes a value for the repairable modules that are evacuated and replaced in the Direct Support/General Support pipeline and also includes the repair flow (TMFO) at Direct Support when computing module demands. TMFOI is the repairable evacutation flow rate from Direct Support to higher maintenance facilities. ASLO is the round trip cost per item (SHTOI) for shipping and the round trip pipe length in hours (TOIT) for module demand. DMOD is zero for shipping cost. For module demands DMOD is the supply allowance in hours (TUMO) at Direct Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service.

3. QFMI=TMFI*GMOD+TMFID*ASLI
This statement computes a value for the repairable modules that are evacuated and replaced in the General Support/Depot pipeline and also includes the repair flow (TMFI) at General Support when computing module demands. TMFID is the repairable evacutation flow rate from General Support to the Depot. ASLI is the round trip cost per item (SHTID) for shipping and round trip pipe length in hours (TIDT) for module demand. GMOD is zero for shipping cost. For module demand GMOD is the supply allowance in hours (TUMI) for modules at General Support to cover the time between removal of a module from a LRU until the time the module is repaired and

returned to service.

4. QFMD=TMFD*DEPMOD
This statement computes only the module demand at the Depot. DEPMOD is zero for the shipping rate costs. TMFD is the rate per hour for modules through the Depot Repair facility. For module demands, DEPMOD is the supply allowance in hours (TUMD) for modules at the Depot to cover the time between removal of a module from an LRU until the module is repaired and returned to service.

C. This section of BASIC computes either shipping costs or mean quantity of demand for scrapped modules. The values computed here are for a one way flow between maintenance facilities.

1. QME=TSME*EALR
This statement computes a value for the flow of modules from Direct Support to Equipment to replace modules scrapped at Equipment. TSME is the modules scrapped per hour per installation. EALR is the cost per item (CDOE) to ship modules one way from Direct Support to Equipment when computing shipping costs. For module demand, EALR is the number of supply hours (REOT) for condemned modules at the Equipment level.

2. QMO=(TSME+TSMO)*DALR
This statement computes a value for the flow of modules from General Support to Direct Support to replace modules scrapped at Direct Support and Equipment. TSME and TSMO are the modules scrapped per hour at Equipment and Direct Support, respectively. For shipping costs, DALR is the cost per item (CDIO) to ship modules from General Support to Direct Support. When computing module demand, DALR is the number of supply hours (ROIT) for condemned modules at Direct Support.

3. QMI=(TSME+TSMO+TSMI)*GALR
This statement computes a value for the flow of modules from Depot to General Support to replace modules scrapped at Equipment, Direct Support, and General Support. TSME, TSMO and TSMI are modules scrapped per hour at the Field facilities. When computing shipping costs, GALR is the cost per item (CDDI) to ship modules from Depot to General Support. For Module demands, GALR is the number of supply hours (RIDT) for condemned modules at General Support.

4. QMD=(TSME+TSMO+TSMI+TSMD)*FIXM
This statement computes a value for the flow of modules from the factory to the Depot to replace modules

scrapped at all maintenance levels. When computing shipping costs, FIXM is the shipping cost (CDFD) per item for modules from factory to Depot. For module demands, FIXM is the reprocurement time in hours (168.*FTM) for modules at the factory.

5. AHPM=BHM-FIXM
This statement computes the discretionary procurement holding time (24.*HPM) in hours for a module. For shipping costs, AHPM is zero.

6. QMDH=(TSME+TSMO+TSMI+TSMD)*AHPM
This statement computes the module demand as a function of discretionary procurement holding time. This term will be zero when computing shipping costs.

D. This section of BASIC computes either the parts demand or the parts shipping cost, as a result of scrap at the Field and Depot maintenance facilities.

1. QPO=TSPO*DALR
This statement computes a value for the flow of parts from General Support to Direct Support as a result of parts scrapped at Direct Support. When computing shipping costs, DALR is the cost (CDIO) per item. For parts demand, DALR is the number of supply hours (ROIT) for condemned modules and parts at Direct Support.

2. QPI=(TSPO+TSPI)*GALR
This statement computes a value for the flow of parts from Depot to General Support as a result of parts scrapped at Direct Support and General Support. When computing shipping costs, GALR is the cost (CDDI) per item. For parts demand, GALR is the number of supply hours (RIDT) for condemned modules and parts at General Support.

3. QPD=(TSPO+TSPI+TSPD)*FIXP
This statement computes a value for the flow of parts from the factory to the Depot to replace parts scrapped at all maintenance levels. When computing shipping costs, FIXP is the shipping cost (CDFD) per item. For parts demand, FIXP is the reprocurement time (168.*FTP) in hours for parts at the factory.

4. AHPP=BHP-FIXP
This statement computes the discretionary procurement holding time (24.*HPP) in hours for parts. For shipping costs, AHPP is zero.

5. QPDH=(TSPO+TSPI+TSPD)*AHPP

This statement computes the parts demand as a function of discretionary procurement holding time for parts scrapped at all maintenance levels.

E. This section of BASIC computes either the shipping cost or the stock demand for replacement of scrapped LRUs. The values computed will be for round trip flow between maintenance levels. TSU in the following statements is the total scrap rate per hour per materiel installation.

1. QUE=TSU*ASLE
   This statement computes a value for the scrap and replacement LRUs in the Equipment/Direct Support pipe. ASLE is the round trip cost (SHTEO) per item for shipping and pipe length time (TEOT) in hours for LRU demands.

2. QUO=TSU*ASLO
   This statement computes a value for the scrap and replacement flow of LRUs in the Direct Support/General Support pipe. ASLO is the round trip cost (SHTOI) per item for shipping and pipe length time (TOIT) in hours for LRU demands.

3. QUI=TSU*ASLI
   This statement computes a value for the scrap and replacement flow of LRUs in the General Support/Depot pipe. ASLI is the round trip cost (SHTID) per item for shipping and pipe length time (TIDT) in hours for LRU demands.

4. QUD=TSU*(FIXU+REQD)
   This statement computes a value for the flow of LRUs from the factory to the Depot as a result of scrapped LRUs. When computing shipping costs, FIXUs the cost (CDFD) per item and REQD=0. For LRU demands, FIXU is the reprocurement time (168.*FTU) in hours for LRUs at the factory and REQD is the delay time (24.*RDD) in hours between the request time and the handling time.

5. QUDH=TSU*(BHF-FIXU)
   This statement computes the LRU demand in the Factory/Depot pipe as a function of discretionary procurement holding time (24.*HPU).

F. This section of BASIC computes either shipping costs or the demand for LRUs as a result of LRUs tied up in the repair float pipelines.

1. QFE=(TUFTO+TUFTI+TUFTD+FGO+FGI+FGD)*ASLE+
   (TUFEC*TC+TUFTE*(TE+TER))*EFACT

This statement computes a value for the evacuation and replacement of repairable LRUs in the Equipment/Direct Support pipeline. When computing LRU demands the flow of LRUs through the checkout, test, and repair (TUFEC, TUFTE) facilities at the Equipment level are included. When computing shipping costs, the TUFEC and TUFTE are excluded by setting EFACT to zero. For shipping costs, ASLE is the round trip cost (SHTEO) per item. For LRU supply demands, ASLE is the round trip pipelength (TEOT) in hours.

2. QFO=(TUFTI+TUFTD+FGI+FGD)*ASLO
   This statement computes a value for LRU repair float in the Direct Support/General Support pipeline. When computing shipping costs, ASLO is the round trip shipping cost (SHTOI) per item. For LRU demands, ASLO is the round trip pipe length (TOIT) in hours.

3. QFI=(TUFTD+FGD)*ASLI
   This statement computes a value for LRU repair float in the General Support/Depot pipeline. When computing shipping costs, ASLI is the round trip shipping cost (SHTID) per item. For LRU demands, ASLI is the round trip pipe length (TIDT) in hours.

G. This section of BASIC computes either shipping costs or stock quantities to support the replacement of LRUs at a materiel installation. The parameters evaluated are a function of the removals and the shipping pipe times between maintenance levels. The removals are represented by the TU term; where TU=(A+F+FNG). The value at each maintenance level is summarized by adding the pipe line value between two levels to the value previously evaluated at the lower level.

1. QYE=TU*TRCE
   This statement will compute stock quantities at the Equipment level to support the removal and replacement of LRUs. Since there are no shipping costs associated with moving LRUs within the Equipment level, QYE will be zero when called from LOGAM to compute costs. TRCE is the down time in hours for a materiel system.

2. QYO=TU*(TRCE+ASLE)
   This statement adds the pipeline values (TU*ASLE) for the Equipment/Direct Support pipe to the Equipment level values (TU*TRCE). When computing costs, only the round trip shipping cost between Equipment and Direct Support is included since the shipping cost for QYE=0. For stock quantities summarized at Direct Support, the quantity tied up in the pipe (TU*ASLE) is added to QYE.

3. QYI=QYO+QUO+QFO

This statement summarizes either stock quantities or shipping costs for LRUs at General Support. The value computed at Direct Support (QYO) is added to the values previously computed for LRU Scrap (QUO) and LRU "float" (QFO) in the General Support/Direct Support pipe.

4. QYD=QYI+QUI+QFI
   This statement summarizes either stock quantities or shipping costs for LRUs at a Depot facility. The value computed at General Support (QYI) is added to the values previously computed for LRU scrap (QUI) and LRU "float" (QFI) in the Depot/General Support pipe.

5. QYF=QYD+QUD
   This statement computes either shipping costs or stock quantities for LRUs through the Factory/Depot pipe. The value computed at the Depot level (QYD) is added to the LRU scrap value (QUD) previously computed for the factory/Depot pipe.

6. RETURN
   END
   These statements terminate execution of BASIC and returns program logic to the calling program.

4.5.2 SUBROUTINE IOL

This subroutine computes initial provisioning quantities when using LOGAM supply rules. The stock computations are based on the mean demand quantity plus the safety stock coefficient times the square root of the mean demand quantity. IOL is called from the main program, LOGAM, for each provisioning type (LRU, module, part) at each supply echelon. The outputs of IOL are distributed equally for all stock points at a supply echelon and integer rounded using the "Z" parameters. The distribution function is performed in Subroutine STKDS.

1. SUBROUTINE IOL(XD,CKK,BQU,BQF,BQT)
   This statement assigns the argument names to match the variable lists from the calling program. The argument definitions are:

   XD - The number of supply points at a provisioning level.
   CKK - The safety stock coefficient.
   BQU - Mean demand quantity of scrapped stock.
   BQF - Mean demand quantity of "float" stock.
   BQT - The computed value for initial provisioning.

2. QUF=XD*(BQU+BQF)
   This statement computes the mean demand quantity for stock at all supply points of a provisioning level.

3. BQT=0.
   IF(XD.LE..5)RETURN
   IF(QUF.LE.0.)RETURN
   These statements return an initial provision quantity of zero if there is not a supply point (XD) at the level being considered or if the mean demand is less than or equal zero.

4. BSQ=SQRT(QUF)
   BQS=CKK*BSQ
   These statements add the safety stock to the mean quantity demand. The safety stock (BQS) is computed as the square root of the mean demand times the safety stock factor.

5. BQT=BQU+BQF+BQS
   This statement computes the initial provisions for all supply points at a provisioning level by summing the scrap, repair float, and safety stock requirements.

6. RETURN
   END
   These statements return logic back to the calling program. In this case it is LOGAM.

4.5.3 SUBROUTINE MPROV

This subroutine uses the LOGAM maintenance rule to compute initial provisioning. MPROV is called from the main program when the user has selected the maintenance rule by inputting AYZP greater than or equal to one. MPROV uses the maintenance flows evaluated in Subroutine MFLOW and the time charges input through NAMELIST/L/ to compute the spares.

A. This section contains the computer memory assignments for variables used in the subroutine. The lists for most of the common blocks are excluded here. These lists can be found in the main program.

1. COMMON/INPUT/
   This common block contains all of the variable names that are input through NAMELIST/L/. Refer to section A of LOGAM for a list of these variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),
   SL(4),TAT(4),TAYZ(10),ZM(4),
   ZP(3),ZU(4),SENSY(266),T(2000)
   This statement contains all of the dimensioned variables of common block INPUT.

3. COMMON/PROVQ/
   This common block contains the initial provisioning quantities for LRU, modules and parts.

4. COMMON/MNTFL/
   This common block contains the maintenance flows that were evaluated in Subroutine MFLOW.

5. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
   OR,AYZIS,AYZOS
   This common block contains some miscellaneous variables that are used in various subroutines of the program. SAVE is computd in the main program and is the only variable of this list that is used in this subroutine.

B. This section of MPROV computes provisioning days for LRUs at the following maintenance levels; E,DS,GS and Depot. Maintenance flows from Subroutine MFLOW are multiplied by input supply and maintenance times to get provisioning days. The following supply times used in this section are input in days.

DTE DTI DTO	Pipeline delays in handling repairables shipped rearward from E,DS, and GS, respectively.

TATE Time to get stock from supply at E.

OL Operating level of supply at E,DS,GS and Depot.

SL Safety level of supply at E,DS,GS and Depot.

OST Order and ship time for stock at E,DS,GS and Depot.

STAT Time to ship supply from rear-most of facility to the Depot.

The following inputs are the maintenance times. These inputs are all in hours except for TAT which is input in days. The maintenance times input in hours are converted to provisioning days as a function of the number of hours test and repair crews are scheduled to work in a week, i.e., TER/(24*WEM/168.).

TC Time to screen for false "no-gos".

TE
TF Test time at E,DS,GS and Depot, respectively.
TI
TD

TER
TFR Repair times at E,DS,GS and Depot, respectively.
TIR
TDR

WEM
WOM Scheduled work weeks for test crews at E,DS,GS and
WIM Depot, respectively.
WDM

WER
WOR Scheduled work week for repair crews at E,DS,GS and

WIR Depot, respectively.
WDR

TAT Maintenance turn-around times at E,DS,GS and Depot.

1. OFACT = SAVE*8766./365
   This statement computes an operating factor that will convert the provisioning days from clock time at one Equipment installation to "real time" operation days for all ED locations. The operating time is a function of the fraction of the time the system is operational (OTF) and inherent availability (AYZIS).

2. PDRE = TU*(TRC/(24.*D(WER)/168.)+TAT(1)+TATE)
This statement computes the provisioning days required to account for removal of LRUs at the equipment level. TU is the total removals (A+F+FNG).

3. PDEUE1=(DSUE1+GSUE1+DUE1)*DTE
This statement computes the provisioning days for those LRUs that are evacuated from the Equipment level because there is not a maintenance policy to repair at ED. The evacuations are to Direct Support (DSUE1) General Support (GSUE1) and Depot (DUE1).

4. P D E U E 2 =
(DSUE2+GSUE2+DUE2)*(DTE+(TE+TC)/(24.*D(WEM)/168.))
This statement computes the provisioning days for those LRUs that are evacuated from the Equipment level to higher levels of maintenance because of their "in limbo" status. These LRUs have a maintenance policy to repair at ED, but are not repaired because of the FUE fraction. The evacuations are to Direct Support (DSUE2), General Support (GSUE2) and Depot (DUE2).

5. PDEUE=PDEUE1+PDEUE2
This statement computes the total number of provisioning days at Equipment level due to evacuated LRUs.

6. PDSUE=(SUE1+SUE2+SUE3)*(OL(1)+SL(1)+OST(1))+
(SUE2+SUE3)*(TC+TE)/(24.*D(WEM)/168.)
This statement computes the provisioning days at Equipment level for scrapped LRUs. The LRUs scrapped here are those that have a maintenance policy to discard (SUE1), those "in limbo" and do not have a higher maintenance level (SUE2), and those declared not repairable after testing (SUE3).

7. PDSKUE=FGE*TC/(24.*D(WEM)/168.)
This statment computes the provisioning days at the Equipment level to test and identify false "no gos" (FGE).

8. PDAU=A*(OL(1)+SL(1)+OST(1))
This statement computes the provisioning days at the Equipment level to account for the LRU stock that is lost due to attrition.

9. PDRUE=TUFTE*(168.*((TC+TE)/D(WEM)+TER/D(WER))/24.)
This statement computes the provisioning days to repair LRUs at the Equipment level. TUFTE is the flow of LRUs through the checkout and test facilities that were declared repairable.

10. PDUE=(PDRE+PDEUE+PDSUE+PDAU+PDSKUE+PDRUE)*OFACT
This statement sums the provisioning days at the Equipment level for LRUs removed, evacuated, scrapped, and repaired. OFACT is then used to compute the quantity of initial LRU provisions at ED as a function of "real time" operation at all ED locations.

11. PDEUO1=(GSUO1+DUO1)*(DTO+TC/(24.*D(WOM)/168.)+TAT(2))
This statement computes the provisioning days for those LRUs that are evacuated from the Direct Support level to higher maintenance levels because there is not a maintenance policy to repair at DS. The evacuations are to General Support (GSU01) and Depot (DU01).

12. PDEUO2=(GSUO2+DUO2)*(DTO+(TC+TF)/(24.*D(WOM)/168.)+
TAT(2))
This statement computes the provisioning days for those LRUs that are evacuated from Direct Support to higher levels of maintenance because of their "in limbo" status. These LRUs have a maintenance policy to repair at DS, but are not repaired because of the FUO fraction. The evacuations are to General Support (GSU02) and Depot (DU02).

13. PDEUO=PDEUO1+PDEUO2
This statement computes the total number of provisioning days at Direct Support due to evacation of LRUs to a higher maintenance level.

14. PDSUO=(SUO1+SUO2+SUO3)*(TC/(24.*D(WOM)/168.)+TAT(2)+
OL(2)+SL(2)+OST(2))+(SUO4+SUO5)*((TC+TF)/(24.
*D(WOM)/168.)+TAT(2)+OL(2)+SL(2)+OST(2))
This statement computes the provisioning days for LRUs scrapped at Direct Support. These scrap rates are for maintenance policies that screen for false "no gos" at DS and evacuate LRUs for repairs at GS(SU01) and Depot(SU02), policy to detect false "no gos" at DS and discard failures (SU03), "in-limbo" units evacuated from ED(SU04), and units deemed not repairable at DS(SU05).

15. PDSKUO=FGO*(TC/(24.*D(WOM)/168.)+TAT(2))
This statement computes the provisioning days at Direct Support to test and identify false "no go" LRUs.

16. PDRUO=TUFTO*(168.*((TC+TF)/D(WOM)+TFR/D(WOR))/24.0+
TAT(2))
This statement computes the provisioning days at Direct Support to repair LRUs. TUFTO is the flow of LRUs through the DS test facility that were declared repairable.

17. PDUO=(PDEUO+PDSUO+PDSKUO+PDRUO)*OFACT
This statement sums the provisioning days at Direct Support for LRUs that are evacuated, scrapped, tested as false "no go's", and repaired. OFACT is then used to compute the quantity of initial LRU provisions at DS as a function of "real time" operations of all ED locations.

18. PDEUI=DUI*((TC)/(24.*D(WIM)/168.)+TAT(3)+DTI)
This statement computes the provisioning days at General Support for LRUs evacuated (DUI) to a higher maintenance level (Depot) because there is not a maintenance policy to repair at GS.

19. PDSUI=(SUI1+SUI2)*((TC+TI)/(24.*D(WIM)/168.)+
TAT(3)+OL(3)+SL(3)+OST(3))
This statement computes the provisioning days for LRUs scrapped at General Support. The scrap rates are the "in-limbo" units from DS that are still in limbo at GS(SUI1) and units at GS declared not repairable (SUI2).

20. PDSKUI=FGI*(TC/(24.*D(WIM)/168.)+TAT(3))
This statement computes the provisioning days at General Support required to test and identify false "no go" LRUs.

21. PDRUI=TUFTI*(168.*(TC+TI)/D(WIM)+TIR/D(WIR))/24.+
TAT(3))
This statement computes the provisioning days at General Support to repair LRUs. TUFTI is the flow of LRUs through the test facility that were declared repairable.

22. PDUI=(PDEUI+PDSUI+PDSKUI+PDRUI)*OFACT
This statement sums the provisioning days at General Support for LRUs that are evacuated, scrapped, tested as false "no go's", and repaired. OFACT is then used to compute the quantity of initial LRU provisions as a function of "real time" operations at all ED locations.

23. PDSUD=SUD1*((TC+TD)/(24.*D(WDM)/168.)+
TAT(4)+OL(4)+SL(4)+OST(4))
This statement computes the provisioning days for LRUs scrapped at the Depot (SUD1).

24. PDSKUD=FGD*(TC/(24.*D(WDM)/168.)+TAT(4)+STAT)
This statement computes the provisioning days at the Depot for checkout, turn around, and shipping of false "no go" LRUs.

25. PDRUD=TUFTD*(168.*((TC+TD)/D(WDM)+TDR/D(WDR))/24.+
TAT(4)+STAT)
This statement computes the provisioning days at the Depot required for checkout, test, and repair of LRUs. TUFTD is the flow of LRUs through the test facility that were declared repairable.

26. PDUD=(PDSUD+PDSKUD+PDRUD)*OFACT
This statement sums the provisioning days for LRUs at the Depot that are scrapped, tested for false "no go's", and repaired. OFACT is then used to compute the quantity of intial provisions at DEPOT as a function of "real time" operations at all ED locations.

27. PQTU=PDUE+PDUO+PDUI+PDUD
This statement sums the initial LRU provisions for all stock locations.

C. This section of MPROV computes provisioning days for modules at the following maintenance levels; E,GS,DS and Depot. Maintenance flows from Subroutine MFLOW are multiplied by input supply and maintenance times to get provisioning days for modules. The input supply times described in Section B for LRUs are also used in computing module provisions. Only the test and repair times listed below are different than those times used for LRUs.

TMO TMI TMD	Test times for modules at DS,GS and Depot, respectively.
TMOR TMIR TMDR	Repair times for modules at DS,GS and Depot respectively.

Since there are no maintenance policies to repair modules at the Organizational level, there are no times input to test and repair modules at E.

1. PDEME=TMFEO*(TAT(1)+DTE)
This statement computes provisioning days at the Equipment level for modules evacuated to higher echelons of maintenance (TMFEO).

2. PDSME=TSME*(TAT(1)+OL(1)+SL(1)+OST(1))
This statement computes the provisioning days at the Equipment level for scrapped modules (TSME).

3. PDME=(PDEME+PDSME)*OFACT
This statement sums the provisioning days for evacuations and scrap modules at the Equipment level.

OFACT is used to compute the initial module provisions as a function of "real time" operations at all ED locations.

4. PDEMO=(GSMO+DMO)*(TAT(2)+DTO)
This statement computes the provisioning days for modules evacuated from Direct Support to General Support (GSMO) and Depot (DMO).

5. PDSMO1=SMO1*(TMO/(24.*D(WOM)/168.))
This statement computes the provisioning days for testing modules that are scheduled for repair at Direct Support but were declared not repairable (SMO1).

6. PDSMO2=TSMO*(TAT(2)+OL(2)+SL(2)+OST(2))
This statement computes the provisioning days for replacing all scrapped modules at Direct Support (TSMO).

7. PDSMO=PDSMO1+PDSMO2
This statement computes the provisioning days at Direct Support for the testing and replacement of scrapped modules.

8. PDRMO=TMFO*(168.*(TMO/D(WOM)+TMOR/D(WOR))/24.+TAT(2))
This statement computes the provisioning days at Direct Support for testing and repairing modules. TMFO are the repairable modules.

9. PDMO=(PDEMO+PDSMO+PDRMO)*OFACT
This statement sums the provisioning days for modules that are evacuated, scrapped, and repaired at Direct Support. OFACT is used to compute the quantity of initial module provisions at DS as a function of "real time" operations at all ED locations.

10. PDEMI=DMI*(TAT(3)+DTI)
This statement computes the provisioning days for modules that are evacuated from General Support to the Depot (DMI).

11. PDSMI1=SMI1*(TMI/(24.*D(WIM)/168.)+
TAT(3)+OL(3)+SL(3)+OST(3))
This statement computes the provisioning days at General Support for modules that are scheduled for repair at GS but after testing are declared not repairable (SMI1).

12. PDSMI2=(SMI2+SMI3)*(TAT(3)+OL(3)+SL(3)+OST(3))
This statement computes the provisioning days at General Support for modules that are scrapped because the maintenance policy is to discard (SMI2) and to

evacuate to the (Depot (SMI3). These modules are not charged test time (TMI).

13. PDSMI=PDSMI1+PDSIM2
This statement computes the provisioning days for all modules scrapped at General Support.

14. PDRMI=TMFI*(168.*(TMI/D(WIM)+TMIR/D(WIR))/24.+TAT(3))
This statement computes the provisioning days for modules tested and repaired at General Support (TMFI).

15. PDMI=(PDEMI+PDSMI+PDRMI)*OFACT
This statement sums the provisioning days at General Support for modules evacuated, scrapped, and repaired. OFACT is used to compute the initial module provisions at GS as a function of "real time" operations at all ED locations.

16. PDSM1=SMD1*(TMD/(24.*D(WDM)/168.)+TAT(4)+OL(4)+
SL(4)+OST(4))
This statement computes the provisioning days at the Depot for modules that are tested and declared not repairable (SMD1).

17. PDSM2=SMD2*(TAT(4)+OL(4)+SL(4)+OST(4))
This statement computes the provisioning days at the Depot for modules that are scrapped due to a maintenance "discard" policy and/or after screening for false "no go's" (SMD2).

18. PDSMD=PDSM1+PDSM2
This statement computes the provisioning days for all module scrap at the Depot.

19. PDRMD=TMFD*(168.*(TMD/D(WDM)+TMDR/D(WDR))/24.+
TAT(4)+STAT)
This statement computes the provisioning days at the Depot for modules tested and repaired (TMFD).

20. PDMD=(PDSMD+PDRMD)*OFACT
This statement sums the provisioning days at the Depot for modules that are scrapped and repaired. OFACT is used to compute the quantity of initial module provisions at the Depot as a function of "real time" operations at all ED locations.

21. PQTM=PDME+PDMO+PDMI+PDMD
This statement sums the initial module provisions of all stock locations into a system total.

D. This section of MPROV computes the provisioning days for parts at the following maintenance levels; DS,GS, and Depot. The

provisioning days for repair of modules is used as the provisioning days for parts.

1. PDPO=PDRMO*OFACT
   PDPI=PDRMI*OFACT
   PDPD=PDRMD*OFACT
   These statements compute the initial provisions for parts at Direct Support, General Support, and Depot, respectively. The provisioning days for module repair is used to provision for parts since each time a module is repaired, a part is required.

2. PQTP=PDPO+PDPI+PDPD
   This statement sums the initial provisions for parts at Direct Support, General Support, and Depot, into a total system provisioning.

3. RETURN
   END
   These statements terminate the execution of MPROV and returns logic to the main program.

4.5.4 Subroutine REPLC

This subroutine is used to set the predetermined initial provisions as the computed values. Computed values using either Supply Rules or Maintenance Rules will be replaced by any quantities input in NAMELIST/L/. LOGAM initializes initial provisions to zero before an input case is read, therefore it is assumed that an input value will have preference over a computed value. Any value input greater than .999 is considered to be a legitimate input quantity.

Computed provisions are distributed evenly over the supply points at an echelon and if the predetermined quantities aren't input to this effect then they will be modified by Subroutine STKDS. Predetermined quantities may also be modified if the Over-Ride (OR) factor for AYZP is greater than 50 percent. OR causes stock to either be added or subtracted in order that the desired operational avaialability be met. This modification would occur in Subroutine AVAIL.

A. This section of REPLC contains the assignment of program variables to computer memory.

1. COMMON/INPUT/
   This common block contains the list of variables that are input by NAMELIST/L/. The LRU, module and part initial provisioning quantities are the only variables used in this subroutine. Refer to section A of the main program for a complete list of these variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAYZ(10),
   ZM(4),ZP(3),ZU(4),SENSY(266),T(2000)
   This statement contains the dimensioned variables of COMMON/INPUT/.

3. COMMON/PROVQ/DELU,PDUE,PDUO,PDUI,PDUD,PQTU,
   PDME,PDMO,PDMI,PDMD,PQTM,
   PDPO,PDPI,PDPD,PQTP,
   PQTE,PQTO,PQTI,PQTD,PQT,QT,QTM,QTP
   This common block contains computed and required initial provisions. Required quantities differ from computed quantities when computed quantities are modified by trying to meet a predetermined operational availability (OR). Computed quantities are determined by the maintenance and supply rules or they are predetermined values. Required provisions are distributed in Subroutine STKDS and are the same as computed values unless the Over-Ride (OR) factor is used. PDUE through PQTP are the computed values and PQTE through PQT

are the required values.

B. This section assigns predetermined (input) initial provisions to the computed provisioning variable list. The input variables (QTE,QTME,QTPO,etc.) are included in the COMMON/INPUT/list. These variable names are also used as the distributed provisioning quantities. If any of these quantities are input they will replace their respective "computed" (PDUE,PDME,PDPO,etc.) values. In Subroutine AVAIL the computed values for LRUs are stored into the "required" (PQTE,PQTO,PQTI,PQTD) values. If OR is in force the required LRU values are modified to meet the desired availability. Modules and parts are not changed. The "required" LRUs and "computed" modules and parts are distributed in Subroutine STKDS as QTE,QTME,QTPO,etc.

1. IF(QTE.GT.0.999) PDUE=QTE
   IF(QTO.GT.0.999) PDUO=QTO
   IF(QTI.GT.0.999) PDUI=QTI
   IF(QTD.GT.0.999) PDUD=QTD
   These statements will replace a "computed" LRU provision with the input quantity. Provisioning quantities should be input in whole units but in case of a possible round-off error the value to be exceeded is 0.999.

2. IF(QTME.GT.0.999) PDME=QTME
   IF(QTMO.GT.0.999) PDMO=QTMO
   IF(QTMI.G5.0.000) PDMI=QTMI
   IF(QTMI.GT.0.999) PDMD=QTMD
   These statements will replace a "computed" module provision with the input quantity.

3. IF(QTPO.GT.0.999) PDPO=QTPO
   IF(QTPI.GT.0.999) PDPI=QTPI
   IF(QTPD.GT.0.999) POPD=QTPD
   These statements will replace a "computed" part provision with the input quantity.

4. PQTU=PDUE+PDUO+PDUI+PDUD
   PQTM=PDME+PDMO+PDMI+PDMD
   PQTP=PDPO+PDPI+PDPD
   These statements total the "computed" initial provisions for LRUs, modules and parts, respectively.

5. RETURN
   END
   These statements terminate execution of this subroutine and return logic to LOGAM.

4.5.5 SUBROUTINE STKDS

This subroutine distributes required initial provisions equally for each authorized stock point of a supply echelon. Required provisions can be the computed values using either the LOGAM Maintenance Rules (Subroutine MPROV), the Supply Rules (Subroutine Basic and IOL), or predetermined (INPUT). The required and the computed values will be the same unless the user drives the operational availability with the AYZP Over-Ride (OR) input. The computed LRU provisions will be modified in Subroutine AVAIL to meet an operational availability if one is requested. These modified provisions become the required values. The values for modules and parts are not effected.

A. This section of STKDS contains the assignment of variables in computer memory.

1. COMMON/INPUT/
   This common block contains the list of all the input variables for NAMELIST/L/. Refer to section A of the main program for the list of variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAT(4),TAYZ(10)
   ZM(4),ZP(3),ZU(4),SENSY(266),T(2000)
   This statement is a list of the dimensioned variables of common block INPUT.

3. COMMON/PROVQ/DELU,PDUE,PDUO,PDUI,PDUD,PQTU,
   PDME,PDMO,PDMI,PDMD,PQTM,
   PDPO,PDPI,PDPD,PQTP,
   PQTE,PQTO,PQTI,PQTD,PQT,
   QT,QTM,QTP
   This common block contains the provisioning quantities for LRUs, modules and parts at each supply echelon and the totals. The LRU provisions computed by either the Maintenance Rules or the Supply Rules are PDUE,PDUO,PDUI AND PDUD, where PQTU is the total. These variables also represent the predetermined LRU provisions. Predetermined stock is stock that is input through NAMELIST/L/. Predetermined stock will over-ride computed stock values. PQTE,PQTO,PQTI and PQTD are the required LRU provisions that get distributed in this subroutine. They will have the same value as their computed counterpart unless operational availability is driven in subroutine AVAIL.

B. This section of STKDS distributes the LRU, module, and part stock according to the authorized stock points. The H matrix, which is input, determines the presence or absence of stock at a

stock level. An H value of one (1) authorizes stock at a location. Computed provisioning quantities are distributed equally over all stock points at a particular stock level and rounded to whole units as a function of the "Z" parameter. Any provisions "borrowed" or "left over" from the rounding function will be accounted for in the distribution at the next higher authorized stock level. When stock is not authorized at a particular level the stock is distributed at the next lowest authorized level. The equipment level stock is an exception since there is not a lower stock level. In this case the stock is "left over" and will be distributed at the next higher authorized level.

1. QTE=EDS*AINT((PQTE+(1.-H(2))*PQTO+(1.-H(2)*
(1.-H(3))*PQTI+(1.-H(2))*(1.-H(3))*
(1.-H(4))*PQTD/D(EDS)+ZU(1))*H(1)
This statement, if H(1)=1, will distribute the LRU stock computed at the Equipment level (PDUE) equally over all ED stock points. The stock computed at the next higher echelon(s) for which stock is not authorized will also be distributed here. The distributed stock is rounded (ZU) to whole units.

2. DQ=PQTE-QTE
This statement computes the quantity of LRU stock "borrowed" or "left over" after distribution at the ED level. When the ZU fraction causes a round up of the stock quantity, stock is borrowed from the next higher echelon. Also the stock computed for a higher echelon that does not have stock authorization will be distributed at ED and this stock is also considered borrowed. The left over stock occurs when the computed quantity is rounded off because of the ZU fraction.

3. QTO=ODS*AINT((PQTO+DQ+(1.-H(3))*PQTI+(1.-H(3))*
(1.-H(4))*PQTD/D(ODS)+ZU(2))*H(2)
This statement, if H(2)=1, will distribute the LRU stock computed at the Direct Support level (PDUO) equally over all OD stock points. The stock computed at OD will be adjusted by the "borrowed" or "left over" stock quantity from the ED level before the distribution is made. The stock computed at the next higher echelon(s) for which stock is not authorized will also be distributed here.

4. IF(QTO.LT.0.)QTO=0.
This statement will set the LRU stock quantity at Direct Support to zero when the quantity of stock to be distributed at DS is less than the quantity "borrowed" from the ED level.

5. DQ=PQTO-QTO+PQTE-QTE

This statement computes the quantity of LRU stock either "borrowed" or "left over" when stock was distributed at the ED and OD levels.

6. QTI=DIS*AINT((PQTI+DQ+(1.-H(4))*PQTD/D(DIS)+ ZU(3))*H(3)
This statement, if H(3)=1, will distribute the LRU stock computed at the General Support level (PDUI) equally over all DI stock locations. The stock borrowed or left over from distributions at ED and OD is used to adjust the computed quantity before a distribution is made. The stock computed at Depot will also be distributed here if stock is not authorized at the Depot.

7. IF(QTI.LE.0.)QTI=0.
This statement will set the LRU stock quantity at General Support to zero when the quantity of stock to be distributed at GS is less than the quantity "borrowed" at the lower stock levels.

8. DQ=PQTI-QTI+PQTE-QTE+PQTO-QTO
This statement computes the quantity of LRU stock "borrowed" or "left over" when stock was distributed at the GS, DS, and E levels.

9. QTD=DDS*AINT((PDUD+DQ)/D(DDS)+ZU(4))*H(4)
This statement, if H(4)=1, will distribute LRU stock computed at the Depot level (PDUD) equally over all Depot stock locations. The stock "borrowed" or "left over" from distributions at E, DS, and GS is used to adjust the computed quantity before a distribution is made.

10. IF(QTD.LT.0.)QTD=0.
This statement will set the LRU stock distributed at the Depot to zero if the quantity "borrowed" at the lower echelon distributions is greater than the computed quantity.

11. QTME=EDS*AINT((PDME+(1.-H(2))*PDMO+(1.-H(2))*(1.-H(3))* PDMI+(1.-H(2))*(1.-H(3))*(1.-H(4))*PDMD)/D(EDS)+ ZM(1))*H(1)
This statement distributes the computed module stock (PDME) equally over all ED stock locations. The logic for distributing modules is the same as the distribution for LRUs.

12. DQ=PDME-QTME
This statement computes the quantity of module stock either "borrowed" on "left over" in the Equipment level distributions.

13. QTMO=ODS*AINT((PDMO+DQ+(1.-H(3))*PDMI+(1.-H(3))
*(1.-H(4))*PDMD)/D(ODS)+ZM(2))*H(2)
This statement distributes the computed module stock (PDMO) at Direct Support equally over all OD stock locations.

14. IF(QTMO.LT.0.)QTMO=0.
This statement sets the module stock distributions at Direct Support to zero when the quantity "borrowed" from the Equipment distribution is greater than the quantity to be distributed at OD.

15. DQ=PDMO-QTMO+PDME-QTME
This statement computes the quantity of module stock either "borrowed" or "left over" when distributions where made at the DS and E levels.

16. QTMI=DIS*AINT((PDMI+DQ+(1.-H(4))*PDMD)/D(DIS)+
ZM(3))*H(3)
This statement distributes the computed module stock (PDMI) at General Support equally over all DI stock locations.

17. IF(QTMI.LT.0)QTMI=0.
This statement sets the module stock distribution at General Support to zero when the quantity "borrowed" for the Equipment and Direct Support locations is greater than the quantity to be distributed at GS.

18. DQ=PDMI-QTMI+PDME-QTME+PDMO-QTMO
This statement computes the quantity of module stock either "borrowed" or "left over" after distributions are made at the Equipment, Direct Support, and General Support stock locations.

19. QTMD=DDS*AINT((PDMD+DQ)/D(DDS)+ZM(4))*H(4)
This statement distributes the computed module stock (PDMD) at the Depot equally over all DD stock locations.

20. IF(QTMD.LT.0.)QTMD=0.
This statement sets the module stock distribution at the Depot to zero when the stock "borrowed" for distributions at E, DS, and GS is greater than the quantity computed (PDMD) for the Depot.

21. QTPO=ODS*AINT((PDPO+(1.-H(3))*PDPI+(1.-H(3))*
(1.-H(4))*PDPD)/D(ODS)+ZP(1))*H(2)
This statement distributes the computed part stock (PDPO) at Direct Support equally over all DS stock locations. The logic for distributing part stock is

the same as the distribution for LRU stock.

22. DQ=PDPO-QTPO
This statement computes the quantity of part stock either "borrowed" or "left over" when distributions were made at Direct Support.

23. QTPI=DIS*AINT((PDPI+DQ+(1.-H(4))*PDPD)/
D(DIS)+ZP(2))*H(3)
This statement distributes the computed part stock (PDPI) at General Support equally over all GS stock locations.

24. IF(QTPI.LT.0.)QTPI=0.
This statement sets the part stock distribution at General Support to zero when the stock "borrowed" for distribution at DS is greater then the quantity to be distributed at GS.

25. DQ=PDPI-QTPI+PDPO-QTPO
This statement computes the quantity of part stock either "borrowed" or "left over" when distributions were made at DS and GS.

26. QTPD=DDS*AINT((PDPD+DQ)/D(DDS)+ZP(3))*H(4)
This statement distributes the computed part stock (PDPD) at Depot equally over all DD stock locatons.

27. IF(QTPD.LT.0)QTPD=0.
This statement sets the part stock distribution at Depot to zero when the stock "borrowed" for distribution at DS and GS is greater than the quantity to be distributed at the Depot level.

28. QT=QTE+QTO+QTI+QTD
QTM=QTME+QTMD+QTMI+QTMD
QTP=QTPO+QTPI+QTPD
These statements total the initial provisions for LRUs, modules and parts, respectively.

29. RETURN
END
These statements terminate execution of STKDS and returns logic to LOGAM.

4.5.6 SUBROUTINE REORDR

Subroutine REORDR computes stock reorder quantities based on the material consumed over the system lifecycle and the lot size. Quantity of consumed material, lot sizes, reorder actions and residual stock are all determined in this subroutine.

REORDR computes reorders for LRUs, modules and parts with a separate call to the routine from LOGAM, the main program. The inputs and outputs to the routine are passed through the argument list. The next statement describes the argument list.

1. SUBROUTINE REORDR(QUA,QT,QU,QTD,SUT,SUTD,PUR,YPF,YR8, QMU,UF,QB,RQN,QC,QCD,RU,RUD,QQC)
   This is the entry point for this subroutine with all the input and output being transferred through the argument list. Variables QUA through QMU are the inputs and UF through QQC are the outputs. The variables are described in the following statements remembering that the variables can be for either LRUs, modules, or parts.

INPUTS

QUA	Initial Production quantity. Includes installed equipment and initial provisions.
QT	Initial provisions at all supply levels.
QU	Pipeline scrap quantity.
QTD	Initial provisions at Depot supply.
SUT	Scrap rate at all maintenance locations.
SUTD	Scrap rate at the Depot maintenance facilities.
PUR	Input production rate in units per hour.
YPF	Length of the acquisition phase in hours.
YR8	Length of the O&M phase in hours.
QMU	Minimum reorder lot size.

OUTPUTS

UF	Initial provisions at the field supply levels.
QB	Reorder lot size.
RQN	Number of reorder transactions during O&M phase.
QC	Total reorders during O&M phase.
QCD	Reorders for the Depot levels.
RU	Total residuals at end of program.
RUD	Residuals at the Depot.
QQC	Quantity of consumed material over life of system.

```
QQC=YR8*SUT
QQCD=YR8*SUTD
```

These statements use scrap rates to compute the material consumed during the O&M phase (YR8) for all maintenance levels and for Depot, respectively.

3.
```
UF=QT-QTD
```

This statement computes the initial provisions for the field supply points.

4.
```
PURX=QUA/D(YPF)
PURY=PUR
IF (PURY .LE. 0.1E-19) PURY=PURX
PURZ=2.0*SUT
IF (PURY .LT. PURZ) PURY=PURZ
```

These statements determine the production rate which in turn will be used to compute the buy lot size. The production rate (PURY) must be at least twice the scrap rate (SUT) which is computed as PURZ. If the production rate is not input (PUR) then the rate (PURX) is computed from the initial buys (QUA) and the length of the acquisition phase (YPF). The function D in the statement PURX=QUA/D(YPF) is used to protect against an overflow condition by dividing by zero.

5.
```
QB=AINT(QU/(1.-SUT/PURY))
IF (QB .LT. QMU) QB=QMU
```

These statements compute the reorder lot size in rounded off units. Because of the method of determining PURY, the lot size will range between the value of QU and twice the value of QU unless the user inputs a lot size (QMU) which is greater than the computed lot size. QU is the scrap tied up in the pipelines per supply interval.

6.
```
QQ=AMAX1(0.,QQC-QT)
RQN=AINT(QQ/D(QB)+.99)
```

These statements compute the number of reorder transactions by dividing the quantities to be ordered (QQ) by the lot size (QB) and rounding up. QQ is the material that will be comsumed in addition to the material provisioned (QT).

7.
```
QC=RQN*QB
QCD=AINT(QC*SUTD/D(SUT)+.5)
```

These statements compute the quantity of reorder buys for all supply levels (QC) and for the Depot (QCD). The Depot reorders are computed as a ratio of the Depot scrap rate and the total scrap rate.

8.
```
RU=AMAX1(0.,(QT+QC-QQC))
RUD=AMAX1(0.,(QTD+QCD-QQCD))
```

These statements compute the residuals at all supply levels and at Depot, respectively. The residuals are the difference between consumed material (QQC) and the material that is stocked initially (QT) plus the reorders (QC).

9. RETURN
END
These statements terminate execution of REORDR and transfers logic back to the calling routine (LOGAM).

4.6 LOGAM Test Equipment and Manpower Demands

The subroutines described in this section use the maintenance flows from Subroutine MFLOW to compute test and repair channel maintenance factors. The 2 subroutines of this section are:

1. TEMWF — Computes test equipment and manpower demands as a function of maintenance flow.

2. PERSON — Computes personnel demands as a function of the maintenance manhour requirements evaluated in Subroutine TEMWF. This subroutine also prints the personnel outputs when case totals are requested.

4.6.1 SUBROUTINE TEMWF

This subroutine computes workload demands for maintenance on installed equipments. The routine begins by computing the manhours required to test and repair equipment at the Organization, Direct Support, General Support, and Depot maintenance facilities. Demands for manpower and test equipment are then computed by dividing manhours by the number of hours in a work week.

The manhour requirements are computed by multiplying the maintenance flows through the checkout and repair facilities with the checkout, test and repair times. The maintenance flows are in units of LRUs and modules through the maintenance facilities per hour. Maintenance flow is evaluated in Subroutine MFLOW. The time charges are input through the NAMELIST/L/ data.

Workload demands are either accumulated among groups of LRU cases or dedicated to a single LRU case. This is controlled by the flags (ETE, ETI, ETII) that post out work demands. While a flag is zero, work demands are accumulated for individual LRU cases. When a posting flag is set to unity the manhour accumulations are used to compute manpower and test equipment demands. The variables for accumulating manhours are then reinitialized to zero in LOGAM.

There are three different accumulations for manhours that are made in TEMWF. One was described above that allows dedicating manpower to groups of LRUs in a concept. A second is for expected shared manhours where all LRUs in a concept are accumulated. This accumulation is not used for computing work demands, but as an output for comparison purposes only. A third accumulation (HPD) of manhours is similiar to the second description except that it includes the demand for test equipment self-support and it's units are manhours per day instead of manhours per hour. The HPD accumulations are used in Subroutine PERSON to compute the maintenance personnel count. All of the accumulations are reset to zero in LOGAM once a data concept is completed. To end a data concept, the user inputs IS=1 with the last data case in the concept. A case total or grand total output selection (NU=-2 or NU=-3) will also end a concept.

Manpower and test equipment demands can be posted out either as expected values (shared prorata utilization) or as integer round-off values governed by the ZFL input. A value of unity for an expected value flag gets the expected value. A value of zero gives the round-off. Unity and zero are the only two values that are permitted.

TEMWF is called from LOGAM (main subroutine) for each individual LRU case input.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (TEMWF)

A. This section of TEMWF contains the assignment of program variables to computer memory.

1. COMMON/INPUT/
   This statement includes the variable list from the NAMELIST/L/ inputs. Refer to LOGAM for a description of the variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAT(4),
   TAYZ(10),ZM(4),ZP(3),ZU(4),SENSY(266),T(2000)

   This statement includes all of the variables in COMMON/INPUT/ that are dimensioned.

3. COMMON/MMHWF/AEY,AERY,AOY,AORY,AIY,AIRY,ADY,ADRY,
   SAEY,SAERY,SAOY,SAORY,SAIY,SAIRY,SADY,SADRY,
   CAEY,CAERY,CAOY,CAORY,CAIY,CAIRY,CADY,CADRY,
   ESU,ESUM,ESUR,ESUY,ESURY,DSU,DSUM,DSUR,DSUY,
   DSURY,GSU,GSUM, GSUR,GSUY,GSURY,DEP,DEPM,DEPR,
   DEPY,DEPRY, AME,SAME,DSIM,ESI,ESME,DEPAIE,
   DEPAIM,DEPAIY,DEPAR,DEPARY
   This common block contains the manhours, manpower, and test equipment demands for maintenance at all echelons. These variables are computed in Subroutine TEMWF.

4. COMMON/MNTFL/DSUE1,GSUE1,DUE1,DSUE2,GSUE2,DUE2,SUE1,
   SUE2,SUE3,GSUO1,DUO1,GSUO2,DUO2,SUO1,SUO2,
   SUO3,SUO4,SUO5,GSMO,SMO1,DMO,DMI,SMI1,SMI2,
   SMI3,SMDI,SMD2,SUD1,DUI,SUI1,SUI2,FGE,FGO,
   FGI,FGD,TUFTE,TUFTO,TUFTI,TUFTD,TUFEC,TUFOC,
   TUFIC,TUFDC,TSME,TSMO,TSMI,TSMD,TMFO,TMFI,
   TMFD,TMFEO,TMFOI,TMFID,TSPD,TSPI,TSPO,TSU,
   TSLF,TSLD,F,FNG,A,TU,DPT
   This common block contains the list of variables for the maintenance flow of LRUs, modules and parts through the test and repair facilities at each echelon. These variables are evaluated in Subroutine MFLOW and are used in Subroutines BASIC and MPROV to compute provisioning quantities.

5. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
   OR,AYZIS,AYZOS
   This statement contains a miscellaneous list of variables of which four are used in TEMWF. SAVE and QUA are computed in LOGAM as a conversion factor for flow of material from clock time to real operating time and the initial LRU production rate, respectively. FMWO and AAIE are set in TEMWF. FMWO is the frequency of modification work orders and AAIE is a flag that determines if Type II test equipment is used at the Depot.

6. COMMON/PERSON/PERS(5,2),PERL(5,2),HPD(5,2),
WPD(5,2),WPDL(5,2)
This statement contains the personnel manpower demands for LOGAM maintenance. HPD is the manhour demands evaluated in this subroutine. The remaining variables are computed in Subroutine PERSON using the data from HPD.

B. This section of TEMWF is for initialization.

1. FMWO=YMWO/8766.
TMWO=FMWO*ZO
TMWI=FMWO*ZI
TMWD=FMWO*(1.-ZO-ZI)
These statements compute the frequency of modification work orders for Direct Support, General Support, and Depot, respectively. The frequency is in orders per hour per LRU. ZO and ZI are the fraction of work orders performed at DS and GS, respectively. The values ZO and ZI should never be input such that their sum would be greater than one.

2. SAVEEB=SAVE/D(ED)
SAVEOB-SAVE/D(OD)
SAVEIB=SAVE/D(DI)
SAVEDB=SAVE/D(DD)
These statements compute factors based on operational time for the effective number of installations supported by maintenance facilities at Organization, DS, GS, and Depot, respectively.

C. This section computes work demands to fault isolate, remove, and replace LRUs at the equipment installation site. Manpower and test equipment demands are computed by dividing the manhour requirements by the number of hours in a workweek. Expectd value flags are used to get demands as either expected values or integer round-off values. The ZFL factor determines the round-off procedure.

1. AME=SAVEEB*TU*TRC
This statement computes the number of manhours required to fault isolate, remove, test, and replace LRUs at one installation. SAVEEB converts to the number of effective installations supported by one equipment maintenance facility.

2. SAME=SAME+AME
This statement accumulates the manhours to fault isolate, remove, replace, and test all LRUs at one

equipment location.

3. ESME=ETE*SAME*(1.+FE)
This statement adds the manhours for self-support of type V test equipment to SAME and controls posting out of accumulated work demands. ETE=1, posts work demands, ETE=0 removes work demands.

4. DSIM=EREI*ED*168.*(ESME/D(WMR))+
((1.-EREI)*ED*AINT((168.*
ESME/D(WMR))+ZFL))
This statement computes the number of unscheduled maintenance personnel to do LRU fault isolation, remove, replace, and test at all equipment installations. EREI is the expected value flag. EREI=1 gives expected value and EREI=0 gives integer round-off.

5. ESI=(ETEI*ED*168.*(1.-RF)*
(ESME/D(WMT)))+((1.-ETEI)*ED*
AINT((168.*(1.-RF)*ESME/D(WMT))+ZFL))
This statement computes the demand for Type V test equipment to fault isolate and test LRUs at all equipment installations. ETEI is the expected value flag. ETEI=1 gives the expected value and ETEI=0 gives integer round-off. The factor "1.-RF" in the equation excludes the remove and replace manhour requirements from ESME.

D. This section computes the manhours required to test and repair the LRUs through a maintenance facility at the Organization, DS, GS and Depot levels. The maintenance flows evaluated are charged against the test and repair times to compute manhours. Work demands at DS, GS and Depot also include the demand incurred due to modification work orders.

1. AEY=SAVEEB*(TUFEC*TC+TUFTE*TE)
AERY=SAVEEB*(TUFTE*TER)
These statements compute the test and repair manhours for the LRUs at one equipment maintenance location, respectively. TUFEC and TUFTE are the LRUs per hour through the checkout and repair facilities. TC, TE, and TER are the checkout, test, and repair times. SAVEEB converts to the effective number of installations supported by one organizational maintenance facility.

2. AOY=(SAVEOB*((TUFOC*TC)+(TUFTO*TF)+(TMFO*TMO)))
+((QUA/D(OD))*2.*TMWO*TOMW)
AORY=(SAVEOB*((TUFTO*TFR)+(TMFO*TMOR)))+
((QUA/D(OD))*TMWO*TMOD)

These statements compute test (AOY) and repair (AORY) workloads (manhours) at Direct Support. Maintenance flows for LRUs (TUFOC, TUFTO) and modules (TMFO) are charged against their respective test (TC, TF, TMO) and repair (TFR, TMOR) times to compute the number of maintenance manhours. The workloads also include the test and repair of modification kits. The workload is double charged for testing mod kits; once before installation and once afterwards. The mod kit workloads are applied to each of the initial LRU buys (QUA). SAVEOB is used to convert the workloads to the effective number of equipment installations supported by one DS maintenance facility.

3. AIY=(SAVEIB*((TUFIC*TC)+(TUFTI*TI)+(TMFI*TMI)))+
((QUA/D(DI))*2.*TMWI*TIMW)
AIRY=(SAVEIB*((TUFTI*TIR)+(TMFI*TMIR)))+
((QUA/D(DI))*TMWI*TMID)

These statements compute test (AIY) and repair (AIRY) workloads (manhours) at General Support. Maintenance flows for LRUs (TUFIC, TUFTI) and modules (TMFI) are charged against their respective test (TC, TI, TMI) and repair (TIR, TMIR) times to compute the number of maintenance manhours. The mod kit workloads are evaluated similar to the description for statement 2 above. SAVEIB converts the workloads to the effective number of equipment installations supported by one GS maintenance facility.

4. ADY=(SAVEDB*((TUFDC*TC)
+(TUFTD*TD)+(TMFD*TMD)))+
((QUA/D(DD))*2.*TMWD*TDMW)
ADRY=(SAVEDB*((TUFTD*TDR)+(TMFD*TMDR)))+
((QUA/D(DD))*TMWD*TMDD)

These statements compute test (ADY) and repair (AORY) workloads (manhours) at the Depot. Maintenance flows for LRUs (TUFDC, TUFTD) and modules (TMFD) are charged against their respective test (TC, TD, TMD) and repair (TDR, TMDR) times to compute the number of maintenance manhours. The workload for mod kits is evaluated similar to what is described in statement 2 above. SAVEDB converts the workloads to the effective number of equipment installations supported by one Depot maintenance facility.

E. This section of TEMWF accumulates the manhours for individual LRU cases whose workloads are to be shared and computes demand for manpower and test equipment. The accumulation of workloads is controlled by the flags (ETI, ETII) for posting work demands. Workloads are accumulated until a posting flag is turned on (Unity). When a posting flag is unity, demand for manpower and

test equipment is computed using the accumulated manhours. The manhour accumulations are reset to zero after returning to the main program (LOGAM). There are two posting flags used in this section. ETI posts demands for type I test equipment and ETII is for type II. To post out demands the value of ETI and ETII are input as unity. Otherwise they must have the value zero.

Expected value flags are used to either compute expected manpower and test equipment demands or to round-off the demands to the nearest integer. ZFL is the input parameter used for round-off. Manpower and test equipment demands are computed at each maintenance echelon by dividing the accumulated manhours with the number of hours in a week for test men and test equipment. The demands for manpower and test equipment derived from the manhour accumulations are multiplied by the number of maintenance facilities at an echelon to get total demands.

1. SAEY=SAEY+AEY
   This statement accumulates test manhour demands for the effective number of installations supported by one organizational maintenance facility. SAEY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

2. ESUY=ETI*SAEY*(1.+FI)
   This statement adds the type I test equipment manhour demand for self-support to the LRU test demand (SAEY) at the organizational level.

3. ESU=(EVET*ED*168.*(ESUY/D(WE))+((1.-EVET)*ED
   AINT((168.(ESUY/D(WE)))+ZFL))
   This statement computes the expected value demand for type I test equipment if the expected value flag EVET=1. Otherwise EVET=0 where the demand is computed using integer round-off. Accumulated manhours (ESUY) are divided by the test equipment workweek (WE) and multiplied by the number of maintenance facilities (ED) to get total demand at the organizational level.

4. ESUM=(EVEM*ED*168.*(ESUY/D(WEM)))+((1.-EVEM)*
   ED*AINT((168.*ESUY/D(WEM)))+ZFL)
   This statement computes the expected value demand of test manpower using type I test equipment if the expected value flag EVEM=1. Otherwise EVEM=0 where the demand is computed using integer round-off. Accumulated manhours (ESUY) are divided by the test crew workweek (WEM) and multiplied by the number of maintenance facilities (ED) to get total demand at the organizational level.

5. SAERY=SAERY+AERY
   This statement accumulates repair manhour demands for

This statement accumulates repair manhour demands for the effective number of installations supported by one organizational maintenance facility. SAERY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

6. ESURY=ETI*SAERY
This statement controls posting out the accumulated repair work demand for type I test equipment at organizational level service channels. ETI=1 is input for posting, otherwise, ETI=0 is input.

7. ESUR=(EVER*ED*168.*(ESURY/D(WER)))+((1.-EVER)*
ED*AINT((168.*(ESURY/D(WER)))+ZFL)
This statement computes the expected value demand of repair manpower if the expected value flag EVER=1. Otherwise EVER=0 where the demand is computed using integer round-off. Accumulated manhours (ESURY) are divided by the repair crew workweek (WER) and multiplied by the number of maintenance facilities (ED) to get total demand at the organizational level.

8. SAOY=SAOY+AOY
This statement accumulates test manpower demands for the effective number of installations supported by one direct support maintenance facility. SAOY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

9. DSUY=ETI*SAOY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at the Direct Support facility. To post this work demand, ETI=1 is input, otherwise, ETI=0 is input.

10. DSU=(EVOT*OD*168.*(DSUY/D(WO)))+((1.-EVOT)*OD*
AINT((168.*(DSUY/D(WO)))+ZFL))
This statement computes the expected value demand for Type I test equipment if the expected value flag EVOT=1. Otherwise EVOT=0 where the demand is computed using integer round-off. Accumulated manhours (DSUY) are divided by the test equipment workweek (WO) and multiplied by the number of maintenance facilities (OD) to get total demand at the Direct Support level.

11. DSUM=(EVOM*OD*168.*(DSUY/D(WOM)))+((1.-EVOM)*OD*
AINT((168.*(DSUY/D(WOM)))+ZFL))
This statement computes the expected value demand of test manpower using Type I test equipment if the expected value flag EVOM=1. Otherwise EVOM=0 where the demand is computed using integer round-off. Accumulated manhours (DSUY) are divided by the test

maintenance facilities (OD) to get total demand at the Direct Support level.

12. SAORY=SAORY+AORY
This statement accumulates repair manhours for the effective number of installations supported by one Direct Support maintenance facility. SAORY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

13. DSURY=ETI*SAORY
This statement controls posting out of the accumulated repair work demand for Type I test equipment at the Direct Support service channel. ETI=1 is input for posting, otherwise, ETI=0 is input.

14. DSUR=(EVOR*OD*168.*(DSURY/D(WOR)))+((1.-EVOR)
*OD*AINT((168.*(DSURY/D(WOR)))+ZFL))
This statement computes the expected value demand of repair manpower if the expected value flag EVOR=1. Otherwise EVOR=0 where the demand is computed using integer round-off. Accumulated manhours (DSURY) are divided by the repair crew workweek (WOR) and multiplied by the number of maintenance facilities (OD) to get total demand at the Direct Support level.

15. SAIY=SAIY+AIY
SAIRY=SAIRY+AIRY
These statements accumulate test and repair manhour demands for the effective number of installations supported by one General Support maintenance facility. SAIY and SAIRY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

16. GSUY=ETI*SAIY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at the General Support facility. To post this work demand, ETI=1 is input, otherwise, ETI=0 is input.

17. GSU=(EVIT*DI*168.*(GSUY/D(WI)))+((1.-EVIT)*DI*
AINT((168.*(GSUY/D(WI)))+ZFL))
This statement computes the expected value demand for Type I test equipment if the expected value flag EVIT=1. Otherwise EVIT=0 where the demand is computed using integer round-off. Accumulated manhours (GSUY) are divided by the test crew workweek (WI) and multiplied by the number of maintenalce facilities (DI) to get total demand at the General Support level.

18. GSUM=(EVIM*DI*168.*(GSUY/D(WIM)))+((1.-EVIM)*DI*
AINT((168.*(GSUY/D(WIM)))+ZFL))

This statement computes the expected value demand of test manpower using Type I test equipment if the expected value flag EVIM=1. Otherwise EVIM=0 where the demand is computed using integer round-off. Accumulated manhours (GSUY) are divided by the test crew workweek (WIM) and multiplied by the number of maintenance facilities (DI) to get total demand at the General Support level.

19. GSURY=ETI*SAIRY
This statement controls posting out the accumulated repair work demand for Type I test equipment at the General Support service channel. ETI=1 is input for posting, otherwise ETI=0 is input.

20. GSUR=(EVIR*DI*168.*(GSURY/D(WIR)))+((1.-EVIR)
*DI*AINT((168.*(GSURY/D(WIR)))+ZFL))
This statement computes the expected value demand for repair manpower if the expected value flag EVIR=1. Otherwise EVIR=0 where the demand is computed using integer round-off. Accumulated manhours (GSURY) are divided by the repair crew workweek (WIR) and multiplied by the number of maintenance facilities (DI) to get total demand at the General Support level.

21. SADY=SADY+ADY
SADRY=SADRY+ADRY
These statements accumulate test and repair manhour demand for the effective number of installations supported by one Depot maintenance facility. SADY and SADRY will be reinitialized to zero in the main program (LOGAM) after each LRU case where ETI=1.

22. AAIE=0.
IF(JTED.EQ.0.OR.JTED.EQ.2)AAIE=1.
These statements set the value for a qualifier (AAIE) that determines the type and location of test equipment. IF JTED=1 Type I test equipment is permitted at the Direct Support, General Support, and Depot sites. IF JTED=2 Type II test equipment is located at the Depot instead of Type I.

23. DEPY=ETI*(1.-AAIE)*SADY*(1.+FI)
This statement adds the Type I test equipment manhour demand for self-support to the LRU test demand at one Depot site. DEPY will have a value only when JTED=1.

24. DEP=(EVDT*168.*DD*(DEPY/D(WD)))+((1.-EVDT)*DD*
AINT((168.*(DEPY/D(WD)))+ZFL))
This statement computes the expected value demand for Type I test equipment if the expected value flag EVDT=1. Otherwise EVDT=0 where the demand is computed

using integer round-off. Accumulated manhours (DEPY) are divided by the test workweek (WD) and multiplied by the number of maintenance facilities (DD) to get total demand at the Depot level.

25. DEPM=(EVDM*168.*DD*(DEPY/D(WDM)))+((1.-EVDM)
*DD*AINT((168.*(DEPY/D(WDM)))+ZFL))
This statement computes the expected value demand of test manpower using Type I test equipment if the expected value flag EVDM=1. Otherwise EVDM=0 where the demand is computed using integer round-off. Accumulated manhours (DEPY) are divided by the test crew workweek (WDM) and multiplied by the number of maintenance facilities (DD) to get the total demand at the Depot level.

26. DEPRY=ETI*(1.-AAIE)*SADRY
This statement controls posting out the accumulated repair work demand for Type I test equipment at one Depot service channel. DEPRY will have a manpower value when JTED=1.

27. DEPR=(EVDR*168.*DD*(DEPRY/D(WDR)))+((1.-EVDR)
*DD*AINT((168.*(DEPRY/D(WDR)))+ZFL))
This statement computes the expected value demand for repair manpower if the expected value flag EVDR=1. Otherwise EVDR=0 where the demand is computed using integer round-off. Accumulated manhours (DEPRY) are divided by the repair crew workweek (WDR) and multiplied by the number of maintenance facilities (DD) to get total demand at the Depot level.

28. DEPAIY=ETII*AAIE*SADY*(1.+FII)
This statement adds the Type II test equipment manhours for self-support to the test demand at one Depot service channel. DEPAIY will have a value only if the input flags ETII and JTED have the values 1 and 2, respectively.

29. DEPAIE=(EVDT*168.*DD*(DEPAIY/D(WD)))+((1.-EVDT)
*DD*AINT((168.*(DEPAIY/D(WD)))+ZFL))
This statement computes the expected value demand for Type II test equipment if the expected value flag EVDT=1. Otherwise EVDT=0 where the demand is computed using integer round-off. Accumulated manhours (DEPAIY) are divided by the test equipment workweek (WD) and multiplied by the number of maintenance facilities (DD) to get total demand at the Depot level.

30. DEPAIM=(EVDM*168.*DD*(DEPAIY/D(WDM)))+((1.-EVDM)
*DD*AINT((168.*(DEPAIY/D(WDM)))+ZFL))
This statement computes the expected value demand of

test manpower using Type II test equipment if the expected value flag EVDM=1. Otherwise EVDM=0 where the demand is computed using integer round-off. Accumulated manhours (DEPAIY) are divided by the test crew workweek (WDM) and multiplied by the number of maintenance facilities (DD) to get total demand at the Depot level.

31. DEPARY=ETII*AAIE*SADRY
This statement controls posting out the accumulated repair work demand for Type II test equipment at the Depot. When JTED is input as two, AAIE is set to unity for posting Type II test equipment instead of Type I. ETII=1 is input for posting Type II test equipment demands, otherwise ETII=0 is input.

32. DEPAR=(EVDR*168.*DD*(DEPARY/D(WDR)))+((1.-EVDR)
*DD*AINT((168.*(DEPARY/D(WDR)))+ZFL))
This statement computes the expected value demand of repair manpower using Type II test equipment if the expected value flag EVDR=1. Otherwise EVDR=0 where the demand is computed using integer round-off. Accumulated manhours (DEPARY) are divided by the repair crew workweek (WDR) and multiplied by the number of maintenance facilities (DD) to get total demand at the Depot level.

F. This section of TEMWF accumulates the test and repair manhour demands for the effective number of installations supported by one facility at a maintenance level. These accumulations are made on the individual LRU cases that are input as a case concept. A case concept is defined as all of the individual LRU cases input until the flag IS=1 is input or NU<-1. This flag will cause reinitialization of the accumulations to zero. The accumulations are the expected workloads as shared among all the LRU cases in a concept. These workload factors are only used for outputs as a means of comparison with accumulations using the posting logic described in section E above.

1. CAEY=CAEY+AEY
CAERY=CAERY+AERY
These statements accumulate test and repair manhour demands for the effective number of installations supported by one Organizational maintenance facility. These accumulations are reset to zero in the main program when IS=1.

2. CAOY=CAOY+AOY
CAORY=CAORY+AORY
These statements accumulate test and repair manhour demands for the effective number of installations

supported by one Direct Support maintenance facility. These accumulations are reset to zero in the main program when IS=1.

3. CAIY=CAIY+AIY
CAIRY=CAIRY+AIRY
These statements accumulate test and repair manhour demands for the effective number of installations supported by one General Support maintenance facility. These accumulations are reset to zero in the main program when IS=1.

4. CADY=CADY+ADY
CADRY=CADRY+ADRY
These statements accumulate test and repair manhour demands for the effective number of installations supported by one Depot maintenance facility. These accumulations are reset to zero in the main program when IS=1.

G. This section of TEMWF converts maintenance test and repair manhours to the demand for manpower per day. Accumulations are made for the individual LRU cases in a concept. The HPD array is reinitialized to zero in the main program (LOGAM) when the last case in a concept is indicated by the input IS=1.

1. HPD(1,1)=HPD(1,1)+24.*(1.-RF)*(1.+FE)*AME*TENMAN
HPD(1,2)=HPD(1,2)+(24.*RF*AME+SMF*24.*EE)*TENMAN
These statements accumulate manhours per day required to fault isolate, test, remove, and replace LRUs at one Equipment installation. HPD (1,1), test manpower, is computed as a fraction (1-RF) of TRC that is used to checkout and test. The demand for self-support (FE) of type V test equipment is added to the test manpower. The demand for manpower to remove and replace LRUs is determined by the remove and replace fraction (RF) of TRC and the scheduled maintenance factor (SMF) for a material system (EE). The demands are modified by TENMAN, a manpower productivity factor.

2. HPD(2,1)=HPD(2,1)+24.*(1.+FI)*AEY
HPD(2,2)=HPD(2,2)+24.*AERY
These statements accumulate manhours per day to test and repair LRUs for the effective number of installations supported by one Organizational maintenance facility. The demand for Type I test equipment self-support (FI) is added to the test demand.

3. HPD(3,1)=HPD(3,1)+24.*(1.+FI)*AOY
HPD(3,2)=HPD(3,2)+24.*AORY
These statements accumulate manhours per day to test

and repair LRUs for the effective number of installations supported by one Direct Support maintenance facility. The demand for Type I test equipment self-support (FI) is added to the test demand.

4. HPD(4,1)=HPD(4,1)+24.*(1.+FI)*AIY
HPD(4,2)=HPD(4,2)+24.*AIRY
These statements accumulate manhours per day to test and repair LRUs for the effective number of installations supported by one General Support maintenance facility. The demand for Type I test equipment self-support (FI) is added to the test demand.

5. HPD(5,1)=HPD(5,1)+24.*((1.+FI)*(1.-AAIE)+AAIE*
(1.+FII))*ADY
HPD(5,2)=HPD(5,2)+24.*ADRY
These statements accumulate manhours per day to test and repair LRUs for the effective number of installations supported by one Depot maintenance facility. The demand for either Type I (FI) or Type II (FII) self-support is added to the manpower demands. When JTED is input as unity, AAIE=0 and Type I equipment is used. JTED=2 will set AAIE=1 where Type II equipment will be used.

6. RETURN
END
These statements terminate execution of TEMWF and returns control to the main program (LOGAM).

4.6.2 SUBROUTINE PERSON

This subroutine computes the personnel requirements for test, checkout and repair of equipment at four echelons of repair (ED,DS,GS,Depot). The accumulated manhours (HPD) from Subroutine TEMWF are used to evaluate the personnel requirements. PERSON is called from the main program once a case total is completed. Case totals are evaluated when the user inputs NU=-2 or -3 with the last individual LRU case of a case total concept. The HPD array will be accumulated in TEMWF until these values for NU are input.

The personnel requirements output is also printed from this subroutine. Once the outputs have been printed a return is made to LOGAM where the HPD array will be initialized to all zeros for the accumulation of any additional cases.

A. This section of PERSON contains the assignment of program variables to computer memory.

1. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains the tape and/or disc files used in the program. K004FX, the output file, is the only file used in this subroutine.

2. COMMON/PERSN/PERS(5,2),PERL(5,2),HPD(5,2),
   WPD(5,2),WPDL(5,2)
   This common block contains the assignment of personnel requirements. All of the variables are evaluated in this subroutine except for HPD which is the manhour accumulations from Subroutine TEMWF.

3. COMMON/INPUT/
   This common block contains the list of input variables for NAMELIST/L/. Refer to section A of LOGAM for a listing of the variable names.

4. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),
   ZU(4),SENSY(266),T(2000)
   This statment contains the list of all the dimensioned variables in common block INPUT.

5. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
   OR,AYZIS,AYZOS
   This common block contains a list of variables that are computed and used in various subroutines of this program. There is only one of these variables (AAIE) that is used in this subroutine. AAIE is set in subroutine TEMWF when Type II test equipment is to be used at the Depot maintenance facilities.

B. This section of PERSON computes the personnel requirements. The outputs from this section are the number of test and repair men per week for all maintenance locations (PERS), the number of men per week per maintenance location (PERL), the number of manhours per year per maintenance location (WPDL), and the number of manhours per year at all maintenance locations (WPD).

1. PERS(1,1)=HPD(1,1)*7./D(WMR)
   PERS(1,2)=HPD(1,2)*7./D(WMR)
   PERS(1,1)=ED*(ETEI*PERS(1,1)+(1.-ETEI)*AINT
   (PERS(1,1)+ZFL))
   PERS(1,2)=ED*(EREI*PERS(1,2)+(1.-EREI)*AINT
   (PERS(1,2)+ZFL))
   These statements compute case totals for the number of crews per week that is required to fault isolate and test (PERS(1,1)) and to remove and replace (PERS(1,2)) LRUs at all equipment installations. If the expected value flag ETEI=1, the expected value is computed for PERS; otherwise the value is rounded according to the input value for ZFL. WMR is the number of hours in a work week for repair crews.

2. PERS(2,1)=HPD(2,1)*7./D(WEM)
   PERS(2,2)=HPD(2,2)*7./D(WER)
   These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs at one organizational maintenance location. WEM and WER are the scheduled work weeks in hours for test and repair equipment manpower at the organizational level.

3. PERS(3,1)=HPD(3,1)*7./D(WOM)
   PERS(3,2)=HPD(3,2)*7./D(WOR)
   These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs and modules at one Direct Support maintenance location. WOM and WOR are the scheduled work week in hours for test and repair equipment manpower at the Direct Support level.

4. PERS(4,1)=HPD(4,1)*7./D(WIM)
   PERS(4,2)=HPD(4,2)*7./D(WIR)
   These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs and modules at one General Support maintenance location. WIM and WIR are the scheduled work week in hours for test and repair equipment manpower at the General Support level.

5. PERS(5,1)=HPD(5,1)*7./D(WDM)
   PERS(5,2)=HPD(5,2)*7./D(WDR)

These statements compute case totals for the number of crews per week that are required to test and checkout, and to repair, respectively, the LRUs and modules at one Depot maintenance location. WDM and WDR are the scheduled work week in hours for test and repair equipment manpower at the Depot level.

6.
```
PERS(2,1)=TEMAN*ED*(EVEM*PERS(2,1)+(1.-EVEM)*
          AINT(PERS(2,1)+ZFL))
PERS(2,2)=TERMAN*ED*(EVER*PERS(2,2)+(1.-EVER)*
          AINT(PERS(2,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TEMAN, TERMAN) and by the number of maintenance locations (ED) to get the total number of men per week for test and repair, respectively, at all organizational levels. The expected value flags (EVEM,EVER) are used to get either the computed value or an integer value that is rounded according to the value of ZFL.

7.
```
PERS(3,1)=TDMAN*OD*(EVOM*PERS(3,1)+(1.-EVOM)*
          AINT(PERS(3,1)+ZFL))
PERS(3,2)=TDRMAN*OD*(EVOR*PERS(3,2)+(1.-EVOR)*
          AINT(PERS(3,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TDMAN, TDRMAN) and the by the number of Direct Support locations (OD) to get the number of men per week for test and repair, respectively, at all Direct Support locations. The expected value is an integer value that is rounded according to the value of ZFL.

8.
```
PERS(4,1)=TGMAN*DI*(EVIM*PERS(4,1)+(1.-EVIM)*
          AINT(PERS(4,1)+ZFL))
PERS(4,2)=TGRMAN*DI*(EVIR*PERS(4,2)+(1.-EVIR)*
          AINT(PERS(4,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TGMAN, TGRMAN) and by the number of General Support locations (DI) to get the number of men per week for test and repair, respectively, at all General Support locations. The expected value flags (EVIM, EVIR) are used to get either the computed value or an integer value that is rounded according to the value of ZFL.

9.
```
PERS(5,1)=DD*((1.-AAIE)*TDPMI+AAIE*TDPMII)*(EVDM*
          PERS(5,1)+(1.-EVDM)*AINT(PERS(5,1)+ZFL))
PERS(5,2)=DD*((1.-AAIE)*TDPRI+AAIE*TDPRII)*(EVDR*
          PERS(5,2)+(1.-EVDR)*AINT(PERS(5,2)+ZFL))
```
These statements multiply the number of crews by the number of men per crew (TDPMI, TDPRI) and by the number of Depot locations (DD) to get the number of men per

week for test and repair, respectively, at all Depot locations. The expected value flags (EVDM, EVDR) are used to get either the computed value or an integer value that is rounded according to ZFL. The value of AAIE determines if the test equipment is of Type I or Type II. The number of men per crew will be TDPMII and TDPRII for Type II equipment.

10.
```
WPD(1,1)=HPD(1,1)*ED
WPD(1,2)=HPD(1,2)*ED
```
These statements compute case totals for the manhours per day to fault isolate and test, and to remove and replace, respectively, the LRUs at all equipment installations.

11.
```
WPD(2,1)=HPD(2,1)*ED
WPD(2,2)=HPD(2,2)*ED
WPD(3,1)=HPD(3,1)*OD
WPD(3,2)=HPD(3,2)*OD
WPD(4,1)=HPD(4,1)*DI
WPD(4,2)=HPD(4,2)*DI
WPD(5,1)=HPD(5,1)*DD
WPD(5,2)=HPD(5,2)*DD
```
These statements compute case totals for the manhours per day to test and repair the LRUs at all Organizational, Direct Support, General Support, and Depot maintenance facilities, respectively.

12.
```
      DO 5058 I1=1,5
      DO 5058 I2=1,2
      WPD(I1,I2)=365.25*WPD(I1,I2)
      IF((I1.LE.2).AND.(I2.LE.2))GO TO 5059
      GO TO 5060
5059  WPDL(I1,I2)=WPD(I1,I2)/EDS
      PERL(I1,I2)=PERS(I1,I2)/EDS
      GO TO 5058
5060  IF(I1.EQ.3.AND.(I2.EQ.1.OR.I2.EQ.2)) GO TO 5061
     -  GO TO 5062
5061  WPDL(I1,I2)=WPD(I1,I2)/OD
      PERL(I1,I2)=PERS(I1,I2)/OD
      GO TO 5058
5062  IF(I1.EQ.4.AND.(I2.EQ.1.OR.I2.EQ.2)) GO TO 5063
      GO TO 5064
5063  WPDL(I1,I2)=WPD(I1,I2)/DI
      PERL(I1,I2)=PERS(I1,I2)/DI
      GO TO 5058
5064  IF(I1.EQ.5.AND.(I2.EQ.1.OR.I2.EQ.2)) GO TO 5065
      GO TO 5058
5065  WPDL(I1,I2)=WPD(I1,I2)/DD
      PERL(I1,I2)=PERS(I1,I2)/DD
5058  CONTINUE
```
The statements in this nested Do Loop compute total

manhours per year (WPD), manhours per year per installation (WPDL), and the number of men per week per installation (PERL) required to checkout, replace, test, and repair equipment. The information computed here is used for case total outputs.

C. This section of PERSON prints the personnel requirements output. A case total for personnel will be printed each time a reference is made to PERSON from the main program.

1. CALL PAGE
   This statement calls Subroutine PAGE to print header information at the top of the page.

2. WRITE (K004FX,4000)
   This statement prints a test and repair manpower title for the personnel outputs.

3. WRITE (K004FX,4001) WPD,PERS
   This statement prints the manhours per year (WPD) and the number of men per week (PERS) required at all maintenance locations.

4. WRITE (K004FX,5067)WPDL, PERL
   This statement prints the manhours per year (WPDL) and the men per week (PERL) required at one maintenance location of an echelon.

5. RETURN
   END
   This statement returns logic back to the main program.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

4.7 SUBROUTINE LGCST

This subroutine is called from the main program (LOGAM) to evaluate the costs for individual LRU cases and to accumulate individual LRU costs for a case total. Case totals are accumulated until the user inputs IS=1 or NU<-1 in NAMELIST/L/ with the last LRU case of a case total. After setting IS=1, the cost accumulators are reinitialized to zero in the main program and a new accumulation will begin with the next LRU case.

The outputs from this subroutine are transmitted to the print subroutines through common block COSTS and ZERO. The individual LRU costs are contained in COMMON/COSTS/ and are printed in Subroutine IDVOP. There are 2 case total accumulations in this subroutine. Case totals that are printed in Subroutine TTLOP using a LOGAM output format are transmitted with common block ZERO. Case totals that are printed in Subroutine EIGHT using the DA PAM 11-4 format are accumulated into the SDA array and transmitted with common block DAPAM.

The description of the code in this subroutine is organized into functional groups. The sections containing these groupings are the following:

SECTION A - Computer memory assignments;
SECTION B - Shipping pipeline cost factors;
SECTION C - Development and Procurement;
SECTION D - Cost of materials;
SECTION E - Personnel;
SECTION F - Salvage values;
SECTION G - Supply Administration and Shipping;
SECTION H - Individual LRU totals;
SECTION I - Lifecycle phase totals;
SECTION J - Cost growth and discount;
SECTION K - Case total accumulations.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

A. This section of LGCST contains the list of common blocks and other assignment of variables to computer memory.

1. COMMON/INPUT/
This common block contains the list of input variables of NAMELIST/L/. The order of the variables in this list must correspond to the order in the DATA statement of Subroutine FINLE. Refer to section A of LOGAM for the listing of these parameters.

2. DIMENSION G(20,H(4),OL(4),OST(4),SL(4),TAT(4),TAYZ(10),
ZM(4),ZP(3),ZU(4),SENSY(266),T(2000)
This statement contains all of the dimensioned variables of COMMON/INPUT/.

3. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
This statement contains the list of all tape and/or disc files used by LOGAM. The assignments for these files are made in block data routine BLKDAT. K004FX is the output file and is used in this subroutine to print BUG, the debugging NAMELIST.

4. COMMON/POUT/POUT(13),AMULT
This contains the output array for printing DA PAM 11-4 costs in Subroutine EIGHT and the cost unit multiplier. AMULT is the only element of this common block used in LGCST.

5. COMMON/MMHWF/
This common block contains the list of maintenance work factors that were evaluated in Subroutine TEMWF. Refer to section A of LOGAM for a list of these parameters.

6. COMMON/COSTS/CET,CFT,CIVT,CMPT,CQTE,CQTO,CQTI,CQTD,
CQTME,CQTMO,CQTMI,CQTMD,CQTPO,CQTPI,CQTPD,
CQTT,CQTMT,CQTPT,CRUT,CRMT,CRPT,CROT,CSAT,
CSHT,CTST,CWHT,DELTA,ECMPT,EPVGCT,GCT,
PDELTA,PVGCT
This common block contains the list of individual LRU costs that are evaluated in LGCST and printed in Subroutine IDVOP. A description for each of these variables can be found in the following sections of this subroutine.

7. COMMON/REPSP/
This common block contains a list of the replenishment spares, consumed quantities, buy lots, reorder actions and residuals. These values are evaluated in LOGAM by calling Subroutine REORDR once each for LRUs, modules and parts. Refer to section A of LOGAM for a listing of these variables.

8. COMMON/PROVQ/
This common block contains the list of initial provisioning quantities. These parameters are evaluated in BASIC, IOL, MPROV or are input. The type of provisioning rule requested determines where these values are computed. Refer to section A of LOGAM for a listing of these variables.

9. COMMON/BAS/
This common block contains a list of either supply pipeline quantities or shipping factors, depending on how Subroutine BASIC is referenced. In this Subroutine, BASIC is called with shipping factors to compute shipping costs. Refer to section A of LOGAM for a listing of these variables.

10. COMMON/ZERO/CCET,CCTS,CCTSR,CCF,CCM,CCMF,CCMD,CCMFD,
CTRF,CTRDEP,CTR,CIV,CIVREC,CRT,CWH,CSA,CSAREC,
CSH,CGT,CTREC,PCD,CQTU,PCP,CQTM,PCR,CQTP,PCS,
CQTUMP,PCGT,SEMPT,SEPC,SPCR,SEPV,SDEL,SPDEL,
CAYZ(10),CAYZI(10)
This common block contains the list of cost variables that are accumulatd in this subroutine as case totals and printed in Subroutine TTLOP using the LOGAM output format. CAYZ and CAYZI are availabilty accumulators from the main program and are not used in this subroutine. These variables are reinitialized in the main program after the user inputs IS=1 or NU<-1. Each of the cost parameters are discussed in the following sections.

11. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
OR,AYZIS,AYZOS
Of this miscellaneous list of variables only the first five and FMWO are used in this subroutine. AAIE is a factor (1 or 0) that determines if Type I or Type II test equipment is used at the Depot. EDEE and QUA are the total installed equipments and initial production equipments, respectively. YR8 is the number of operating hours in O&M phase and SAVE converts values in clock time to real time operating hours. FMWO is the frequency of modification work orders. AAIE and FMWO are set in Subroutine TEMWF, the other parameters are either set in the main program (LOGAM) or Subroutine SUMRY (STI, STII).

12. COMMON/DAPAM/SDA(16)
This common block contains the case total accumulator for costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. These case total accumulations are discussed in section K of this subroutine.

13. NAMELIST/BUG/
This statement contains a list of the parameters that are either used in this subroutine or are evaluated here. This list will be printed as a debugging output when the user inputs IBG=1 in NAMELIST/L/. The write statement for BUG is located at the end of this subroutine listing.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

B. This section of LGCST computes cost factors for the shipping and handling of LRUs, modules, parts and mod kits. Subroutine BASIC is called with the round trip shipping costs to compute the cost per pound for all pipeline flows. These shipping cost factors are used in section H of this subroutine to compute shipping costs in the field (CSHTF), Depot (CSHTD) and total (CSHT).

1. ONTIME=8766.*YR*SAVE
   This statement computes a factor that will convert the shipping factors from BASIC that are in clock time to real time operating factors over the O&M phase.

2. SHTEO=CDEO+CDOE
   SHTOI=CDOI+CDIO
   SHTID=CDID+CDDI
   These statements sum the round trip shipping costs in dollars per pound between Equipment/Direct Support, Direct Support/General Support, and General Support/Depot, respectively.

3. CALL BASIC
   (CDFD,CDFD,CDFD,CDFD,CDFD,CDFD,0.,CDOE,CDIO,
   CDDI,0.,0.,0.,0.,SHTEO,SHTOI,SHTID,0.)
   This statement calls Subroutine BASIC to compute the cost of shipping and handling of LRUs, modules and parts. CDFD is the shipping rate in dollars per item per trip between Factory and Depot. The other non zero arguments are described in the statement above.

4. SHU=QYF+(FMWO*(SHTEO+((1.-ZO)*SHTOI)
   +((1.ZO-ZI)*(SHTID*DPT))))
   SHM=QMO+QMI+QMD+QFME+QFMO+QMFI+QFMD+QME
   SHP=QPO+QPI+QPD
   These statements compute the shipping and handling cost factors in dollars per pound of all pipeline flow for LRUs, modules and parts, respectively.

5. SHMF=SHM-QMD-QFMD
   SHPF=SHP-QPD
   These statements compute shipping and handling cost factors with the exclusion of the Depot/factory pipe for modules and parts, respectively.

6. SHKIT=FMWO*(CDFD+(CDDI*(ZO+ZI))+(CDIO*ZO))
   This statement computes the total cost factor for modification Work Orders (MWOs).

7. SHKITF=FMWO*CDIO*ZO
   This statement computes the MWO shipping cost factors to Field and Depot, respectively.

8. SHUF=QYF+(FMWO*(SHTEO+((1.-ZO)*SHTOI)))
This statement computes a cost factor to ship LRUs to the Field.

C. This section of LGCST computes the development and procurement costs for installed equipments, provisioning and test equipments.

1. CED=CEND*REPEAT*AMULT
This statement computes the development cost for the LRU types (REPEAT). CEND is the cost to develop an LRU and will be set to zero in the main program after its use here. The value of CEND will be reinstated to its original value when a new case accumulation begins (IS=1).

2. CEP=(CUP*EDEE*SPE+CPE)*AMULT
This statement computes the procurement cost of prime equipment (installed LRUs). This cost includes both recurring and non-recurring where CPE is the non-recurring cost factor. SPE is the factor used to adjust for prior expenditures ("sunk" costs).

3. CTSD=AMULT*(CI+CII+CCAL+CCSP+CV)
This statement converts the costs to develop test equipment into units specified by AMULT.

4. CTSP=((ESU+DSU+GSU+DEP)*CPI+(DEPAIE*CPII)+(EACAL*
CALSET*CCALP)+(EACSP*CONTCT
*CCSPP)+(ESI*CPV))*AMULT
This statement computes the procurement costs for test equipment.

5. CTSOFT=AMULT*(P*CMODPG+(CLRUPG+CPUBII+CALPUB+
CTCPUB+CPUBV)*REPEAT)
This statement computes the cost of development and programming of software for automated test equipment.

6. CMPPY=((ETI*CTRI)+(ETII*CTRII)+(EACAL*CTRCAL)+
(EACSP*CTRSPT)+ETE*CTRV)*AMULT
This statement computes the non-recurring cost to set up training courses for all test equipment types.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

D. This section of LGCST computes the cost of materials. Initial spares, replenishment spares, test equipment and mod kits are included.

1. CQTE=CUP*QTE*AMULT
   CQTO=CUP*QTO*AMULT
   CQTI=CUP*QTI*AMULT
   CQTD=CUP*QTD*AMULT
   These statements compute the cost of initial LRU provisions at all Equipment, Direct Support, General Support, and Depot supply facilities, respectively. CUP is the unit cost per LRU and the QTs are the quantity of LRUs stocked at the supply points.

2. CQTME=CMP*QTME*AMULT
   CQTMO=CMP*QTMO*AMULT
   CQTMI=CMP*QTMI*AMULT
   CQTMD=CMP*QTMD*AMULT
   These statements compute the cost of modules stocked at all Equipment, Direct Support, General Support, and Depot supply facilities, respectively. CMP is the unit cost per module and QTMs are the quantity of modules stocked at the supply points.

3. CQTPO=CPP*QTPO*AMULT
   CQTPI=CPP*QTPI*AMULT
   CQTPD=CPP*QTPD*AMULT
   These statements compute the cost of parts stocked at all Direct Support, General Support, and Depot facilities, respectively. CPP is the unit cost per part and the QTPs are the quantity of parts stocked at the supply points.

4. CQTT=CQTE+CQTO+CQTI+CQTD
   CQTMT=CQTME+CQTMO+CQTMI+CQTMD
   CQTPT=CQTPO+CQTPI+CQTPD
   These statements sum the initial provisioning costs for LRUs, modules and parts from all supply echelons.

5. CIVP=SPEV*(CQTT+CQTMT+CQTPT)
   This statement computes the total cost of initial spares and repair parts (PAM 2.090). SPEV adjusts the cost for prior expenditures ("sunk" costs).

6. CIVPD=SPEV*(CQTD+CQTMD+CQTPD)
   This statement computes the cost of initial provisioning for LRUs (QTD), modules (QTMD) and parts (QTPD) at the Depot facilities. SPEV is the factor to adjust for prior expenditures ("sunk" costs).

7. CIVPF=CIVP-CIVPD
   This statement subtracts the cost of initial spares at

the Depot from the total provisioning cost to get the cost of initial spares and repair parts at the field facilities.

8. CKITF=YR8*FMWO*CKIT*AMULT
This statement computes the cost of a mod kit for one LRU over the life of the system.

9. CIVR=CKITF*QUA+SPEVR*(CUP*QC+CMP*QCM+
CPP*QCP)*AMULT
This statement computes the cost of reordered material during the Operation and Support phase. The costs include mod kits, LRUs (QC), modules (QCM), and parts (QCP). SPEVR adjusts for prior expenditures ("sunk" costs).

10. CIVRD=AMULT*((SPEVR*((CUP*QCD)
+(CMP*QCMD)+(CPP*QCPD))))
This statement computes the cost of LRUs, modules and parts that are reordered at the Depot supply points. The factor SPEVR is used to adjust for prior expenditures ("sunk" cost).

11. CIVRF=CIVR-CIVRD-CKITF*QUA
This statement computes the cost of LRUs, modules and parts that are reordered during lifecycle at the Field facilities. This cost is reduced by the cost for mod kits.

12. DMM=CKITF*QUA*(1.-SVR)
This statement computes the mod kit costs for all initial LRU buys with credit for the salvage value.

13. U20=((ESU+DSU+GSU)*CRI+EACAL*CALSET*CCALR+
EACSP*CONTCT*CCSPR+ESI*CRV)*YR*AMULT
This statement computes the cost of supporting test stations at the field facilities. This cost includes material cost (CRI) for Type I equipment, cost to support calibration sets (CCALR), contact support teams (CCSPR) and training programs for Type V test sets (CRV).

14. U21=(DEP*CRI+DEPAIE*CRII)*YR*AMULT
This statement computes the cost of material to support either Type I or Type II test stations at the Depot facilities.

E. This section of LGCST computes the costs for test and repair personnel training, test and repair crews, military pay and allowances and civilian maintenance labor.

1. U1=CDPMAN*YR*AMULT*(DEPM*TDPMI+DEPAIM*TDPMII)
This statement computes the cost of test personnel for either Type I or Type II test equipment at the Depot facilities during the Operation and Support phase.

2. U2=168.*YR*AMULT*(ETI*FI*((1.-AAIE)*TDPMI*
CDPMAN*SADY/D(WDM))+(ETII*FII*AAIE*
TDPMII*CDPMAN*SADY/D(WDM)))
This statement computes the cost of test personnel for self-support of either Type I or Type II test equipment at the Depot during the Operation and Support phase.

3. U3=YR*AMULT*(DEPR*CDPRMN*TDPRI+DEPAR*
CDPRMN*TDPRII)
This statement computes the cost of repair personnel at the Depot using either Type I or Type II test equipment during the Operation and Support phase.

4. CMPRT=168.*YR*AMULT*(ETE*FE*ED*TENMAN*CEMAN*
SAME/D(WMR)+ETI*FI*(ED*TEMAN*CEMAN
*SAEY/D(WEM)+OD*TDMAN*
CDMAN*SAOY/D(WOM)+DI*TGMAN*
CGMAN*SAIY/D(WIM)))+U2
This statement computes the cost of test personnel for self-support of test equipment during the Operation and Support phase. The posted test manpower demands for the expected values are used to evaluate this cost. SAME, SAEY, SAOY and SAIY are the expected manpower demands. U2 is the cost of self support for test equipment at the Depot.

5. TRNGF=AMULT*CTRA*ARA*YR*(TEMAN*ESUM+
TGMAN*GSUM+TDMAN*DSUM+EACAL*
TALMAN*CALSET+EACSP*TONMAN*
CONTCT+TENMAN*(1.-RF)*DSIM)
This statement computes the recurring cost to train field maintenance test personnel at Organization, General Support and Direct Support using Type I, Calibration, Contact, and Type V test equipments.

6. TRNGD=AMULT*CTRAD*ARAD*YR*(DEPM*TDPMI+DEPAIM*TDPMII)
This statement computes the recurring cost to train Depot maintenance test personnel using either Type I or Type II test equipment.

7. TRNG=TRNGF+TRNGD
This statement sums the cost of training test personnel at the field and Depot echelons.

8. CMANE=CUCE*SMF*YR*ED*EE*AMULT
This statement computes the cost of scheduled preventive maintenance for test crews at the Equipment level during the Operation and Support phase.

9. CMPR=YR*AMULT*((1.-RF)*DSIM*CEMAN*TENMAN+
ESUM*CEMAN*TEMAN+DSUM*CDMAN*TDMAN+
GSUM*CGMAN*TGMAN+EACAL*CALSET*CALMAN*
TALMAN+EACSP*CONTCT*CONMAN*TONMAN)+
U1+TRNG+CMANE-CMPRT
This statement computes the cost of test personnel for tests on prime equipment at all echelons. The cost (CMPRT) for test personnel to self-support test equipment is subtracted from this equation since the self-support factors are included in the manpower for DSIM, ESUM, DSUM, GSUM and DEPM. U1 is the test cost at the Depot where DEPM is a part of that evaluation. This cost includes the training (TRNG) of test crews and scheduled preventive maintenance (CMANE).

10. TRNGRF=AMULT*CTRA*ARA*YR*(ESUR*TERMAN+
DSUR*TDRMAN+GSUR*TGRMAN+
TENMAN*RF*DSIM)
This statement computes the recurring cost to train repair personnel at the field facilities during the Operation and Support phase.

11. TRNGRD=AMULT*CTRAD*ARAD*YR*(DEPR*TDPRI+DEPAR*TDPRII)
This statement computes the recurring cost to train Depot repair personnel.

12. TRNGR=TRNGRF+TRNGRD
This statement sums the cost of training repair personnel at the field and Depot echelons.

13. CMPRR=YR*AMULT*((DSUR*CDRMAN*TDRMAN)+(GSUR*
CGRMAN*TGRMAN)+(ESUR*TERMAN*CERMAN)+(RF*
DSIM*CEMAN*TENMAN))+U3+TRNGR
This statement computes the cost of repair personnel at all maintenance facilities, including the training cost (TRNGR), during the Operation and Support phase. U3 is the repair personnel cost at the Depot.

14. U12=TRNG+TRNGR
This statement computes the cost for training test and repair personnel during O&M.

15. DML=U1+U2+U3
This statement computes the total personnel maintenance costs at the Depot. U1 and U3 are the test and repair costs for prime equipment. U2 is the self-support

costs for test equipment.

16. CLS=FLM*(EACAL*CALSET*CALMAN*TALMAN+EACSP*CONTCT *CONMAN*TONMAN)*YR*AMULT

This statement computes the cost of civilian maintenance labor to support calibration/Type III and contact support/Type IV test equipment during the Operation and Support phase. This cost is already included in the test personnel cost (CMPR). When FLM is input through NAMELIST this cost is assigned as civilian labor and must be substracted from the military Pay and Allowances (PA) in the next statement. The range of FLM must be from 0 to 1.
A non zero factor will assign that portion of the cost as civilian labor.

17. PA=CMPR-U1+CMPRT-U2+CMPRR-U3-TRNG-TRNGR-CLS

This statement computes the pay and allowances for military personnel by subtracting depot, training, and civilian labor costs from the total maintenance costs.

F. This section of LGCST computes the end of program salvage values for installed equipments, consumed materials, residual materials and test equipment.

1. CEV=SVE*EDEE*CUP*AMULT
   This statement computes the salvage value for the installed LRUs (EDEE).

2. CRUT=CUP*RU*AMULT
   CRMT=CMP*RM*AMULT
   CRPT=CPP*RP*AMULT
   These statements evaluate the original cost for LRUs, modules and parts that are residuals (RU,RM,RP) at end of program. These values are used in the evaluation of salvage values.

3. CRUTD=CUP*RUD*AMULT
   CRMTD=CMP*RMD*AMULT
   CRPTD=CPP*RPD*AMULT
   These statements evaluate the original cost for LRUs, modules and parts that are residuals (RUD,RMD,RPD) at the Depot. These values are used in the evaluation of salvage values.

4. CSVR=-SVR*(CIVR-(CRUT+CRMT+CRPT))
   This statement computes the salvage value for total material consumed during the O&S phase. CIVR is the cost of total material reordered. The cost of residual material is subtracted from CIVR to get the cost of consumed material.

5. CSVRD=-SVR*(CIVRD-(CRUTD+CRMTD+CRPTD))
   This statement computes the salvage value for material consumed at the Depot. CIVRD is the cost of material reordered at the Depot. The cost of residual material at the Depot is subtracted from the reorder cost to get the cost of consumed material.

6. CSVRF=CSVR-CSVRD
   This statement computes the salvage value for material consumed at the field facilities by subtracting the Depot value from the total salvage value.

7. CIVV=-SVV*(CRUT+CRMT+CRPT)
   This statement computes the salvage value for total residual stock.

8. CIVVD=-SVV*(CRUTD+CRMTD+CRPTD)
   This statement computes the salvage value for residual stock at the Depot.

9. CIVVF=CIVV-CIVVD
This statement computes the salvage value for residual stock at the field facilities by subtracting the Depot value from the total residual salvage value.

10. CTSV=-SVT*CTSP
This statement multiplies the fraction of salvage cost with the test equipment procurement cost to get the salvage value of test equipment at the end of program.

G. This section of LGCST computes the cost of supply administration and shipping of material.

1. CFT=(CFTD*((DEP*FTI)+(DEPAIE*FTII)))*AMULT*YR*12
This statement computes the cost during the Operation and Support phase to house Type I or Type II test equipment at the Depot level.

2. CROT=SPEVR*AMULT*(CRU*RQN+CRM*RQMN+
CRP*RQPN)
This statement computes the transaction cost of reordering LRUs, modules, and parts. Prior expenditures ("sunk" costs) are accounted for with the SPEVR factor.

3. CWHT=AMULT*YR*12*((CSDSU*((CUBEU*QTO)+
(CUBEM*QTMO)+(CUBEP*QTPO))+(CSGSU*
((CUBEU*QTI)+(CUBEM*QTMI)+(CUBEP*
QTPI)))+(CSDEP*((CUBEU*QTD)+(CUBEM*
QTMD)+(CUBEP*QTPD)))+(CSESU*
((CUBEU*QTE)+(CUBEM*QTME))))
This statement computes cost of storage of LRUs and modules at Equipment DS, GS and Depot, and parts at DS, GS and Depot.

4. CSMULT=(REPEAT*AB(QT)+P*AB(QTM)
+PP*FNSP*AB(QTP))*AMULT
This statement computes a factor that will be used to compute the cost of entering and retaining line items in the supply system. This factor is the sum of the LRU types in a material system, the number of module types in the LRUs, and the number of non-standard part types in the LRUs.

5. CSAP=CSMULT*CEN
CSAR=CSMULT*CAD*YR
These statements compute the costs to enter a line item into the supply system and to retain the item in the supply system, respectively.

6. FSAC=(EDS*(AB(QTE)*REPEAT+AB(QTME)*P)+
ODS*(AB(QTO)*REPEAT+AB(QTMO)*P
+AB(QTPO)*FNSP*PP)+DIS*(AB(QTI)
*REPEAT+AB(QTMI)*P+AB(QTPI)*FNSP
*PP))*FSA*AMAXI(0.,YR-1.)*AMULT
This statement computes the cost of supply administration at the Field facilities per line item type. FSA is the cost factor per line item per supply point.

7. CSAR=CSAR+FSAC
CSAT=CSAP+CSAR

These statements compute the costs to retain and administer, and to enter and maintain line items in supply system, respectively.

8. U17=CROR+CWHT+CSAT
This statement computes the subtotal cost for reordering, storing, and administering a line item within the supply system.

9. CSHT=AMULT*(YR8*QUA*WTKIT*SHKIT+ONTIME
*(WU*SHU+WM*SHM+WP*SHP)+
CDIST*(WU*(QT+EDEE)+WM*QTM+WP*QTP))
This statement computes the total cost of shipping mod kits, LRUs, modules, and parts for stocked and deployed items.

10. CSHTF=AMULT*(YR8*(QUA-QTD)*WTKIT*SHKITF+ONTIME*
(WU*SHUF+WM*SHMF+WP*SHPF)+CDIST*
(WU*(QUA-QTD)+WM(QTM-QTMD)+WP*(QTP-QTPD)))
CSHTD=CSHT-CSHTF
These statements compute the cost of shipping mod kits, LRUs, modules and parts to the Field and Depot facilities, respectively.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

H. This section of LGCST computes individual LRU cost totals for installed equipments, test equipments, spares, civilain labor, military pay and allowances, and cost of maintenance personnel.

1. CET=CED+CEP+CEV
   This statement computes the subtotal cost for all installed LRUs which includes development, procurement, and salvage value.

2. CTSR=U20+U21+CMPRT
   This statement computes the total cost of materials and personnel to support test stations during the O&M phase. U20 is the cost of supporting test stations at the field facilities (includes material and personnel). U21 is the material cost at the Depot and CMPRT is the total personnel cost for self-support of test equipment.

3. CTST=CTSD+CTSP+CTSR+CTSOFT+CTSV
   This statement computes the cost associated with the development, procurement, maintenance, and programming of test equipment software (excluding salvage).

4. CIVT=CIVP+CIVR+CSVR+CIVV
   This statement computes the cost of initial provisions and reordered material after taking into account the salvage value (CSVR,CIVV).

5. REPSF=CIVRF+CSVRF+CIVVF+U20
   RFPSD=CIVRD+CSVRD+CIVVD+U21
   These statements compute the cost of replenishment spares after taking credit for the salvage value at the Field and Depot facilities, respectively.

6. CMPT=CMPPY+CMPR+CMPRR
   This statement computes the manpower cost for test (CMPR),and repair (CMPRR) personnel during the Operation and Support phase. This cost includes training of personnel and the non-recurring cost to set up training courses (CMPPY).

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

I. This section of LGCST computes the totals for lifecycle phases, end of life salvage value for material, and system total.

1. CD=CED+CTSD+CTSOFT
   This statement sums the development costs for prime equipment (CED), test equipment (CTSD) and test software (CTSOFT).

2. CP=CEP+CTSP+CIVP+CSAP+CMPPY
   This statement sums the procurement costs for prime equipment (CEP), test equipment (CTSP), spares and repair parts (CIVP), entering items in inventory (CSAP), and setting up maintenance training facilities (CMPPY).

3. CR=CTSR+CFT+CMPR+CMPRR+CIVR
   +CROT+CWHT+CSAR+CSHT+CSVR
   This statement sums the O&M costs for test station support (CTSR), housing of test equipment (CFT), test (CMPR) and repair (CMPRR) personnel, replenishment spares (CIVR), reorder transactions (CROT), storage of material (CWHT), retaining spares in storage (CSAR), shipping of material (CSHT) and consumed material salvage (CSVR).

4. CS=CEV+CTSV+CIVV
   This statement sums the salvage value for material at end of O&M for installed equipment (CEV), test equipment (CTSV) and residual inventory (CIVV).

5. GCT=CD+CP+CR+CS
   This statement computes the total lifecycle cost of the system.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

J. This section of LGCST computes present value costs for the lifecycle spending phases. Present value costs are either discounted or inflated dependent on the input value for FINT. A negative value of FINT discounts and a positive value inflates. Refer to the input descriptions in section 5 for the use of FINT.

1. PVCD=CD
PVCP=CP
PVCR=CR
PVCS=CS
These statements set the present value of each spending phase to the value just computed. The assumption is made that the computed values are the "constant dollar" values. Present value costs are evaluated when an annual interest rate (FINT) is input, otherwise present value is the same as the constant value. Present value computation begins with statement 6 below.

2.
```
DELTA=YR*AMULT*((CDMAN*TDMAN*(DSUM-
          (168.*OD*(DSUY/D(WOM))))+(CDRMAN
          *TDRMAN*(DSUR-(168.*OD*(DSURY/D
          (WOR))))+(CGMAN*TGMAN*(GSUM-(168.*
          DI*(GSUY/D(WIM))))+(CGRMAN*TGRMAN*
          (GSUR-(168.*DI*(GSURY/D(WIR))))+
          (CEMAN*TEMAN*(ESUM-(168.*ED*(ESUY/
          D(WEM))))+(CERMAN*TERMAN*(ESUR-(168.*
          ED*ESURY/D(WER)))+(DSIM-(168.*ED*ESME/
          D(WMR))))
```
This statement computes the difference between the expected value and the dedicated manpower costs. If dedicated manpower values (DSUM, DSUR, GSUM, GSUR, ESUM, ESUR, DSIM) are computed as expected value then DELTA will be zero; otherwise a cost penalty of dedicated manpower as opposed to shared manpower will exist.

3. ECMPT=CMPT-DELTA
EPCGT=GCT-DELTA
These statements compute expected value costs by subtracting the manpower cost penalty (DELTA) from the dedicated costs for maintenance manpower and system total, respectively.

4. PDELTA=DELTA
This statement sets the difference between expected and dedicated manpower costs (DELTA) as the present value manpower cost penalty (PDELTA).

5.
```
    IF(FINT)400,40,400
400 CONTINUE
```
These statements test the value of the input variable FINT for the presence of a yearly interest rate. FINT

is the net rate between discount rate and inflation rate. If FINT is non-zero then cost growth (yearly) factors are computed below; otherwise the logic to be executed will begin at FORTRAN statement 40.

6.
```
PART=1.+FINT
PART1=PART**YD
PART2=PART**YP
PART3=PART**(-YR)
PART4=PART**YZ
```
These statements compute cost growth factors for Annual Rate, Development phase, Production phase, O&M phase, and a Time Shifting phase, respectively. PART4 is used in time phasing the costs by shifting the point of reference at which present value is started. The zero point of YZ is the beginning of the O&S phase (YR).

7.
```
PVCD=(CD/D(YD))*((PART1-1)/FINT)*PART4*PART2
PVCP=(CP/D(YP))*((PART2-1.)/FINT)*PART4
FACTOR=((1.-PART3)/FINT*PART4
PVCR=(CR/D(YR))*FACTOR
PDELTA=(DELTA/D(YR))*FACTOR
PVCS=CS*PART3*PART4
```
These statements recompute the present value costs as a function of cost growth for Development (PVCD), Acquisition (PVCP), O&M (PVCR), present value DELTA (PDELTA), and end of life salvage credit (PVCS). FACTOR is an internal computation used in the determination of PVCR and PDELTA.

8.
```
40 PVGCT=PVCD+PVCP+PVCR+PVCS
   EPVGCT=PVGCT-PDELTA
```
These statements compute the costs for present and expected present value totals, respectively. When FINT is input as zero, transfer is made to statement 40 from the IF test in line item 6 above, the totals would not include cost growth.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (LGCST)

K. This section of LGCST accumulates individual costs from each LRU case to keep a running total of all LRUs evaluated in a case total. The outputs for the DA PAM 11-4 format are accumulated into the SDA array.

1. CCET=CCET+CET
CCTS=CCTS+CTST
CCF=CCF+CFT
CCM=CCM+CMPT
CIV=CIV+CIVT
CRT=CRT+CROT
CWH=CWH+CWHT
CSA=CSA+CSAT
CSH=CSH+CSHT
CGT=CGT+GCT
These statements accumulate the costs for development of prime equipment (CET), development of test equipment (CTST), housing of test equipment at Depot (CFT), manpower to operate test equipment (CMPT), acquisition of prime equipment (CIVT), reordering of prime equipment (CROT), cost of storage for prime equipment (CWHT), entering and maintaining line items in the supply system (CSAT), shipping of prime equipment and mod kits (CSHT), and case total (GCT).

2. SDA(1)=SDA(1)+PA
SDA(2)=SDA(2)+REPSF
SDA(3)=SDA(3)+DML
SDA(4)=SDA(4)+RFPSD
SDA(5)=SDA(5)+CSHTD
SDA(6)=SDA(6)+DMM
SDA(7)=SDA(7)+CLS
SDA(8)=SDA(8)+U12
SDA(9)=SDA(9)+CSHTF+CFT
SDA(10)=SDA(10)+U17
SDA(11)=SDA(11)+CED+CTSD
SDA(12)=SDA(12)+CEP+CEV+CTSV
SDA(13)=SDA(13)+CTSOFT
SDA(14)=SDA(14)+CMPPY
SDA(15)=SDA(15)+CIVP
SDA(16)=SDA(16)+CTSP
These statements accumulate case totals for the following maintenance support costs; pay and allowance for military personnel (PA), consumed and support material at field facilities (REPSF), personnel cost at Depot (DML), consumed and support material at Depot (RFPSD), shipping prime equipment and mod kits to Depot (CSHTD), mod kits (DMM), civilian maintenance labor (CLS), training test and repair personnel (U12), housing test equipment and shipping mod kits to field (CSHTF+CFT), maintaining line items in the supply system (U17), development of prime and test equipment

(CED+CTST), procurement of installed LRUs less salvage value (CEP+CEV+CTSV), development of test software (CTSOFT), training courses for test equipment (CMPPY), stock inventory (CIVP), and procurement of test equipment (CTSP). These values are output in Subroutine Eight using the DA PAM 11-4 format.

3. PCD=PCD+PVCD
PCP=PCP+PVCP
PCR=PCR+PVCR
PCS=PCS+PVCS
These statements accumulate the present value costs for Development, Acquisition, O&M, and Salvage, respectively.

4. PCGT=PCGT+PVGCT
This statement accumulates the present value cost total.

5. SDEL=SDEL+DELTA
This statement accumulates the difference (DELTA) between the expected value and the dedicated value of manpower costs.

6. SEPV=SEPV+EPVGCT
SEMPT=SEMPT+ECMPT
SEPC=SEPC+EPCGT
These statements accumulate the costs for the present value grand total, test equipment manpower, and the expected grand total, respectively. Total manpower cost is the personnel support for both test and prime equipment. An expected value is determined by subtracting DELTA from the dedicated value.

7. SPDEL=SPDEL+PDELTA
This statement accumulates the present value difference in dedicated manpower cost and expected manpower cost.

8. SPCR=PCR-SPDEL
This statement reduces the present value cost of the O&M phase by the accumulated manpower delta (SPDEL).

9. CCTSR=CCTSR+CTSR
This statement accumulates the support cost for all test equipment. This includes both material and personnel cost.

10. CTRF=CTRF+TRNGF+TRNGRF
CTRDEP=CTRDEP+TRNGD+TRNGRD
These statements accumulate the O&M cost to train personnel for test and repair at Field and Depot maintenance, respectively. The cost at the Depot level

level is for civilian personnel.

11. CCMF=CCMF+(CMPR+CMPRR)-(U1+U3-U2)-(TRNG+TRNGR)
This statement accumulates the maintenance manpower costs for the field facilities. CMPR and CMPRR include the test and repair maintenance cost at the field and Depot and the cost of training test (TRNG) and repair (TRNGR) personnel. Since field maintenance manpower cost is all that is expected in this statement, the Depot cost (U1+U2-U3) and the training costs are substracted.

12. CCMD=CCMD+U1+U3-U2
This statement accumulates the Depot test and repair manpower cost for maintenance of installed equipments. U1 and U3 are the Depot test and repair costs and U2 is the cost of self-support for either Type I or Type II test equipment. Since U1 and U3 contain the self-support costs, U2 is subtracted to get only maintenance cost for the installed equipments.

13. CIVREC=CIVREC+CIVR
CSAREC=CSAREC+CSAR
These statements accumulate the cost of consumed material and the cost of inventory management, respectively.

14. CQTU=CQTU+CQTT
CQTM=CQTM+CQTMT
CQTP=CQTP+CQTPT
These statements accumulate the initial provisioning cost for LRUs, Modules and Parts stocked at all levels, respectively.

15. CQTUMP=CQTU+CQTM+CQTP
This statement computes the total cost of initial provisions.

16. CCMFD=CCMF+CCMD
This statement computes the total cost of maintenance. This includes the cost at Depot and the field facilities.

17. CTR=CTRF+CTRDEP
This statement computes the total cost of training civilian personnel at the field and Depot facilities.

18. CTREC=CCTSR+CCF+CCMFD+CTR+CIVREC+CRT+
CWH+CSAREC+CSH
This statement computes the total recurring investment costs.

19. IF(IBG.EQ.1) WRITE (K004FX,BUG)
This statement prints the values for the list of variables in NAMELIST/BUG/ if the user selects the debugging option (IBG=1).

20. RETURN
END
These statements terminate execution of LGCST and returns to the logic of the calling program (LOGAM).

4.8 TOE Operation and Support Costs

The subroutines of this section compute the costs based on a TOE. The inputs to these routines are the "T" array of NAMELIST/L/ and the AR570-2 MACRIT data base. The outputs from these subroutines are accumulated into an array that is printed in the DA PAM 11-4 format. These costs are included with the case totals output from LGCST. The subroutines that are described in this section are:

1. OPER — Driver routine from the TOE analysis. OPER calls the subroutines that compute the costs for the TOE line items.
2. ONE — Computes the total people/category and Pay and Allowances.
3. TWO — Computes personnel related costs such as transients, replacements, quarters, etc.
4. THREE — Computes the cost of Petroleum, Oil and Lubricants.
5. FOUR — Computes the cost of Ammunition.
6. FIVE — Sums the instrumentation costs.
7. SIX — Computes Artillery/Ordnance and Follow-on-Training costs.
8. SEVEN — Converts the TOE costs to the requested output units (AMULT) and accumulates the pertinent TOE costs with LOGAM maintenance costs in preparation for output in the DA PAM 11-4 format.
9. SUPTMH — Reads the AR570-2 MACRIT data file and accumulates maintenance positions by MOS.
10. SGAPER — Computes the number of personnel by grade for the maintenance positions accumulated in Subroutine SUPTMH. A Standard of Grades Authorization (SGA) table is used to compute the personnel numbers.
11. TOEPER — This routine incorporates the MACRIT maintenance support personnel into the TOE data array (T).

4.8.1 SUBROUTINE OPER

This subroutine is the driver for computing Operation and Support (O&S) costs based on a Table of Organization and Equipment (TOE). OPER interrogates a line item of the TOE input table (T) and calls the relevant subroutine for evaluating that line item. The "T" array is a 2000 element table which is dimensioned to input 200 line items with a maximum of 10 values per line item. "T" inputs are read through NAMELIST/L/. To execute OPER, the user must set IOPER=1 in the Namelist inputs. Subroutine OPER will be called from the main program if a case total has been requested (NU=-2 or -3) and IOPER=1.

Maintenance positions obtained from the AR570-2 MACRIT data base can be incorporated into the TOE namelist inputs. The user inputs the personnel fields including the grade and annual costs and the program will supply the quantity and the flags for the type of charges. For this logic to work the user must not input personnel quantities in these fields. Any personnel data not pertaining to the MACRIT data is input in the normal way.

Subroutine SUPTMH is referenced from OPER to read the MACRIT data file and build the personnel position table (POS). Subroutine TOEPER is called to incorporate these positions into the NAMELIST TOE inputs.

A. This section of OPER contains the assignment of variables in computer memory with labled COMMON blocks and also assigns the TOE data input and output list using NAMELIST.

1. DIMENSION POS (3,13)
   This statement dimensions an array for storing maintenance support personnel by grade. Maintenance personnel are stored for 13 grades at 3 maintenance levels (E,DS,GS). The number of maintenance positions per grade are evaluated in Subroutine SGAPER.

2. COMMON/POUT/POUT(16)
   COMMON/OFF/O(6)
   COMMON/ENLM/ENLM(7)
   The first reference to these COMMON blocks was in LOGAM. All of the blocks are initialized in BLKDAT. POUT is used in the subroutines referenced by OPER to store O&S cost for output in Subroutine EIGHT. COMMON OFF and ENLM are used to store numbers of personnel and pay and allowances for officers and enlisted men, respectively.

3. COMMON/INPUT/
   This common block contains the list of input variables read for NAMELIST/L/. The TOE input array (T) is one of

these variables. IMF and IBG are the only other variables of this list that are used in OPER. IMF is the file position of the MACRIT data tape for the theater being evaluated. IBG is a debugging flag. Refer to section A of the main program for a listing of these variable.

4. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
SENSY(266),T(10,200)
This statement contains all of the dimensioned variables in COMMON/INPUT. T(10,200) is the TOE input array.

5. NAMELIST/OPERER/T
This statement is used to print TOE input data when an error is encountered in the TOE inputs.

B. This section of OPER interrogates the TOE inputs to determine the type of cost to be evaluated. There are nine subroutines called here. Six of these compute costs based on the type of inputs. The type of data is designated by the first element of a line item's input. Two subroutines are used to read the MACRIT data file and to incorporate maintenance positions into the personnel fields for NAMELIST TOE. One subroutine (SEVEN) is used to convert O&S costs to LOGAM cost units. The data are input in fields of 10 values per line item with a maximum of 200 line items. In this section the first location of each set of 10 values is checked to determine which of the subroutines to call to evaluate the cost of the set. The argument list for the subroutines called in this section include the variable J, where J is the line item number.

1. LIX=0
This statement is used in Subroutine SIX to initialize variables on the first pass through the routine.

2. IF (IMF.GT.0) CALL SUPTMH (IMF,POS)
This statement will call subroutine SUPTMH to read the MACRIT data file when the file position (IMF) for the data is input. IMF allows for having multiple sets (theaters) of data on the same tape or disk file. POS is returned from SUPTMH with the positions by grade.

3. J=0
10 J=J+1
IF(J.EQ.201) GO TO 20
These statements set and increment the counter for the next line item to be evaluated. If more than 200 line items are entered, a transfer is made to statement 20 where the TOE inputs are printed and execution of the

program terminated.

4. IF (T(I,J).NE.1.) GO TO 15
This statement tests the TOE input field for line item J to check for personnel data. If the value is unity the next statement will be executed.

5. IF (IMF.GT.0.AND.T(2,J).EQ.0.) CALL TOEPER(J,POS)
This statement is entered only if the TOE line item (J) is personnel related. A check is made on the second field for input quantities of personnel. If quantities are input, the MACRIT maintenance position logic will not be entered. After the MACRIT file has been read ($IMF \geq 1$) and the quantity of personnel is to be determined ($T(2,J)=0$), then TOEPER is called. TOEPER will incorporate personnel positions into the grade specified by line item J.

6. IF(T(1,J).EQ.1.)CALL ONE(J)
This statement will call Subroutine ONE to compute total people/category and pay and allowances.

7. IF(T(1,J).EQ.2.)CALL TWO(J)
This statement will call Subroutine TWO to compute personnel related costs.

8. IF(T(1,J).EQ.3.)CALL THREE(J)
This statement will call Subroutine THREE to compute fuel costs.

9. IF(T(1,J).EQ.4.)CALL FOUR(J)
This statement will call Subroutine FOUR to compute ammunition costs.

10. IF(T(1,J).EQ.5.)CALL FIVE(J)
This statement will call Subroutine FIVE to compute instrumentation costs.

11. IF(T(1,J).EQ.6)CALL SIX(J)
This statement will call Subroutine SIX to compute ARTY/ORD costs, Follow-On-Training (FOT) firing cost for missiles, and adds the cost of instrumentation per firing from Subroutine FIVE.

12. IF(T(1,J).EQ.7.)CALL SEVEN(IBG)
This statement will call Subroutine SEVEN to prepare the output list in the proper units of cost.

13. IF(T(1,J).EQ.8.)GO TO 30
A value of 8 in the first location of a cost element's field will terminate The TOE evaluation. Logic is transferred to statement 30 where a return is made to

LOGAM.

14. GO TO 10
This statement "loops" back to FORTRAN statement 10 where the line item counter (J) is incremented and the evaluation for the next line item will begin.

15. WRITE(6,OPERER)
This statement prints the TOE data when more than 200 line items (fields of 10 values) are input.

16. 20 CONTINUE
STOP
This statement will stop program execution once an input error is detected.

17. 30 RETURN
END
These statements signal a normal termination to the TOE evaluation and will return logic back to LOGAM.

4.8.2 SUBROUTINE ONE

This subroutine is called from Subroutine OPER to compute total people/category and Pay and Allowances.

A. This section of ONE contains the assignment of variables in computer memory with labeled COMMON blocks and also assigns TOE and other data to NAMELISTs for printing.

1. SUBROUTINE ONE(J)
   This statement is the entry point for the subroutine. ONE is called from Subroutine OPER when the value of T(1,J)=1. A value of one for T(1,J) indicates that line item J is personnel related.

2. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

3. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

4. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

5. COMMON/ENLM/ENLM(7)
   COMMON/OFF/O(6)
   These common blocks are used to accumulate TOE people and Pay and Allowances for enlisted and officer personnel.

6. REAL MAINT, NMAINT, NCREW, NDED, DED
   This statement assigns floating point status (real) to variables that would ordinarily be typed integer.

7. NAMELIST/ONER/T
   NAMELIST/BUGONE/J,POUT,ENLM,O
   These statements assign TOE inputs and TOE evaluated data to NAMELIST names for output when debugging of the data is required.

B. This section of ONE interrogates a "personnel count" entry of TOE data and sets the assignments for crew, maintenance, and dedicated individuals. The logic to transfer to either officer or enlisted men evaluations is included here.

1. MAINT=AB(T(9,J))
   NMAINT=1.-MAINT
   These statements set the maintenance personnel multipliers. If T(9,J)=1., then the personnel assigned to TOE level J is for maintenance. When T(9,J)=0., the assignment will be for non-maintenance (NMAINT=1) personnel.

2. CREW = AB(T(7,J))
   NCREW=1. - CREW
   These statements set the crew personnel multipliers. If T(7,J)=1., then the personnel assigned to TOE level J is for crew. When T (7,J)=0., the assignment will be non-crew (NCREW=1.) personnel.

3. DED=AB(T(5,J)+T(6,J))
   NDED=1.-DED
   These statement set the dedicated and/or overhead personnel multipliers. If either the dedicated flag T(6,J) or the overhead flag T(5,J) are on (=1), the dedicated multiplier DED will equal 1.

4. IF(T(3,J).EQ.0.) GO TO 10
   IF(T(3,J).EQ.1.) GO TO 20
   WRITE (6, ONER)
   STOP
   These statements check the officer/enlisted men flags to determine which section of this program to transfer to for personnel cost computations. T(3,J)=0 will transfer logic to statement 10 to evaluate officer data. T(3,J)=1 will transfer logic to statement 20 to evaluate the enlisted men costs. Any other value for T(3,J) will result in the printing of the TOE inputs and termination of program execution.

C. The next section of ONE computes the pay and allowance for officers/WOs. The computations include costs for crew, maintenance, and support that are accumulated for all sets of TOE data. The user should be aware that to input both the crew and maintenance flags equal to 1 will result in a double accounting of dedicated/overhead costs. This will also be true for the costing of enlisted men. The computation of O(4) and O(5) for officers and E(4) for enlisted men are the costs impacted. In the following statements T(2,J) is the quantity of officers and T(10,J) is the cost per officer per year.

1. 10 O(01)=O(01)+CREW*T(2,J)

This statement accumulates the number of officers assigned to crew.

2. O(02)=O(02)+NCREW*T(2,J)
This statement accumulates the number of officers that are not assigned to the crew. This will be the number of officers charging indirect.

3. O(03)=O(03)+DED*NCREW*NMAINT*T(10,J)*T(2,J)
This statement accumulates the total annual Pay and Allowance for dedicated organizational/overhead officers less crew and maintenance officers.

4. O(04)=O(04)+DED*MAINT*T(10,J)*T(2,J)
This statement accumulates the total annual Pay and Allowance for all dedicated/overhead maintenance officers.

5. O(5)=O(05)+DED*CREW*T(10,J)*T(2,J)
This statement accumulates the total annual Pay and Allowance for all dedicated/overhead officers assigned to the crew.

6. O(6)=O(06)+T(2,J)
This statement accumulates the total number of officers in the TOE organization.

7. POUT(01)=POUT(01)+T(10,J)*CREW*YR*T(2,J)
This statement accumulates the total O&S lifecycle Pay and Allowance for officers assigned to crew into the total direct P&A array.

8. POUT(02)=POUT(02)+T(10,J)*NCREW*YR*T(2,J)
This statement accumulates the total O&S lifecycle Pay and Allowance for officers not assigned to crew into the total non-direct P&A array.

9. IF (IBUG.EQ.1) WRITE (6,BUGONE)
This statement will print the values of the variable list in NAMELIST/BUGONE/when the debugging flag IBG is turned on (unity) in the NAMELIST/L/input data.

10. RETURN
This statement returns program logic control back to Subroutine OPER.

D. The following section of code computes the Pay and Allowance for enlisted men. The variables used here are the same as those used above for officers except that the results are accumulated into "E" instead of "O". T(2,J) is the quantity of enlisted men and T(10,J) is the annual cost per enlisted men.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (ONE)

1. 20 ENLM(01)=ENLM(01)+CREW*T(2,J)
   This statement accumulates the number of enlisted men that are assigned to crew.

2. ENLM(02)=ENLM(02)+NCREW*NMAINT*T(2,J)
   This statement accumulates the number of enlisted men at the organization that are neither crew nor maintenance personnel.

3. ENLM(03)=ENLM(03)+DED*NCREW*NMAINT*T(10,J)*T(2,J)
   This statement accumulates total annual Pay and Allowance for all overhead/dedicated organizational enlisted men.

4. ENLM(04)=ENLM(04)+DED*MAINT*T(10,J)*T(2,J)
   This statement accumulates the total annual Pay and Allowance for all overhead/dedicated maintenance enlised men.

5. ENLM(05)=ENLM(05)+CREW*T(10,J)*T(2,J)
   This statement accumulates the total annual Pay and Allowance for all crew enlisted men.

6. ENLM(06)=ENLM(06)+T(2,J)*DED*MAINT
   This statement accumulates the total number of overhead/dedicated enlisted men in organizational maintenance.

7. ENLM(07)=ENLM(07)+T(2,J)*NMAINT
   This statement accumulates the total number of enlisted personnel that are non-maintenance (dedicated, overhead and crew).

8. POUT(01)=POUT(01)+T(10,J)*CREW*YR*T(2,J)
   This statement accumulates the total O&S lifecycle Pay and Allowance for enlisted crew into the total direct P&A array.

9. POUT(02)=POUT(02)+T(10,J)*NCREW*MAINT*YR*T(2,J)
   This statement accumulates the total O&S lifecycle Pay and Allowance for organizational maintenance enlisted personnel into the non direct P&A array.

10. IF(IBUG.EQ.1) WRITE (6,BUGONE)
    This statement prints the values for the parameter list of NAMELIST/BUGONE/ when debugging of a data case is requested (IBG=1).

11. RETURN
    END
    These statements terminate subroutine ONE execution and returns control to Subroutine OPER.

4.8.3 SUBROUTINE TWO

This subroutine computes O&S costs for a TOE that are personnel related. Some of the types of cost evaluated here are replacements, transients, PCS, quarters, and other indirect costs.

A. The first section of TWO includes the subroutine name and the assignment of variables in computer memory.

1. SUBROUTINE TWO(J)
   This statement is the entry point for the subroutine. TWO is called from Subroutine OPER when the value of T(1,J)=2. J is the TOE line item number.

2. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

4. COMMON/PERSON/PERS(5,2),PERL(5,2),HPD(5,2),
   WPD(5,2),WPDL(5,2)
   This common block contains the manpower accumulations for a case total. PERS is used in this subroutine to compute the transient cost of enlisted men for test and repair personnel. HPD is the accumulated test and repair manhours from Subroutine TEMWF. The other parameters are accumulated in Subroutine PERSON.

5. COMMON/ENLM/ENLM(7)
   COMMON/OFF/O(6)
   These common blocks are used to accumulate TOE people and Pay and Allowances for enlisted and officer personnel.

6. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

7. NAMELIST/BUGTOO/J,POUT,ENLM,O
   NAMELIST/TWOER/T,J,POUT,ENLM,O

These statements set up a parameter list of TOE and personnel related data to output when debugging of a data case is required or when an error is encountered in the TOE inputs.

B. The next section of TWO sums the LOGAM maintenance people and checks the value of the second element in a TOE field, T(2,J), to determine the area of TWO to transfer to. There are three possible values for T(2,J). T(2,J)=1 signifies that the associated input field is for enlisted men. T(2,J)=2 and its associated data will compute costs for replacements, transients, and quarters for officers. T(2,J)=3 will cost officer contribution to permanent change of station.

1. QEPM=PERS(1,1)+PERS(2,1)+PERS(3,1)+PERS(4,1)+
   PERS(1,2)+PERS(2,2)+PERS(3,2)+PERS(4,2)
   This statement computes the number of field level enlisted test and repair personnel as determined by the LOGAM logistics support calculations. These values were computed in Subroutine PERSON. The assignments for these values are:

   PERS(1,1) - Fault isolate and test at E
   PERS(2,1) - Test and checkout at E
   PERS(3,1) - Test and checkout at DS
   PERS(4,1) - Test and checkout at GS
   PERS(1,2) - Remove and replace at E
   PERS(2,2) - Repair at E
   PERS(3,2) - Repair at DS
   PERS(4,2) - Repair at GS

2. If(T(2,J).EQ.1) GO TO 10
   IF(T(2,J).EQ.2) GO TO 20
   IF(T(2,J).EQ.3) GO TO 30
   These statements will transfer logic to either statement 10, 20,, or 30 to evaluate TOE inputs for enlisted men, officer replacements, and officer PCS, respectively.

3. WRITE(6,TWOER)
   STOP
   This statement prints TOE input and computed values when an unacceptable number is input for T(2,J). Any value other than a 1,2 or 3 for T(2,J) will cause this print and termination of program execution.

C. This section of TWO computes the Operation and Support costs contributed by enlisted men to PCS, replacement, other direct, and other indirect costs. YR is the number of years of operation and support.

1. 10 PSCE=(QEPM+ENLM(7))*T(3,J)*T(4,J)*YR

POUT(3)=POUT(3)+PSCE
These statements compute the O&S cost of Permanent Change of Station for enlisted men and accumulate the value with other PCS costs (P(3)). T(3,J) and T(4,5) are the rate of change of station and the cost per change, respectively. ENLM(7) is the number of non-maintenance enlisted men.

2. POUT(6)=POUT(6)+T(5,J)*YR
This statement computes Other Direct O&S cost of the organization for enlisted men. T(5,J) is the cost per year.

3. CEPRC=ENLM(1)*T(6,J)*T(7,J)*YR
This statement computes the O&S cost for enlisted crew attrition. ENLM(1) is the number of enlisted crew personnel, T(6,J) is the attrition rate, and T(7,J) is the replacement cost per man including training.

4. ODEPRC=ENLM(2)*T(8,J)*T(9,J)*YR
This statement computes the O&S Attrition cost for enlisted personnel at organizations that are neither crew nor maintenance. ENLM(2) is the number of enlisted personnel at organization, T(8,J) is the attrition rate, and T(9,J) is the replacement cost per man.

5. POUT(7)=POUT(7)+CEPRC+ODEPRC
This statement adds the Attrition cost for enlisted crew and organization to the personnel replacement cost accumulation.

6. POUT(11)=POUT(11)+T(10,J)*YR
This statement accumulates O&S "Other Indirect" costs. T(10,J) is other indirect costs per year.

7. IF(IBUG.EQ.1) WRITE (6,BUGTOO)
This statement prints the O&S evaluations from the post processing routines when debugging of the output is required.

8. RETURN
This statement terminates the enlisted personnel computations and returns logic back to Subroutine OPER.

D. This section of TWO computes the O&S cost contributed by officers for replacements, transients, and quarters. YR is the number of O&S years. This section of code also includes the cost of enlisted personnel, transients, and quarters.

1. 20 CORC=O(1)*T(3,J)*T(4,J)*YR
OORC+O(2)*T(5,J)*T(6,J)*YR

These statements compute the O&S Attrition cost for officers assigned to crew and to organization, respectively. O(1) and O(2) are the number of officers, T(3,J) and T(5,J) are the attrition rates, and T(4,J) and T (6,J) are the cost per replacement.

2. POUT(7)=POUT(7)+CORC+OORC
This statement adds the officer Attrition costs to the accumulated O&S replacement cost (POUT(7)).

3. CTPPD=O(3)*T(7,J)+ENLM(3)*T(8,J)
This statement computes the Transient cost for officers and enlisted men. O(3) and ENLM(3) are the annual Pay and Allowance for dedicated organizational/overhead officers and enlisted men, respectively. T(7,J) and T(8,J) are the percentages of P&A that are charged for transient cost.

4. CTPPM=O(4)*T(7,J)+(ENLM(4)+CCM/(YR*AMULT*T(8,J)
This statement computes Transient costs as a factor of Pay and Allowance for dedicated/overhead maintenance personnel for officers and enlisted men, respectively. T(7,J) and T(8,J) are the factors. CCM was computed in LGCST as the maintenance manpower cost of test and repair personnel for prime equipment. CCM belongs to the COMMON/ZERO/ list.

5. CTPPC=O(5)*T(7,J)+ENLM(5)*T(8,J)
This statement computes the Transient cost of officers and enlisted men as a factor of the Pay and Allowance for dedicated/overhead crew. T(7,J) and T(8,J) are the percentage factors.

6. POUT(8)=POUT(8)+(CTPPD+CTPPM+CTPPC)*YR
This statement adds the Transient costs into the accumulated O&S output array.

7. CQMUO=O(6)*T(9,J)*YR
CQMUE=ENLM(6)*T(10,J)*YR
These statements compute the cost of living quarters for officers and enlisted men, respectively. O(6) and ENLM(6) are total officers and enlisted men in the TOE organization. T(9,J) is the annual cost per officer and T(10,J) is the average annual cost per enlisted men.

8. POUT(9)=POUT(9)+CQMUD+CQMUE
This statement sums the cost of quarters for all organizational personnel.

9. IF(IBUG.EQ.1) WRITE(6,BUGTOO)
This statement prints the O&S post processor arrays

when a debugging request is made.

10. RETURN
This statement terminates execution of this subroutine and returns logic to OPER.

E. This section of Code computes the contribution of officers to the cost of Permanent Change of Stations (PCS). YR is the number of years of O&S. The cost of medical support is also computed in this section.

1. 30 PCSO=T(3,J)*T(4,J)*YR*O(6)
These statements computes the cost of PCS for officers. T(3,J) and T(4,J) are the rate of change and the cost per officer, respectively. O(6) is the number of officers in a TOE organization.

2. POUT(3)=POUT(3)+PCSO
This statement accumulates the cost of PCS for officers (PCSO) into the PCS output element.

3. POUT(10)=POUT(10)+(O(6)+QEPM+ENLM(7))*T(5,J)*YR
This statement computes the cost of medical support. O(6), QEPM, and ENLM(7) are the number of TOE organization officers, number of field level test and repair personnel, and the number of enlisted men that are non-maintenance, respectively. T(5,J) is the average annual cost per person for medical support.

4. IF(IBUG.EQ.1) WRITE (6,BUGTOO)
This statement prints the O&S cost post processor arrays when a debugging request is made.

5. RETURN
END
These statements terminate execution of the subroutine and transfers logic to OPER.

4.8.4 SUBROUTINE THREE

This subroutine computes the O&S cost of Petroleum, Oil and Lubricant (POL). THREE is called from Subroutine OPER when the value for T(1,J)=3.

A. This section of THREE includes the entry point (subroutine name), the common assignments, and the NAMELIST debugging statement.

1. SUBROUTINE THREE (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. There are a maximum of 200 line items with each line item having 10 possible input values.

2. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

4. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

5. NAMELIST/BUGTRI/T,J,POUT
   This statement provides a list of TOE inputs and outputs to print when debugging of output is requested.

B. This section of code computes the POL cost and terminates execution of the subroutine.

1. POUT(4)=POUT(4)+T(10,J)*T(4,J)*T(5,J)*T(6,J)*
   T(7,J)*YR
   This statement computes the cost of Petroleum, Oil, and Lubricants for a specific item of equipment and accumulates the cost into the POL element (POUT(4)) of the output array. The other parameters used in the equation are:

   T(10,J) Quantity of TOE line item using

	the fuel.
T(4,J)	Hours per year that the line item is used.
T(5,J)	Rate of fuel usage for the line item.
T(6,J)	Cost of fuel per usage rate unit.
T(7,J)	Fractional increase over fuel use to allow for oil and lube.
YR	Number of O&S years

2. IF(IBUG.EQ.1) WRITE (6,BUGTRI)
This statement prints the TOE inputs and the O&S output array from NAMELIST/BUGTRI/ when a debugging request is made in LOGAM inputs.

3. RETURN
END
These statements terminate execution of Subroutine THREE and returns logic to Subroutine OPER.

4.8.5 SUBROUTINE FOUR

This subroutine computes the O&S cost of ammunition. Subroutine FOUR is called from Subroutine OPER when the TOE input T(1,J)=4.

A. This section of Subroutine FOUR includes the entry point (subroutine name), the COMMON assignments, and the NAMELIST debugging statement.

1. SUBROUTINE FOUR (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. For TOE inputs there is a maximum of 200 line items with each line item having 10 possible input values. T(1,J) must equal four for this subroutine to be referenced.

2. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

4. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

5. NAMELIST/BUGFOR/T,J,POUT
   This statement provides a list of TOE inputs and outputs to print when a debugging request is made from LOGAM.

B. This section of FOUR computes the cost of ammunition during the O&S phase of a program and terminates subroutine execution.

1. POUT(5)=POUT(5)+(T(2,J)+T(3,J)+T(4,J)+T(5,J))*YR
   This statement computes the cost of ammunition for a TOE line item and accumulates this cost into the output array (P(5)). The factors that make up ammunition cost are:

   T(2,J) Cost per year to transport ammunition.

T(3,J)	Cost per year associated with ammo firings for unit training.
T(4,J)	Cost per year for range support of ammo firings.
T(5,J)	Cost per year for A.P.L. data takers associated with firings.
YR	Number of O&S years.

2. IF(IBUG.EQ.1)WRITE(6,BUGFOR)
This statement prints the TOE inputs and the O&S output array from NAMELIST/BUGFOR/ when a debugging request is made from LOGAM.

3. RETURN
END
These statements terminate the execution of Subroutine FOUR and returns control to Subroutine OPER.

4.8.6 SUBROUTINE FIVE

This subroutine sums instrumentation costs temporarily into the output array, POUT(12), for later use in Subroutine SIX. FIVE is called from Subroutine OPER when the value for T(1,J)=5.

A. This section of FIVE includes the subroutine name, COMMON assignments, and the NAMELIST debugging statement.

1. SUBROUTINE FIVE (J)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. There are a maximum of 200 line items with each line item having 10 possible input values.

2. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

3. DIMENSION G(20),H(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

4. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

5. NAMELIST/BUGFIV/T,J,POUT
   This statement provides a list of TOE inputs and outputs to print when a debugging request is made from LOGAM.

B. This section of FIVE sums the instrumentation costs into location POUT(12) of the O&S printout array and terminates the subroutine execution.

1. DO 10 K=2,10
   10 POUT(12)=POUT(12)+T(K,J)
   This DO loop accumulates the 9 possible instrumentation cost inputs for line item J into output array element P(12).

2. IF(IBUG.EQ.1) WRITE (6,BUGFIV)
   This statement prints the TOE inputs and the O&S output

array from NAMELIST/BUGFIV/ when a debugging request is made from LOGAM.

3. RETURN
END
These statements terminate execution of this subroutine and transfers control to Subroutine OPER.

4.8.7 SUBROUTINE SIX

This subroutine computes Artillery/Ordnance and Follow-On-Training (FOT) firing costs for missiles. The instrumentation cost summed in Subroutine FIVE is used to compute ARTY/ORD costs. SIX is called from Subroutine OPER when the value of T(1,J)=6.

A. This section of SIX includes the subroutine name, COMMON assignments, and debugging NAMELIST statements.

1. SUBROUTINE SIX (J,LIX)
   The argument "J" in this statement is the line item number for which a cost is being evaluated. For TOE inputs there is a maximum of 200 line items with each line item having 10 possible input values. LIX is an initialization flag used to reset the temporary location POUT(12) for instrumentation cost.

2. COMMON/INPUT/
   This common block contains the list of variables read with the NAMELIST/L/ inputs. IBG, YR and T are the only variables of this list that are used in this subroutine. IBG is the debugging flag and YR is the number of years of O&M. T is the array of inputs for the TOE.

3. DIMENSION G(20),H(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),ZU(4),
   SENSY(266),T(10,200)
   This statement contains all of the dimensioned variables of common block INPUT.

4. COMMON/POUT/POUT(13),AMULT
   This common block contains the array for accumulating the Operation and Support costs that are printed in Subroutine EIGHT using the DA PAM 11-4 format. AMULT is the cost unit multiplier input in LOGAM.

5. NAMELIST/SIXER/T,J,POUT
   NAMELIST/BUGSIX/T,J,POUT
   These statements provide a list of the O&S data to be printed if an error occurs in the TOE inputs or a debugging request is made in LOGAM.

B. This section of SIX stores the instrumentation costs, resets the temporary location used for instrumentation and determines whether to cost ARTY/ORD or FOT.

1. IF(LIX.EQ.1) GO TO 30
   LIX is set to zero in Subroutine OPER so that on the first call to Subroutine SIX, the instrumentation cost

of missiles will be stored into a new variable and the temporary location POUT(12) can be reset. This occurs only one time per TOE evaluation because LIX is set to one in this subroutine and the resetting logic will be bypassed on the next call to SIX. For missile instrumentation costs to be included in SIX, the TOE line item data for instrumentation must first be evaluated in Subroutine FIVE.

2. SIGIN=POUT(12)
This statement stores the missile instrumentation cost that was evaluated in Subroutine FIVE into the local variable SIGIN.

3. POUT(12)=0.
This statement resets the temporary location for the instrumentation cost.

4. LIX=1
This statement sets the instrumentation cost flag so that the two statements above will never be entered again.

5. 30 IF(T(2,J).EQ.1.) GO TO 10
IF(T(2,J).EQ.2.) GO TO 20
These statements check the second input value of a TOE line item to determine if the input is for ARTY/ORD or FOT costs. A value of 1 will transfer logic to do ARTY/ORD costs and a value of 2 will cost FOT.

6. WRITE(6,SIXER)
STOP
Any value for T(2,J) other than a 1 or 2 is an error condition that will result in the printout of the variables in NAMELIST/SIXER/ and termination of program execution.

C. This section of SIX computes the Artillery/Ordnance costs during the O&S phase of a program. This logic is executed when T(2,J)=1.

1. 10 CATAO=T(3,J)*T(10,J)
This statement computes the annual cost of transport for ARTY/ORD firings. T(3,J) is the cost of transport per firing and T(10,J) is number of firings per year.

2. CAMIAO=T(10,J)*SIGIN
This statement computes the annual missile instrumentation costs where SIGNIN is the instrumentation cost per firing.

3. CARSUO=T(10,J)*(T(5,J)+T(6,J))

This statement computes the annual cost of range support and use. T(5,J) is the support cost per firing and T(6,J) is cost of range use per firing.

4. CACSAO=T(10,J)*T(7,J)
This statement computes the annual contractor support for ARTY/ORD firings. T(7,J) is the cost of support per firing.

5. POUT(5)=POUT(5)+(CATAO+CAMIAO+CARSUO+CACSAO+T(4,J))*YR
This statement computes the ARTY/ORD firing costs over the O&S phase and accumulates this cost into the unit training AMMO and missile cost POUT(5). T(4,J) is the cost per year for APL data takers associated with firings and YR is the number of O&S years.

6. IF(IBUG.EQ.1) WRITE (6,BUGSIX)
This statement prints the TOE inputs and O&S outputs from the NAMELIST/BUGSIX/ data list when a debugging request is made in LOGAM.

7. RETURN
This statement terminates Subroutine SIX execution and returns program control to Subroutine OPER.

D. This section of SIX computes the Follow-on-Training firing costs. T(10,J) in the following statements is the number of firings per year.

1. 20 CATFOT=T(10,J)*T(3,J)
This statement computes the annual cost of transportation for FOT firings. T(3,J) is the cost per firing.

2. CAIFOT=T(10,J)*SIGIN
This statement computes the annual instrumentation costs for FOT firings. SIGIN is the instrumentation cost per firing.

3. CARSUF=T(10,J)*(T(5,J)+T(6,J))
This statement computes the cost of FOT range support and use. T(5,J) is the cost per firing for range support and T(6,5) is the annual cost per firing for range use.

4. CACSFO=T(10,J)*T(7,J)
This statement computes the annual cost of FOT contractor support where T(7,J) is the cost per firing.

5. CAPFOT=T(10,J)*T(8,J)
This statement computes the annual cost of overseas missile preparation where T(8,J) is the cost per FOT

firing.

6. POUT(5)=POUT(5)+(CATFOT+CAIFOT+CARSUFT+CACSFO+
CAPFOT+T(4,J))*YR
This statement computes the FOT firing costs over the O&S phase and accumulates this cost into the unit training AMMO and missile cost POUT(5). T(4,5) is the annual cost for data takers associated with firings. YR is the number of O&S years.

7. IF(IBUG.EQ.1)WRITE(6,BUGSIX)
This statement prints the TOE inputs and the O&S outputs from the NAMELIST/BUGSIX/ data list when a debugging request is made from LOGAM.

8. RETURN
END
These statements terminate execution of Subroutine SIX and returns program control to Subroutine OPER.

4.8.8 SUBROUTINE SEVEN

This subroutine prepares the output list (POUT) in the proper units of cost. SEVEN is called from Subroutine OPER when the value of T(1,J)=7. A value of 7 should be input to TOE just prior to printing the outputs. AMULT from LOGAM is used in the conversion.

A. The next section of SEVEN contains the subroutine name, assignment of variables in computer memory using labled COMMON blocks and assigns the output list POUT to a NAMELIST name for debugging purposes.

1. SUBROUTINE SEVEN (IBUG)
   IBUG is a debugging flag set in NAMELIST /L/.

2. COMMON/POUT/POUT(13),AMULT
   COMMON/DAPAM/SDA(10),WD(5)
   These common blocks contain the costs for case accumulations that are to be printed in the DA PAM 11-4 format. POUT includes those LOGAM maintenance costs that are a part of the same PAM 11-4 format as the TOE costs. When TOE costs are evaluated they are accumulated with certain LOGAM O&S costs in this subroutine. DAPAM contains the costs from the LOGAM evaluations.

3. NAMELIST/BUGSEV/POUT
   This statement assigns the TOE ouput data array POUT to NAMELIST BUGSEV for printing when debugging of a data case is required.

B. The next section of SEVEN converts cost to the proper units, adds outputs from LOGAM to TOE outputs, and totals the costs.

1. DO 10 I=1,11
   10 POUT(I)=POUT(I)*AMULT
   This DO loop uses AMULT, the conversion factor from LOGAM, to put the TOE costs in the proper units.

2. POUT(6)=POUT(6)+SDA(10)
   This statement adds the cost of maintaining a line item in the supply system (SDA(10)) to "Other Direct" TOE costs. SDA(10) is computed in LGCST and is the case total accumulation of U17.

3. POUT(7)=POUT(7)+SDA(8)
   This statement adds the cost of training test and repair personnel (SDA(8)) to the "Personnel Replacement" TOE costs. SDA(8) is computed in LGCST and is the case total accummulation of U12.

4. POUT(11)=POUT(11)+SDA(9)
This statement adds the cost of housing test equipment and shipping of mod kits (SDA(9)) with "Other Indirect" TOE costs. SDA(9) is computed in LGCST and is the case total accumulation of CSHTF+CFT.

5. POUT(12)=CGT
This statement stores the system maintenance support case total cost CGT from LOGAM into operation and support output array. CGT was computed in Subroutine LGCST.

6.
```
   POUT(13)=0
   DO 20 I = 1,12
20 POUT(13)=POUT(13)+P(I)
```
These statements accumulate the TOE and LOGAM maintenance outputs into a case total for System Operations and Support.

7. IF(IBUG.EQ.1) WRITE (6,BUGSEV)
This statement will print the post processor output array (POUT) when debugging of a data case is requested.

8.
```
RETURN
END
```
These statements terminate the execution of Subroutine SEVEN and returns logic to Subroutine OPER.

4.8.9 SUBROUTINE SUPTMH

This subroutine accumulates annual maintenance manhour and positions for TOE support personnel. The input to SUPTMH is a data file (8) that is sorted by MOS from the AR570-2 MACRIT data base. This data base is maintained by the U.S. Logistics Center, Fort Lee, VA. The sorted input file is created with a program written by the System Planning Corporation (SPC) for support to the LOGAM program.

The SPC program searches the AR570-2 data tape for line items input and locates all MOSs associated with a line item. File 8 is written with the manhours and positions for each MOS. Subroutine SUPTMH reads file 8 and sums the manhours and positions for all identical MOS entries.

The accumulated MOS data (SUM) is used in Subroutine SGAPER to determine the annual support personnel by grade. SUPTMH is referenced from Subroutine OPER. Two conditions must be met before SUPTMH is entered. First, TOE evaluations must be selected by inputting the flag IOPER=1. Second, IMF must be input as a positive non zero value. IMF is the file position of tape 8 containing maintenance data for a particular theater or deployment.

A. This section of SUPTMH contains the subroutine name and its arguments and the assignment of variables in central memory.

1. SUBROUTINE SUPTMH(IMF,POS)
   This statement is the entry point to the subroutine. SUPTMH is referenced from subroutine OPER. The arguments IMF and POS are described as:

   IMF — File position of tape 8 containing the maintenance support data for the theater being evaluated.

   POS — Array for storing maintenance positions per grade per supply level.

2. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains a list of all of the tape and disc files used by LOGAM. Only K004FX (output print file) and K8 (MACRIT data file) are used in the subroutine. These values are set in the block data (BLKDAT) routine.

3. DIMENSION SUM(6),TOT(6),XX(6),ITITLE(8),
   NOM(4),POS(3,13)
   The dimensioned variables for SUPTMH are described as:

SUM(6) An array to accumulate annual maintenance manhours and positions at Organization, Direct Support, and General Support for identical MOS entries.

TOT(6) Accumulated manhours and positions for all MOSs.

XX(6) Manhours and positions for one MOS entry read from the data file.

ITITLE(8) Eighty character description of the data case stored on file 8. This description is the first record on the data file.

NOM(4) Forty character description of a line item's nomenclature.

POS(3,13) Maintenance positions at 3 levels (E, DS, GS) for 13 personnel grades.

4. DATA IBL/3H /
This statement sets IBL to a blank field. IBL is used to reset the MOS description to a blank after all MOSs of one type have been processed.

B. This section of SUPTMH is the initialization code.

1. IF (IMF.LE.0) RETURN
This statement returns logic to the calling routine if the file position on the MACRIT data file is not specified.

2. REWIND K8
This statement readies the MACRIT data file for reading.

3. ICOUNT=0
INDF=0
These statements initialize the output page line counter and the end of the file flag for K8.

4.
```
  DO 5 I=1,3
  DO 5 K=1,13
5 POS (I,K)=0.0
```
These statements initialize the maintenance support array that accumulates positions by grade for three maintenance levels.

5.
```
  DO 20 I=1,IMF
```

```
      IF(I.EQ.IMF) GO TO 30
   10 READ (K8)
      IF(EOF(K8)) 20,10
   20 CONTINUE
```

These statements move the MACRIT data file to the correct position for reading the information specified by IMF.

6. `30 READ (K8,40) ITITLE,CAT`

This statement reads the first record of the file IMF of K8. ITITLE is an eighty character data case description and CAT is the annual productivity category code.

7.
```
      IF(EOF(K8)) 50,70
   50 CONTINUE
      WRITE(K004FX,60) IMF
      RETURN
   70 CONTINUE
```

These statements check for an end of file on the first read of the MACRIT data file. If an end of file is found, a void data file message is printed and logic is returned to the calling subroutine (OPER). Otherwise a transfer is made to statement 70 to continue reading the data file.

8.
```
      DO 80 I=1,6
      TOT(I)=0.
   80 SUM(I)=0.
      MOSS=IBL
```

These statements initialize the arrays for accumulating manhours and positions for all MOSs (TOT) and identical MOSs (SUM). The MOS description (MOSS) is initialized to a blank field.

9. `CALL PAGE`

This statement prints a page header before printing maintenance manhours and positions.

10. `WRITE(K004FX,90) ITITLE,CAT`

This statement prints the case title and productivity category.

C. This section of SUPTMH reads the MACRIT data file sorted by MOS and accumulates maintenance support manhours and positions for identical MOS entries. After all entries for an MOS are processed, SGAPER is called to convert the maintenance positions accumulated for this MOS into personnel positions per grade. The MOS data accumulations (SUM) are for manhours and positions at the Organization, Direct Support, and General Support maintenance levels.

1. 100 READ(K8,110) MOS,DEN,LINE,(XX(I),I=1,6),NOM
This statement reads a record from the MACRIT data file.

MOS 3-5 character code that has been sorted numerically and alphabetically.
DEN The number of items per line number.
LINE 6 character line item description.

Annual maintenance support manhours at:
XX(1) Organization
XX(3) Direct Support
XX(5) General Support

Annual maintenance support positions at:
XX(2) Organization
XX(4) Direct Support
XX(6) General Support

NOM 40 character description of the line item nomenclature.

2. IF(EOF(K8))115,120
This statement checks for end of information on the MACRIT data file. If all information has been read, a transfer is made to statement 115 where an end of file flag is set. Otherwise the program transfers to statement 120 to process the MOS data.

3. 115 INDF=1
GO TO 125
When end of information is reached on the MACRIT file, INDF is turned on. INDF is used to terminate execution in SUPTMH once the last set of MOS data are processed.

4. 120 CONTINUE
IF(MOSS.EQ.IBL) GO TO 160
IF(MOSS.EQ.MOS) GO TO 160
These statements check the MOS to determine if the MOS for the last record read from the file is the same as the MOS from the previous read. The code that accumulates data having the same MOS begins with statement 160. On the first read MOSs is equal to IBL blank characters, therefore a transfer is made to statement 160 to store the data into the SUM array. On succeeding reads MOSs will have the same value as the MOSs from the previous read. When the MOS differs from the previous data record the data from the previous accumulations are printed and accumulated into a system total (TOT) in the following statements.

5.
```
125 CONTINUE
    ICOUNT=ICOUNT+1
    IF(ICOUNT.LE.40)GO TO 130
    ICOUNT=0
    CALL PAGE
    WRITE(K004FX,90)ITITLE,ICAT
130 WRITE(K004FX,140)MOSS,SUM
```
These statements print the annual manhours and positions (SUM) accumulated from the MACRIT data file for all entries having the same MOS. There is a limit of 40 MOS lines printed per page. On the first print and subsequent prints when the number of lines exceed 40, subroutine PAGE is called to print a page header. The MACRIT data file title and the productivity category flag (ICAT) are printed with each page header.

6.
```
CALL SGAPER(SUM,POS)
```
This statement calls subroutine SGAPER to equate the maintenance positions SUM(2),SUM(4) and SUM(6) to maintenance positions by grade (POS). Positions are computed for 13 grades at 3 maintenance levels. All positions above E6 are considered supervisory positions.

7.
```
    DO 150 I=1,6
    TOT(I)=TOT(I)+SUM(I)
150 SUM(I)=0
```
These statements are executed when the MOS for the last record read from the MACRIT data file is different from the MOS of the previous read (MOS ≠ MOSs). The annual manhours and positions (SUM) for MOS are accumulated (TOT) with other MOSs and the SUM array is reinitialized for the next MOS accumulation.

8.
```
160 DO 170 I =1,6
170 SUM(I)=SUM(I)+XX(I)
    MOSS=MOS
    GO TO 100
```
These statements accumulate the annual manhours and positions (SUM) for all line items having the same MOS. XX(1), XX(3), XX(5) are the annual maintenance manhours and XX(2), XX(4), XX(6) are the annual maintenance positions for Organization, Direct Support, and General Support, respectively. The last MOS read from the data file is saved (MOSs) and the program transfers to statement 100 to read the next data entry.

9.
```
180 CONTINUE
    WRITE(K004FX,190) TOT
    RETURN
    END
```
These statements terminate execution of SUPTMH after

printing the totals for annual maintenance manhours and positions.

4.8.10 SUBROUTINE SGAPER

This subroutine uses a standard of grades authorization (SGA) table to equate maintenance support positions from the AR570-2 MACRIT data file to personnel numbers by grade. A "typical" SGA table is used to evaluate the personnel count for all MOSs. The table used is for MOS 35B, an electrical instrument repairer. Since the use of a SGA table for each MOS would be too massive of a data base to input to LOGAM, the 35B table was used to simplify the problem and still give a "ball park" result. SGAPER has the capability to evaluate personnel for 13 grades including 7 enlisted and 6 officer grades. Presently, there are only 5 levels of data entered into the SGA table. The first four levels are for the grades E3 through E6 which are personnel that are charged to maintenance. The fifth grade is for an E7 which is a supervisory position that is charged to overhead. The remaining grades are not included on the SGA table at this time, but if they were they would also be supervisory positions.

The personnel positions (POS) evaluated are at one maintenance location for each of three maintenance levels (E, DS, GS). The maintenance positions (SUM) accumulated for a MOS are used as an index to the SGA array to determine the number of personnel in each grade that equates to the MOS positions. Only the grades E3 through E6 which are charged to maintenance will sum up to the number of positions by MOS.

For example, the number of positions at DS is computed to be 3.39 (SUM (4) = 3.39). Since SGAPER rounds up to the next higher position, the personnel count from the SGA table would be SGA (4,1) = 1 E3, SGA (4,2) = 2 E4, SGA (4,3) = 1 E5 and SGA (4,4) = 0 E6 totaling 4 positions. The supervisory position for E7 would be zero. A list of the SGA inputs is listed in section (A-3) of this subroutine description. SGAPER is called from Subroutine SUPTMH each time an accumulation of maintenance positions for an MOS is completed. After all MOSs have been processed, POS will have the accumulation of personnel for each of 13 grades at the three maintenance levels.

A. This section of SGAPER includes the subroutine name, its arguments, allocation of variables in computer memory and the SGA table.

1. SUBROUTINE SGAPER (SUM, POS)
   This is the entry point for the subroutine where;

   SUM is the maintenance manhour positions accumulated for an MOS.

   POS is the personnel count by grade that equates to the number of maintenance positions.

2. DIMENSION SGA (10,13), SUM (6), POS (3,13)
This statement sets the size of the arrays where:

SGA is the standard of grades authorization table for 10 positions and 13 grades.

SUM is the accumulated maintenance manhours and positions for an MOS. Indicies 1, 3, 5 are the manhours and 2, 4, 6 are the positions.

POS is the positions by grade for 3 maintenance levels and 13 grades.

3. DATA SGA/ 0., 1., 1., 1., 2., 2., 2., 3., 4., 4.,
1., 1., 2., 2., 2., 2., 2., 2., 2., 3.,
0., 0., 0., 1., 1., 1., 2., 2., 2., 2.,
0., 0., 0., 0., 0., 1., 1., 1., 1., 1.,
0., 0., 0., 0., 1., 1., 1., 1., 1., 2.,
80 * 0./
This statement stores the number of personnel by grade for the standard of grades authorization table. The grades for E3 through E7 are the first 5 lines. The sixth line initializes the positions for all other grades to zero.

B. The next section of SGAPER builds the personnel count array (POS). Each of the 10 columns for the first 4 rows of the SGA table sum to equate with the number of positions extracted from the MACRIT data file. When the number of positions exceeds 10, the tenth position (or a multiple thereof) of a SGA grade is added to the position occupying the remainder (number of positions modulo 10) location. This section will continue to accumulate positions by grade at three maintenance levels until all MOSs on the MACRIT data file have been processed in Subroutine SUPTMH.

1. DO 20 I = 1,3
This statement loops through the logic once for each maintenance level.

2. IK = 2 * I
This statement computes the index of the SUM array where maintenance positions are stored. Positions are stored in locations 2, 4, and 6. Manhours are stored in locations 1, 2, and 3 but are not used in this program except for printing in Subroutine SUPTMH.

3. NP = SUM(IK)+.999
This statement rounds the maintenance position to the next higher position.

4. IF (NP.LE.0) GO TO 20
This statement skips the accumulation logic when there are not any positions assigned to a maintenance level.

5. IP = NP
IF (IP.GT.10) IP=MOD(NP,10)
These statements compute the column index for the SGA array where the positions by grade will be extracted. When IP is greater than the column array size (10), IP is computed as the remainder after dividing by 10.

6. NRPT=(NP-1)/10
This statement computes the repeat factor for the number of times the maintenance positions exceed the maximum column index (10) of SGA. This term along with IP will compute the number of personnel from SGA to equate with the number of authorized positions. For example, if the number of positions is 13 then the positions for an E4 is SGA(10,2)=3 plus SGA(3,2)=2 which gives a total of 5 E4s.

7. DO 10 K=1,13
10 POS(I,K)= POS(I,K)+NRPT*SGA(10,K)+SGA(IP,K)
These statements accumulate the personnel for 13 grades that equate to the number of authorized maintenance positions for an MOS. Accumulations are stored for three levels (I=1,3) of maintenance (E,DS,GS).

8. 20 CONTINUE
This statement is the end of the loop for accumulations by maintenance level. When I=3 the logic for this subroutine will be terminated.

9. RETURN
END
These statements terminate the execution of SGAPER and returns to Subroutine SUPTMH where the maintenance positions for another MOS is accumulated.

4.8.11 SUBROUTINE TOEPER

This subroutine constructs the TOE namelist data entries for personnel related to maintenance support positions. The personnel positions by grade that are determined from the MACRIT data file and the SGA table are stored into locations provided by the user when he inputs the NAMELIST TOE data. The maintenance positions (POS) determined in Subroutine SGAPER are for one maintenance location at Equipment, Direct Support, and General Support. These positions are multiplied by the number of maintenance locations (ED,OD,DI) to get the total personnel count for maintenance support before storing into the TOE input array (T).

The user must input certain fields in the TOE inputs before the data from this subroutine can be stored. Table 5 of section 3 in the programmer's manual (VOL III) gives a description of the TOE namelist fields. The fields that must be input are columns I, II,III, IV, and X. A one in column I indicates that the inputs are for personnel and column II must be zero indicating that the quantity of personnel is to be stored in this location. Column III indicates either officer or enlisted grades. A zero denotes the grade is for an officer and a one is for enlisted. Column IV and column X are the personnel grade and cost per man year, respectively. All other fields are input as zeros.

The user can override any personnel grade determined by the SGA table by simply not entering that grade (column IV) into the TOE data. The labor category fields for the TOE data are stored by this subroutine. For grades E6 and lower, a one is stored in column IX indicating maintenance personnel. For grades E7 and above, a one is stored in column V indicating overhead personnel.

Each TOE data item consists of a row of 10 input fields. The column numbers described above are one of these 10 fields. The row number in this subroutine is referred to as J; where each grade input will have its own J number. TOEPER is called by Subroutine OPER for each data item (J number) input by TOE when the first entry (column I) and the second entry (column II) are a one and a zero, respectively.

For a description of the TOE input logic refer to the description for Subroutine OPER.

A. This section of TOEPER includes the subroutine entry point, the arguments and the assignment of variables in computer memory.

1. SUBROUTINE TOEPER (J,POS)
   This is the subroutine entry point where:

   J is the row number or data item number in the order that the data item is input to NAMELIST

TOE.

POS is the personnel positions by grade for one location at each of these maintenance locations (E, DS, GS).

2. COMMON /INPUT/
This common block contains the variable list for the LOGAM inputs of NAMELIST/L/. The list is not included here but can be found in the description for LOGAM.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAT(4),
TAYZ(10),ZM(4),ZP(3),ZU(4),SENSY(266),
T(10,200)
This statement contains the list of all the dimensioned variables of common block INPUT.

4. DIMENSION POS(3,13)
This statement dimensions the variable that stores personnel positions at 3 maintenance levels for 13 grades. POS is passed through the calling argument of TOEPER when referenced from Subroutine OPER. The contents of POS were evaluated and stored in Subroutine SGAPER.

B. The next section of TOEPER takes the grade level from the TOE data item to find the number of positions for that grade as stored in POS. The number of positions at each maintenance level are then multiplied by the number of maintenance locations (ED,OD,DI) to get the total personnel count. The quantity of personnel along with the appropriate labor category flags are stored into the TOE input array.

1. GRADE=T(4,J)
This statement sets the personnel grade for line item J of the TOE inputs.

2. IF(T(3,J).EQ.0.0) GO TO 10
This statement will transfer to statement number 10 if the personnel grade for line item J is for an officer.

3. IG=(GRADE-2.0)+.5
This statement computes the index for the POS array where maintenance positions are stored for enlisted grades. The grade number is reduced by 2 because the first entry in POS is for an E3.

4. IF(IG.LT.5) T(9,J)=1.0
This statement sets the labor category flag for maintenance personnel when the grade is on E6 or lower.

5. GO TO 20
This statement will bypass the logic for storing personnel data for officers.

6. 10 CONTINUE
IG=(GRADE+7.0)+.51
These statements compute the index for the POS array where maintenance positions are stored for officer grades. The first officer grade is a warrant officer which is denoted in the TOE inputs as 0.5. Officer data are stored in POS beginning with the eighth position.

7. 20 CONTINUE
This statement begins the logic for constructing the TOE inputs with the MACRIT position data.

8. IF(IG.GE.5) T(5,J)=1.0
This statement sets the labor category flag for overhead personnel when the grade is an E7 or above.

9. T(2,J)=ED*POS(1,IG)+OD*POS(2,IG)+DI*POS(3,IG)
This statement computes the total personnel count for a grade level.

10. RETURN
END
These statements end the execution of TOEPER and return to Subroutine OPER for the next line item.

4.9 LOGAM Input Routines

The subroutines described in this section are used to Fault Isolate Namelist Errors (FINLE) and to store the Namelist values into the INPUT common block. These routines replace the system NAMELIST. Instead of the variable list being in a NAMELIST/L/ statement the variable names are stored in a DATA statement of Subroutine FINLE. There are 2 modes in which FINLE is selected. One mode (IRUN=0) is to preprocess all Namelist cases for errors without executing LOGAM. The second mode (IRUN=1) will process one Namelist case at a time for errors and return to LOGAM to execute the case.

The subroutines that make up the FINLE program are:

1. FINLE — Driver subroutine for fault isolating errors and storing input values.
2. READFL — Reads a data record from the input medium.
3. IMPAK — Packs portions of a data record into an array for parsing of legitimate characters.
4. PARSV — Parses a section of an input record for valid inputs.
5. DCODR — Used in conjuction with PARSV to isolate errors and to decode the input characters into numeric values.
6. VALUE — Used by DCODR to decode the input characters into either a real or integer value.
7. MXINP — Checks the number of inputs for a variable to prevent storing more values than the variable is dimensioned for.
8. COLINE — Prints a line of column counters and the input record when an error is found.

4.9.1 SUBROUTINE FINLE

This subroutine is the driver for Fault Isolation of NAMELIST ERRORS (FINLE). FINLE simulates the function normally provided by the NAMELIST section of the FORTRAN compiler. The advantage of FINLE over NAMELIST is that FINLE will process all input records; whereas, NAMELIST will terminate with the data case containing an error. There are some limitations to FINLE that may be available in NAMELIST. First, only a single subscript can be part of the variable name input. This does not mean that inputs to a double dimensioned array can not be made, but that the reference to the variable on the Namelist record must be a single subscript. Other limitations are the use of octal, double precision and complex numbers.

With the exception of the limitations mentioned above, the inputs to FINLE are the same as for the FORTRAN NAMELIST function. Each data case starts with a "$NAME" followed by the inputs separated with commas and ending with a "$". The first character of each input record must also be a blank character. "NAME" in $NAME is an optional name that user assigns to his Namelist data. The name the user assigns must be set in a DATA statement in FINLE. For example if "L" is the Namelist name then "DATA NLIST(1)/1HL/" must be coded before the first executable statement of FINLE. The use of NLIST and other program variables will be discussed later.

There are 2 modes of operation for FINLE. One mode will process the input records (cards) for errors and terminate after the last record is read. With this mode (IRUN=0) all input records are read by FINLE including comments. The input record image is printed and if the record is a comment (not a Namelist input) a warning message is output. If the input is a Namelist record and an error is found, a diagnostic is output describing the nature of the error. With the second mode (IRUN=1) only one case at a time is processed. FINLE will read input records until a valid Namelist case is found. Any comment cards or errors will be output as they are in the first mode. With mode 2, the input values are stored into appropriate locations of common block INPUT and a return is made to LOGAM to execute the case. Some input errors are recoverable by FINLE. Imbedded blanks in data are illegal by NAMELIST standards but FINLE will pack the input characters and store the value in INPUT. For those errors that are non-recoverable (illegal variable name, non-numeric character in value, etc.) the INPUT value will either be the default value (BLKDAT) or the value input from a previous case. Comment cards in mode two are read from the main program (LOGAM). There must be at least the number of comment records in the input stream described in section 5 of this document for mode 2 to work properly. Any extraneous comment card outside the Namelist case (not between $NAME and $ terminating the case) will not effect the execution of mode 2.

The Namelist variable names are set in a Namelist Array (NLA) using the FORTRAN DATA statement. The order of these names in NLA must be in the exact order as the variable list of COMMON/INPUT/. FINLE uses the relative location of these variables in the NLA list for storing the input values into the common block (INPUT). The names on the input records must match the names in NLA. The order of the names on the input record are arbitrary; they will have no effect on the processing of the data. Any name in the input stream that does not match an NLA name will be flagged as an error.

FINLE reads input records until a "$" character is found. When finding a "$" a determination is made for any subsequent characters which may comprise of a valid Namelist name. If the characters match a name of NLIST (DATA statement) the logic for processing Namelist variables will be entered. When a match in Namelist names is not made the input record will be treated as a comment card and all subsequent input records will also be assumed to be comments until a match is made.

Namelist variable names (NLA) are compared to the input names. When a legal name is found the input value for that name is passed for errors and the value stored into COMMON/INPUT/ unless the error is a non-recoverable error. When an illegal name is input, the parsing of it's input value is skipped and a search is made for the next input name. The search for variable names and parsing of input values will continue until finding a "$" character. Under mode 2 the "$" will return logic to LOGAM to execute the data case; otherwise, FINLE will search for another Namelist case.

A. The following section describes the input variables, output variables and internal variables used by FINLE.

1. SUBROUTINE FINLE(IRUN,IRD5,IRD6,IEND)
   This statement includes the input and output arguments for FINLE where;

   IRUN is the run mode option. The value for IRUN is read from column 70 of the eighth input card. The eighth card is one of the one-time inputs read in the initialization section of LOGAM. Possible values are:
   IRUN=0, preprocess for Namelist errors only.
   IRUN=1, preprocess each Namelist case and execute LOGAM.

   IRD5 Input file containing the Namelist records. Same as K001FX in LOGAM.

IRD6 Output print file. Same as K004FX in LOGAM.

IEND Flag set in Subroutine READFL when an end of file is reached on IRD5. Used in LOGAM to terminate program execution.

2. COMMON/INPUT/
This common block contains the list of NAMELIST variables recognized by FINLE. The list of variables here are in the exact order as the list in DATA NLA.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
TAT(4),TAYZ(10),ZM(4),ZP(3),
ZU(4),SENSY(266),T(2000)
This statement sets the array size for all dimensioned variables in COMMON/INPUT/. In order for FINLE to work properly the array sizes are stored in the NLA array along with the input variable names. NLA is discussed below.

4. DIMENSION NLIST(2),NLVAR(2),IVAL(80)
DIMENSION NLA(280,3),A(80)
DIMENSION SAV(2592),ISAV(2592)
EQUIVALENCE (SAV,ISAV),(ARA,SAV)
These statements contain the arrays required by the FINLE program. NLIST, NLVAR and NLA are used to store Namelist names, number of variable names per Namelist and the variable names in ASCII code. These arrays are discussed in the following DATA statements. Arrays "A" and IVAL are used to store one ASCII character per word for a maximum input record of 80 characters. "A" contains the record as it exists on the input medium and IVAL contains segments of "A" that are being parsed for either names or input values. The contents of IVAL are set in Subroutine IMPAK where any blank characters will be eliminated and the non blank characters packed. SAV and ISAV are dimensioned to the number of values that can be input to COMMON/INPUT/. These variables are equivalenced to INPUT whereby either real (SAV) or integer (ISAV) values can be read by FINLE and stored into the common block. ARA is the first variable name in COMMON/INPUT/.

5. DATA IBLN/6H /
DATA NLIST/6HL , /
NLIST contains the ASCII coded characters for the Permissable Namelist names. Presently only NAMELIST "L" is coded. The second field of NLIST is blank coded. When the "$" character is read from an input record, the contigious non-blank characters following the $ are stored into a variable called NAME. NAME is then compared with NLIST to determine if there is a match of

a Namelist name. IBLN is used to reinitialize NAME prior to storing of characters from the input record.

6. DATA NLVAR/277,0/
DATA NNAM/1/
NLVAR is the number of variable names for the Namelist names of NLIST. For NLIST(1) there are 277 variable names coded in the NLA array. These 277 names correspond to the 277 names in the COMMON/INPUT/ list. When $L is located on an input record, any of these 277 names are valid inputs. NNAM is used in a "DO-LOOP" to search for the valid Namelist names. Currently this is one since NLIST(1)="L" is the only valid name coded in the DATA statement.

7. DATA (NLA(1,I),I=1,3)/6HARA ,0 ,0/
DATA (NLA(135,I),I=1,3)/6HG ,20,0/
DATA (NLA(150,I),I=1,3)/6HNU ,0,1/
DATA (NLA(277,I),I=1,3)/6HT ,2000,0/
These statements are a partial list of the DATA statements that contain the Namelist "L" variable names. These particular statements were selected to illustrate the first and last element names and the various options for a NAMELIST. The variable names can be either simple variables or dimensioned and they can be either real (floating point) or integer. The first DATA statement for variable ARA is a simple floating point variable. The second and third parameters in the list are "0s" signifying simple and floating point, respectively. A positive non-zero value as the second item in the list instructs FINLE that the variable name in the list is dimensioned. Variable "G" for NLA(135) is a dimensioned variable where "20" is the maximum input size. An attempt to input more than 20 values for "G" will result in an error diagnostic being printed by FINLE. The third DATA element NLA(150) has a "1" in position three which indicates that the variable NU is of type integer. The fourth DATA element is the last variable name (T) in the list (NLA) and corresponds to the last variable of COMMON/INPUT/. "T" is the TOE O&S data array that is dimensioned by 2000 words.

B. There are Five major sections to Subroutine FINLE. Each of these sections will be discussed in general terms but not down to the line by line code. The sections to be discussed are the (1) data record inputs, (2) NAMELIST name search, (3) variable name search, (4) parsing for input values and (5) storing input values into COMMON/INPUT. Intermingled within these sections are the warning and error diagnostics.

1. Data Record Inputs - Statement "10" of FINLE begins the

input section. Once the data for an input record has been processed a return is made to "10" to read the next record. Under mode 1 an attempt to read the next record will be made. If an end-of-file is found on the input medium while being read in Subroutine READFL, IEND=1 is set where program termination will occur after returning back to LOGAM. When running under mode 2, the next data record will be read unless the last record processed was terminated with a "$" character. A "$" found after a valid Namelist name sets IDOL=0 which with IRUN=1 (mode 2) sends logic back to LOGAM where the data case will be executed.

It is in this section that the test for a Namelist name or the end of a Namelist case is made by checking for a "$" on the input record. A "$" causes a transfer to statement 30 where the Namelist name processing will begin. Once a Namelist name ($NAME) is found, IDOL=1 is set and any input record preceding the next "$" character will be parsed for input variable names beginning with statement 25. "NAME" of $NAME is a generic name and is not intended as a reference to a valid Namelist input name.

2. Namelist Name Search - Each input record is checked for a "$" character. Once a "$" is found, a transfer is made to statement 30 where any contigious non-blank characters are coded into NAME. Any non-blank value for NAME will be compared to the valid Namelist names of the NLIST array. An invalid NAME results in an error diagnostic output and a transfer back to the input section to read the next data record. If the "$" is followed by a blank character (this condition normally terminates a Namelist case) a test for IDOL=1 is made to determine if a valid Namelist name was input on an earlier record. With IDOL=1, any data preceding the "$" will be parsed for a variable name (NLA array) beginning at statement 210.

A Namelist name must be preceded by all blank characters on the input record. If the condition exists where $NAME is preceded by a non-blank character, the data record is assumed to be a comment and a transfer is made to statement 10 to read the next record. Once a valid Namelist name is found, IDOL=1 is set and the indices for the first and last variable name of NLA are computed. The indicies for the variables of $NAME are computed as IV1 and IV2. IV1 is the word position of NLA where the first variable is located and IV2 is the position for the last variable. A transfer is made to statement 210 where a data record is parsed for variable names and a comparison made with

the NLA names.

3. Variable Name Search - This secion of code is entered after a valid Namelist name is found. Multiple inputs can be entered on the same input record. The rule is that a variable name and it's value(s) is separated by the "=" character and each variable name separated from the previous input with a comma. Any characters preceding the "=" are encoded into VARN for comparison with the NLA names. If the first character of VARN is not alphabetic, or a match with NLA was not made, an error diagnostic is output and a search for another variable name on the input record is made.

   Several things occur when a valid variable name is found before the value for the name can be parsed. The initial position (IPOS) this variable shares in COMMON/INPUT/ must be determined. IPOS is used to store the input value into the SAV(ISAV) array. For example if the variable "H" of NLA(136) is the input variable then IPOS equals 154 for the initial position of "H". NLA(135) is "G" which is dimensioned by 20. Since "G" is the first dimensioned variable of the list, IPOS=135+20-1=154. As each search is made through the NLA table the positions are accumulated by adding 1 or the dimension size less one (which ever is larger). The dimension size for a variable is stored in NLA(I,2) where "I" is the array index containing the variable name.

   The type of variable name is determined by the third element of an NLA entry. A zero (NLA(I,3)=0) in the third element denotes a real variable and a one denotes an integer variable. The type of variable is set as ITYPE=0 or ITYPE=1. ITYPE is used when calling the VALUE subroutine to decode the input value. After setting ITYPE the variable name is checked for the presence of a subscript.

   A dimensioned variable may be referenced with or without subscripts. Beginning at statement 340 there is logic which determines if the variable is referenced as a subscripted variable. If a left parenthesis is the next character following the name, ILSUB is set to it's character position on the input record. A search is made for a matching right parenthesis. If the right parenthesis is found and there are only numeric characters between the parentheis, a transfer is made to the section to parse input values. Any other condition in the subscript will result in an error diagnositics output and a transfer to logic for finding the next variable name (statement 220).

4. Parse Input Values - Once the determination is made that the variable name is valid the logic beginning with statement 410 is entered and the values are parsed for errors. The code on an input record that represents the input value(s) is moved into the IVAL array and packed if there are blank characters in the data. Subroutine IMPAK is called to store IVAL and remove any blanks. In the system NAMELIST "embedded" blanks are illegal and will cause a program termination. IMPAK will print an error diagnositc and continue execution after the blanks are removed (packed).

Subroutine PARSV is called with the IVAL data to parse the input values for errors and to DECODE the ASCII code into either real or integer values. If there is more than one data item input for a variable (i.e., H=1,1,1,0,), PARSV will be called once for each item. A pointer is computed for IVAL on each of the calls to IVAL. The comma seperating the data items dictates what the pointer will be. PARSV will check each data item for illegal characters. Normally any character that is not numeric, a plus (+) sign, a minus sign (-) or a decimal point (.) will be illegal. There are exceptions to this in this program. When used in the right context the letter "E" and the asterisk (*) are legal characters.

The asterisk is used as a repeat factor. For example if H=4*1 were input, PARSV would first decode the "4" as the repeat factor and the "1" as the input value. FINLE would then store (using a DO LOOP) four values of H into the SAV array beginning at the relative location of H(1) in SAV. The asterisk must be preceded with an integer number and followed by any of the legal characters mentioned above (including "E"). A repeat factor greater than the dimensioned size of the variable will result in an error diagnostic when Subroutine MXINP is called later. Only the valid number of inputs are stored into a variable's relative location in SAV. Any remaining data items for that variable are parsed for errors but will not be stored.

Data can be input using scientific notation (E). There can be several forms used. For example, 1E05,E5,1.E+5,and +1E5 are all legal and will give the same results if input for a real variable. The absence of a numeric value (E5) preceding the E assumes a positive 1. The "E" notation can be used to input values for integer variables but the value must not contain a decimal point. PARSV parses the data item for an "E". When an "E" is found the characters

preceding the E are checked for illegal characters. Legal characters are decoded and stored in PDE as the product factor. The exponent portion is parsed similarly and decoded into EX as the exponent portion. Before leaving PARSV the value is computed by V=PDE*10**EX.

5. Store Input Values - Arrays SAV and ISAV share the same computer memory locations. They also share the same locations as the variable list of COMMON/INPUT/. To store the input values into the correct locations the variable names coded in the NLA array and the names in the COMMON/INPUT/ list must be in the same order. Also the dimension sizes set in NLA; i.e., NLA (135,2)=20, must be the same as the dimensioned variable (G(20)) in COMMON/INPUT/.

   The decoded values are stored into the COMMON/INPUT/ memory locations using the relative storage position (IPOS) as the initial address. When a subscripted variable is input the initial address is modified (IP) by the subscript (ISUB). The first input value is stored in the modified address and if there is more than one value the address modifier is incremented (NINP=NINP+1) for each additional value. The address is modified (IP=IPOS+NINP) in a "DO LOOP" using NRPEAT as the repeat counter. NRPEAT=1 is set in PARSV and remains this value unless a repeat factor (*) is input with values for a dimensioned variable. For variable array inputs where the repeat factor is not used, a pass is made through this section of code for each value input. Each value seperated by a comma is parsed for errors in PARSV and decoded. Since the relative location (IPOS) and the address modifier (NINP) are not reset until a new variable name is found, the decoded value will be stored in the next (NINP+1) consecutive location of /INPUT/. After all input values for an input variable are processed, there is a return to statement 220 where a search of the input record is made for a new variable name.

   The input values are decoded in Subroutine VALUE as either FVAR or IVAR. When ITYPE=0 the value is for a real variable and the value is decoded into FVAR. ITYPE=1 flags an integer variable and the value is decoded into IVAR. This same logic is used in FINLE to store the decoded values into either the real array (SAV) or the integer array (ISAV). Even though SAV and ISAV share the same memory locations real and integer values must be stored according to their type in order to prevent type conversion by the FORTRAN compiler.

C. Following is a list of the subroutines used by the FINLE program and a brief description of each.

4.9.2. SUBROUTINE READFL.

This subroutine is called from FINLE to read a data record from input medium IRD5. Eighty characters are read and stored into array "A" with one character per word (80A1 format). After the data in "A" is processed a return is made to statement 10 in FINLE to read the next record unless an end of file was encounter on the last read. IEND=1 is set in READFL when reading an end of file and FINLE will transfer to statement 900 for a return to LOGAM.

4.9.3. SUBROUTINE IMPAK.

This subroutine is called from FINLE with segments of the input record (A) to pack the non-blank characters into IVAL. IMPAK checks for imbedded blanks and prints a diagnostic when encountering one. An embedded blank is one that occurs betwen characters of a variable name or between characters in an input value. Other blanks are not illegal but they will be removed by IMPAK.

4.9.4. SUBROUTINE PARSV.

This subroutine is called from FINLE to parse an input value for illegal characters. PARSV checks the ASCII characters of IVAL for the presence of an asterisk (*) or an "E". If either of these are found, DCODR is called to check if there are illegal characters associated with these input options.

An asterisk denotes that a repeat factor is being used. Any character preceding the "*" except for numerics will be flagged in DCODR with either IDEC, ISGN or IERR. A non-zero value for either of these will cause an error diagnostic. The parsing of the input value portion of the term will continue but the repeat factor (NRPEAT) is set to zero. With NRPEAT=0 there will not be an attempt to store data into COMMON/INPUT/ when returning to FINLE.

An "E" denotes that scientific notation may be used. When an "E" is found DCODR is called once to find illegal characters in the product factor (characters preceding the E) and once to find illegal characters in the exponent factor (characters following the E). Any of the characters that are legal in a simple value (numeric, signs, decimal) are legal for both the product and exponent factor. The product factor (PDE) is parsed first and if an error is found a diagnostic is printed and the parsing will continue for the exponent factor (EX). If both PDE and EX are valid inputs then the input value returned to FINLE will be computed as FVAR=PDE*10**EX when the value is a real (IDEC=1)

number. If a decimal point was not input (IDEC=0) the value is computed and returned to FINLE as IVAR=FVAR+.00001, an integer value.

When neither an "*" nor an "E" is part of the input code, PARSV assumes that the input is a simple value where only numerics, signs and the decimal point are legal. If the constraints for the input characters are not violated the value will be decoded in Subroutine DCODR and returned to FINLE as either FVAR or IVAR. If ITYPE=0 the value is decoded as real (IVAR) and with ITYPE=1 the value is decoded as integer (IVAR). This part of PARSV is also used to decode the input value portion when a repeat factor is being used. For example if the input is OL=4*15, the "15" is the input value and "4" is the repeat factor. The "15" would be parsed and decoded in this section of the code the same as if OL=15,15,15,15.

In the example above where "OL=15,15,15,15" PARSV will be called from FINLE four times, once for each of the values input. FINLE recognizes the commas as delimiters and will compute pointers for PARSV to use in parsing the IVAL array. The contents of the IVAL array for this example is "15,15,15,15". The pointers (I1,I2) are 1 and 2 on the first pass, 4 and 5 on the second and similarily computed on the remaining passes. PARSV uses I1 and I2 as the first and last character positions of IVAL where an input value will be parsed and decoded. When entering PARSV, IVAL will contain all the non-blank characters found on input record between an "=" sign and the next variable name or the end of the record (whichever occurs first). These characters will include all the input values for a variable name.

4.9.5. SUBROUTINE DCODR.

This subroutine is called from PARSV to check for illegal characters in an input value. The only valid characters are the numeric characters, the decimal, the plus sign and the negative sign. The first character is checked for an arithmetic sign since these characters are legal in most of the inputs. The parameter ISGN is set to a negative one or a positive one depending if the character is a "+" or a "-". Either of these characters will set an error flag (IERR=I) if encountered in any character position except the first. If a non-numeric character is found after the first position, except for the decimal, the error flag will be set and a diagnostic printed upon returning to PARSV. More than one decimal point in the value will also set the error flag. After returning to PARSV a determination is made if the characters assummed legal in DCODR are still legal when used in the context of PARSV. The sign (ISGN) and the decimal (IDEC) are the characters that may be in error in PARSV. Unless an error is found in DCODR the input characters are decoded into either a real or integer value by calling Subroutine VALUE.

4.9.6. SUBROUTINE VALUE.

This subroutine is called from DCODR after input values have been parsed for errors. VALUE encodes the characters for a value from the IVAL array into an ASCII word (IBUF) so that the value can be decoded into either its real or integer value. Each word (location) of IVAL contains one ASCII character. The ENCODE function packs these characters into one word. The DECODE function converts the ASCII word (IBUF) into either FVAR or IVAR depending on the ITYPE flag. ITYPE=0 will use a floating point format (F10.0) and ITYPE=1 will use an integer format (I10). ITYPE from FINLE and L in VALUE are synonymous.

4.9.7. SUBROUTINE MXINP.

This subroutine is called from FINLE to determine if the number of values input for a varible exceeds the dimensioned size of the variable. Error diagnostics are printed if the size is exceeded. Any excessive inputs are ignored by FINLE while storing values into the /INPUT/ common block.

4.9.8. SUBROUTINE COLINE.

This subroutine is called from FINLE to print a line of column counters that relate to the column positions for the printout of the input record. This routine is called when an error is found in the input record. An error diagnostic is printed by FINLE with the column number where the error was encountered. The output of COLINE aids the user in locating the error.

4.10 LOGAM OUTPUT ROUTINES

With the exception of case total personnel requirements printed by Subroutine PERSON all LOGAM outputs are printed with the following subroutines.

1. INPOP Prints the namelist inputs in alphabetical order.

2. IDVOP Prints individual LRU case outputs and summarized LRU case totals.

3. IPVOP Prints initial provisioning by supply echelon for an individual LRU case.

4. TTLOP Prints in a LOGAM format the accumulated costs for individual LRU cases as case and grand totals.

5. EIGHT Prints case and grand totals for LOGAM System Maintenance and Support Costs using the DA PAM 11-4 format. Also prints the case total when TOE costs are included with the LOGAM costs.

6. SUMRY Does not print results but it does accumulate the outputs from LRU cases that are being summarized. The summarized cases are output to a disc file where they will be printed by Subroutine IDVOP.

7. PAGE Prints a header title for each page of output.

8. SENSIT Prints the sensitivity inputs at the top of an individual LRU output page.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (INPOP)

4.10.1 SUBROUTINE INPOP

This subroutine is used to print the NAMELIST /L/ input values with their variable names listed in alphabtical order.

The data for each individual LRU data case is written to a working file (ND) in LOGAM. In addition to the namelist inputs, (SAVV), the LRU unit description, (UNITIS), and the REMARK statement are written to an unformatted record (binary) of ND. The values for a set of variables are read from ND and printed for each LRU case. A set of output variables are composed of the number of outputs that can be effectively printed on one line. Once all LRU cases for a set of variables are printed, the file (ND) is rewound and the next set of values are read from ND and printed. SAVV is a 326 word array that is equivalenced to the variable list of COMMON/INPUT/. Reading an LRU case into SAVV is the same as reading the data into the common block.

All of the code for this subroutine will not be included in this description. The computer memory assignments and 2 sets of variable outputs will be listed to show the logic of INPOP. For all other output sets the logic is repetitive with only the variable names changing. The order the names are output will be in the order they are listed in COMMON/INPUT/.

A. This section contains the assignment of program variables to computer memory.

1. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This statment assigns all of the program input, output and working files. The units for these variables are assigned in the BLOCK DATA routine. K004FX is used in this routine as the print file and ND is the working file that contains the LRU case data.

2. COMMON/LRU/LRU,NRU,ILRU,NLRU,ILRU2,NLRU2
   This statement contains the counters for LRU cases or case groups. "LRU" is the counter for the number of cases input and is the only variable of this list that is used in INPOP. File ND has "LRU" number of cases written on it when INPOP is entered.

3. COMMON/INPUT/ARA,ARAD,AYZP,CAD,CALMAN,CALPUB,CALSET,
   CCAL,CCALP,CCALR,CCSP,CCSPP,CCSPR,CDDI,CDEO,
   CDFD,CDID,CDIO,CDIST,CDMAN,CDOE,CDOI,CDPMAN,
   CDPRMN,CDRMAN,CEMAN,CEN,CEND,CERMAN,
   CFTD,CGMAN,CGRMAN,CI,CII,CKIT,CKMD,CKME,CKMI,
   CKMO,CKPD,CKPI,CKPO,CKUD,CKUE,CKUI,CKUO,CLRUPG,
   CMODPG,CMP,CONMAN,CONTCT,CPE,CPI,CPII,CPP,
   CPUBII,CPUBV,CPV,CRI,CRII,CRM,CRP,CRU,CRV,
   CSDEP,CSDSU,CSESU,CSGSU,CTCPUB,CTRA,CTRAD,
   CTRCAL,CTRI,CTRII,CTRSPT,CTRV,CUBEM,CUBEP,CUBEU,

CUCE,CUP,CV,DAOQL,DD,DDS,DI,DIS,DTE,DTI,DTO,
E,EACAL,EACSP,ED,EDS,EE,EREI,ETE,ETEI,ETI,
ETII,EVDM,EVDR,EVDT,EVEM,EVER,EVET,EVIM,
EVIR,EVIT,EVOM,EVOR,EVOT,FE,FI,FII,FINT,
FLM FMD,FMI,FMO,FN,FNGF,FNSP,FSA,FTI,
FTII,FTM,FTP,FTU,FUD,FUE,FUI,FUO,G,H,
HPM,HPP,HPU,IBG,IFLAG,IMF,INHIB,IO,
IOPER,IS,JTED,NA,NB,NU,OD,ODS,OL,OST,
OTF,P,PMR,PP,PPR,PUR,QMM,QMP,QMU,QTD,
QTE,QTI,QTMD,QTME,QTMI,QTMO,QTO,QTPD,
QTPI,QTPO,RDD,REO,REPEAT,RF,RID,ROI,SL,
SMD,SME,SMF,SMI,SMO,SPE,SPEV,SPEVR,STAT,
SUD,SUE,SUI,SUO,SVE,SVR,SVT,SVV,TALMAN,
TAT,TATE,TAYZ,TC,TD,TDI,TDMAN,TDMW,TDPMI,
TDPMII TDPRI,TDPRII,TDR,TDRMAN,TE,TEMAN,
TENMAN,TEO,TER,TERMAN,TF,TFR,TGMAN,TGRMAN,TI,
TID,TIMW,TIO,TIR,TMD,TMDD,TMDR,TMI,TMID,TMIR,
TMO,TMOD,TMOR,TOE,TOI,TOMW,TONMAN,TRC,TUMD,
TUMI,TUMO,WD,WDM,WDR,WE,WEM,WER,WI,WIM,WIR,
WM,WMR,WMT,WO,WOM,WOR,WP,WTKIT,WU,YAT,YD,
YMWO,YP,YR,YZ,ZFL,ZI,ZM,ZO,ZP,ZU,SENSY,T

This common block contains the list of input variables found in NAMELIST/L/. A description for these variables can be found in Section 5. The variables that comprise the first 326 memory locations are in alphabetical order and these are the values printed by INPOP. The last 2 variables (SENSY and T) are not part of this output.

4. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
TAT(4),TAYZ(10),ZM(4),ZP(3),
ZU(4),SENSY(266),T(200)

These are the dimensioned variables of COMMON/INPUT/. Variables G through ZU along with the undimensioned variables of /INPUT/ total to 326 memory locations. These are the 326 values printed by INPOP.

5. DIMENSION SAVV(326)
EQUALIVANCE (ARA,SAVV)

These statements equivalence the SAVV array to the first 326 values of COMMON/INPUT/. ARA is the first variable name listed in /INPUT/. Reading the inputs into one variable name (SAVV) simplifies the reading of the LRU working file.

6. DIMENSION UNITIS (5),REMARK(9)

When file ND is created in LOGAM, the units and remark and descriptions are written to the file along with COMMON/INPUT/ values. This statement dimensions these 2 arrays to the number of memory locations written to ND.

B. The following section is an abbreviated list of the code in INPOP. The remaining code is repetitive with only the variable names in the output lists differing.

1. REWIND ND
   Rewinds the LRU data working file and readies it to read data for the first input case.

2.
```
      WRITE (K004FX,6000)
6000 FORMAT(1H1,26X,3HARA,5X,4HARAD,6X,4HAYZP,7X,3HCAD,
           7X,6HCALMAN,4X,6HCALPUB,4X,6HCALSET,8X,
           4HCCAL,4X,5HCCALP,5X,CCALR,5X,4HCCSP)
```
   These statement will skip to the top of a page (1H1) and output a set of variable names to a line of print.

3.
```
      DO 7000 I = 1,LRU
      READ(ND) SAVV,UNITIS,REMARK
7000 WRITE (K004FX,6001)I,UNITIS,ARA,ARAD,AYZP,CAD,
          CALMAN,CALPUB,CALSET,CCAL,CCALP,CCALR,CCSP
```
   These statements read a record of data from ND and prints the values for the variable list of the WRITE statement. The same variables are input and printed until all LRU input cases are exhausted. "I" is the LRU case number and UNITIS is a 20 character LRU case description. When SAVV is read from ND the values for variable ARA through CCSP of the COMMON/INPUT/ list are automatically stored.

4.
```
6001 FORMAT(1X,I4,5A4,F5.2,4X,F5.2,4X,F6.2,6X,
            F5.0,5X,F6.0,4X,F5.0,7X,F5.2,3X,
            F7.0,3X,F8.0,2X,F7.0,3X,F7.0)
```
   This statement is formatted such that the values for the WRITE statement above will be printed beneath the variable names printed in statement 2. The outputs from the above statements will form a column of values under their respective variable names with each row of print being the input values for an LRU case.

5. REWIND ND
   The LRU input working file is rewound to ready it for the output of the next set of variable names. The logic is repeated from here on, in that the same LRU cases are read. Only the list of variables to be printed are different and the page skip control (1H1) is not a part of the format statements. The last variable to be output in this subroutine is ZU.

4.10.2 SUBROUTINE IDVOP(IL)

This subroutine prints output for individual LRU cases and summarized LRU cases. The output level to be printed is determined by the IL flag passed through the suroutine calling argument. With IL=1, individual LRU outputs are printed. There are two pages of output for each individual LRU case. The first page is the initial provisions for each supply echelon. This page is printed in Subroutine IPVOP. IPVOP is called from this subroutine (IDVOP). After calling IPVOP for provisioning outputs, the next page of output that includes initial stock totals, replenishment spares, residuals, work and test equipment demands, and costs is printed.

With IL=2, outputs summarized for distinct LRU cases are printed using essentially the same format as described for page two of the individual cases. The only relevant differences are that initial provisions are printed on this page and not on a seperate page and the availabilities are not printed.

IDVOP is called from the main program when the print flag for individual case outputs INHIB=0. If summarized cases are selected (IFLAG=0) in LOGAM then this subroutine is called to print the summarized output regardless of the value of INHIB. LRU cases are summarized and stored on file ITAP in Subroutine SUMRY. Each summarized case is read from ITAP into the "C" array and printed using the individual LRU case formats.

A. This section of IDVOP contains the assignment of variables to computer memory.

1. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains the assignment of tape and disc files used by LOGAM. K004FX is the output print file and ITAP is the file containing summarized LRU data. These two files are the only files used in this subroutine.

2. COMMON/INPUT/
   This common block contains the list of variables input by NAMELIST/L/. Refer to section A of this main program (LOGAM) for a list of these variables.

3. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),TAT(4),
   TAYZ(10),ZM(4),ZP(3),ZU(4),SENSY(266),
   T(2000)
   This statement contains the dimensioned variables of common block INPUT.

4. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5),
   REMARK(18),COSTIS(9),IPAGE
   This common block contains the information that is

printed as a page header for each page of LOGAM output.

5. COMMON/REPSP/
This common block contains the list of variables evaluated in Subroutine REORDR for replenishment spares, consumed quantities and residuals. Refer to the main program (LOGAM) for a list of these variables.

6. COMMON/COSTS/
This common block contains the list of individual LRU cost variables. These costs are computed in Subroutine LGCST. Refer to LGCST or the main program (LOGAM) for a list of these variables.

7. COMMON/MMHWF/
This common block contains the list of manpower and test equipment variables. These variables are evaluated in Subroutine TEMWF. Refer either to the description of TEMWF or the main program (LOGAM) for a list of these variables.

8. COMMON/ZERO/
This common block contains the list of variables that accumulate case total costs. These totals are accumulated in Subroutine LGCST. Refer to the main program (LOGAM) for the list of these variables.

9. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
OR,AYZIS,AYZOS
This common block contains a miscellaneous list of variables that are used in various subroutines of this program. The variables that are referenced in this subroutine are EDEE,STI,STII,AYZIS,AYZOS. EDEE,AYZIS and AYZOS are outputs for the individual LRU cases. STI and STII are Type I and Type II test equipment posting flags that control the print of work demands with the summarized outputs.

10. COMMON/LRU/LRU,NRU,ILRU,NLRU,ILRU2,NLRU2
This common block contains individual LRU case counters. ILRU and NLRU are used to enumerate the individual LRU cases that have their manpower and test equipment demands shared when using Type I test equipment. ILRU2 and NLRU2 are used for the same purpose except they are for Type II test equipment.

11. COMMON/PROVQ/
This common block contains the list of variables for initial provisioning. These variables are evaluatd in LOGAM,MPROV,IOL,REPLC and AVAIL. Refer to the main program (LOGAM) for a list of these variables.

12. DIMENSION UNS(5),C(110),SAV(326)
This statement dimensions variables that are used internal to this subroutine. UNS is the individual LRU case description written to the summarized LRU data file (ITAP). The variable C is used to read the list of summarized LRU data from the data file. SAV is equivalenced to the common INPUT block and is used in the call to Subroutine SENSIT to print the value of the variable under sensitivity analysis.

13. EQUIVALENCE(ARA,SAV)
This statement equivalences the SAV array with the common block INPUT variables. ARA is the first variable in the common block INPUT list.

B. This section of the program prints the results for individual LRU cases. Each LRU output page includes the stock, work demands, and costs for the current LRU evaluated as well as cumulative values from the previous LRU evaluations. The output from this section can be inhibited by setting INHIB=1 in the NAMELIST /L/ data deck.

1. IF(IL.EQ.2) GO TO 10
This statement transfers logic to statement 10 where summarized LRU case totals are printed when this subroutine is called from LOGAM with IL=2.

2. CALL IPVOP
This statement calls Subroutine IPVOP where initial stock provisions are printed for an individual LRU case.

3. CALL PAGE
This statement calls Subroutine PAGE to print the header information for the individual LRU output.

4. WRITE(K004FX,3000)UNITIS,REMARK
This statement prints the two line descriptions that precede the namelist inputs for the individual LRU case being output.

5. IF (SENSY(1).NE.0.) CALL SENSIT (SAV)
This statement calls Subroutine SENSIT when a sensitivity case is being evaluated to print additional header information describing the nature of the sensitivity run.

6. WRITE (K004FX,3001)COSTIS,AYZOS,AYZIS
This statement prints the the cost unit description, operational availability and inherent availability,

respectively. The format for this WRITE statement as well as the other WRITE statements in this section is not included.

7. WRITE (K004FX,3002)CET,CTST,CFT,CMPT,CIVT,CROT,CWHT,
CSAT,CSHT,GCT
This statement prints the cost values for prime equipment, test equipment, test equipment space, manpower, supply, ordering, storage, administration, shipping, and total, respectively.

8. WRITE(K004FX,3003)EDEE,QT,RQN,QB,QC,QQC,RU,
QTM,RQMN,QBM,QCM,QQCM,RM,
QTP,RQPN,QBP,QCP,QQCP,RP
This statement prints the quantity of installed LRUs (EDEE), initial provisions, reorders, consumed material and residuals for LRUs, modules, and parts, respectively.

9. WRITE(K004FX,3005)AEY,AERY,AOY,AORY,AIY,AIRY,ADY,ADRY,
ILRU,NLRU,SAEY,SAERY,SAOY,SAORY,SAIY,SAIRY,
SADY,SADRY,CAEY,CAERY,CAOY,CAORY,CAIY,CAIRY,
CADY,CADRY
This statement prints the test equipment and repair channel maintenance manhours for one maintenance location at the Organization, Direct Support, General Support, and Depot levels. The parameters AEY through ADRY are the test and repair data for a single LRU input case. Parameters SAEY through SADRY are the cumulative test and repair data for LRU input cases that are to be shared. Data will be shared between cases until the flag ETI=1 is input. ILRU and NLRU are the case numbers for the first and last LRU cases that are shared. Parameters CAEY through CADRY are the cumulative test and repair data for all LRU input cases in a concept.

10. IF(ETI.EQ.1.0)WRITE(K004FX,3006)ILRU,NLRU,ESUY,ESURY,
ESU,ESUM, ESUR,DSUY,DSURY,DSU,DSUM,DSUR,
GSUY,GSURY,GSU,GSUM,GSUR,DEPY,DEPRY,DEP,
DEPM,DEPR
This statement prints the test and repair demands for Type I test equipment at one maintenance location for Organizational, Direct Support, General Support and Depot Maintenance levels. The data printed here are the accumulated values for LRU cases shared until the flag ETI=1 is input. The order of output is test hours (ESUY), repair hours (ESURY), test equipment (ESU), test men (ESUM), and repair men (ESUR). ILRU and NLRU denote the LRU case numbers for the first and last case in the set being shared.

11. IF(ETII.EQ.1.0)WRITE(K004FX,3007)ILRU2,NLRU2,
DEPAIY,DEPARY,DEPAIE,DEPAIM,DEPAR
This statement prints the accumulated demands for Type II test equipment at the Depot. The demands printed here are for test and repair manhours (DEPAIY,DEPARY), test equipment (DEPAIE), and test and repair manpower (DEPAIM, DEPAR). This output is printed only if the posting of cummulative demands is requested by the input flag ETII. ILRU2 and NLRU2 denote the individual LRU cases that were accumulated for this output.

12. WRITE (K004FX,3021)CQTE,CQTO,CQTI,CQTD,CQTT,CRUT,
CQTME,CQTMO,CQTMI,CQTMD,CQTMT,CRMT,CQTPO,
CQTPI,CQTPD,CQTPT,CRPT
This statement prints the cost of initial provisions and the residual values for LRUs, modules and parts. CQTE,CQTO,CQTI and CQTD are the LRU costs at each supply echelon where CQTT is the total LRU cost and CRUT is the end of program residual value. Similarily values for modules (CQTME,CQTMO,CQTMI,CQTMD,CRTMT,CRMT) and parts (CQTPO,CQTPI,CQTPD,CQTPT,CRPT) are printed. Since parts are not stocked at the equipment level there is not a variable printed for this level, hence CQTPO for Direct Support is the first part cost printed.

13. WRITE(K004FX,3022)PVGCT,PCGT,CMPT,PDELTA,
EPVGCT,SEPV,ECMPT,DELTA
This statement prints the present value and expected value costs as well as the deltas between expected values and dedicated values. PVGCT is the present value cost total for the LRU case being evaluated. PVGCT would include escalation if that option had been selected. DELTA is the difference between the present value total and the expected value total, where expected value is the value computed without dedicating work demands. PDELTA is the DELTA factor with escalation cost added. Expectd present value cost total (EPVGCT) for the individual LRU case is PVGCT less PDELTA. PCGT and SEPV are the summations of PVGCT and EPVGCT, respectively. These summations are made for individual LRU cases in a case total concept. CMPT is the present value cost of maintenance manpower for an LRU case and ECMPT is the expected value, where ECMPT is CMPT minus DELTA.

14. GO TO 20
This statement transfers logic to statement 20 where a return is made back to the main program once individual LRU outputs are printed.

C. This section of IDVOP prints the summarized LRU case outputs.

Values for distinct LRU cases are accumulated (summarized) in Subroutine SUMRY when the value of IFLAG=0 is input in NAMELIST/L/. NDLRU designates the number of distinct individual LRU cases in a summarization process. For example, if NDLRU=11 there would be eleven LRU cases for which summaries would be made. For the summarization process to work properly there would have to be two or more sets of eleven individual LRU cases. Using the example of NDLRU=11 and two sets (11) of LRU cases, the cases summarized would be $LRU_1 + LRU_{12}$, $LRU_2 + LRU_{13}$, etc. NDLRU is read from the eighth data card of the input data stream.

This section of code is entered when IDVOP is called from the main program (LOGAM) with IL=2. IDVOP will be called when IFLAG=0 and grand total (NU=-3) outputs are requested in LOGAM.

1. 10 CONTINUE
   A transfer is made to this statement when IDVOP is called with IL=2.

2. REWIND ITAP
   This statement readies the summarized LRU data file to begin reading information to print.

3. DO 8014 KKK=1,NDLRU
   This statement is the beginning of the DO loop for reading the summarized data from file ITAP. NDLRU is the number of related LRU cases for which a summarization was performed.

4. CALL PAGE
   This statement calls Subroutine PAGE to print page header information.

5. READ (ITAP)(UNS(I),I=1,5)
   READ (ITAP)(C(I),I=1,110)
   These statements read the unit descriptions and the summarized data array, respectively, from the current data file (ITAP).

6. WRITE(K004FX,8013)UNS,TLRU,
   This statement prints the LRU case description and the description for the type of LRU summarization.

7. WRITE (K004FX,8015) COSTIS
   This statement prints the cost unit description (COSTIS).

8. WRITE(K004FX,3002) (C(I),I=3,12)
   This statement prints the summarized cost values for development of prime equipment (CET), development of test equipment (CTST), housing of test equipment (CFT),

test and repair maintenance manpower (CMPT), supply (CIVT), ordering (CROT), storage (CWHT), supply administration (CSAT), shipping (CSHT), and grand total, (CGT), respectively.

9. WRITE(K004FX,3003) C(110),(C(I),I=13,14),QB,(C(I), I=15,19),QBM,(C(I),I=20,24),QBP,(C(I),I=25,27)
This statement prints the summarized values for installed LRUs, initial provisions, reorders and residuals. QB, QBM, and QBP are the reorder lot sizes. The values printed will be for the last individual LRU case of the group summarized. The other outputs in the order printed are EDEE, QT, RQN, QC, QQC, RU, QTM, RQMN, QCM, RM, QTP, RQPN, QCP, QQCP and RP.

10. WRITE (K004FX,3005) (C(I),I=28,35),KKK,KKK,(C(I),I = 36,51)
This statement prints the summarized values for test equipment and repair channel maintenance manhours for one maintenance location at the Organizational, Direct Support, General Support, and Depot facilities. The variables output are the summarized values for AEY, AERY, AOY, AORY, AIY, AIRY, ADY, ADRY, SAEY, SAERY, SAOY, SAORY, SAIY, SAIRY, SADY, and SADRY, CAEY, CAERY, CAOY, CAORY, CAIY, CAIRY, CADY, AND CADRY. KKK is the case number for the summarized outputs read from ITAP.

11. IF(STI.EQ.1.0)WRITE (K004FX,3006)KKK,KKK, (C(I),I=52,71)
This statement prints summarized values for the accumulated work demands of Type I test equipment at Organization, Direct Support, General Support and Depot. These outputs are printed only if the work demand flag for Type I test equipment was turned on (ETI=1.) in an individual LRU data case. The outputs here are summarized values for ESUY, ESURY, ESU, ESUM, ESUR, DSUY, DSURY, DSU, DSUM, DSUR, GSUY, GSURY, GSU, GSUM, GSUR, DEPY, DEPRY, DEP, DEPM, DEPR.

12. IF (STII.EQ.1.0) WRITE (K004FX,3007) KKK,KKK,(C(I),I=72,76)
This statement prints summarized values for the accumulated work demands of Type II test equipment at the Depot facilities. These outputs are printed only if the work demand flag for Type II test equipment was turned on (ETII=1.) in an individual LRU data case. The outputs here are the accumulated values for DEPAIY,DEPARY,DEPAIE, DEPAIM, and DEPAR, respectively.

13. WRITE (K004FX,3020)(C(I),I=82,92)
This statement prints the summarized values for the initial provisions of LRUs (QTE, QTO, QTI, QTD),

modules (QTME,QTMO, QTMI, QTMD), and parts (QTPO, QTPI, QTPD), respectively.

14. WRITE (K004FX,3021) (C(I),I=93,109)
This statement prints the summarized cost of initial provisions for LRUs (CQTE, CQTO, CQTI, CQTD, CQTT, CRUT), modules (CQTME,CQTMO, CQTMI, CQTMD, CQTMT, CRMT), and parts (CQTPO, CQTIPI, CQTPD, CQTPT, CRPT), respectively. CQTT, CQTMT, and CQTPT are the total provisioning costs and CRUT, CRMT, and CRPT are the end of life salvage value for LRUs, modules, and parts, respectively.

15. WRITE (K004FX,3022)C(1),C(2),C(6),C(81),(C(I),I=77,80)
This statement prints summarized values for present value and expected value costs. The outputs here for summarized values of PVGCT,PCGT,CMPT, PDELTA,EPVGCT, SEPV,ECMPT, AND DELTA, respectively. Refer to statement B-13 above for a description of these variables.

16. 8014 CONTINUE
This statement is the end of the DO Loop for printing summarized results. After NDLRU loops have been made through the preceeding statements, logic will begin with the next statement.

17. 20 CONTINUE
RETURN
END
These statements terminate execution of IDVOP and returns logic to the main program (LOGAM).

4.10.3 SUBROUTINE IPVOP

This subroutine prints the outputs for initial provisioning. IPVOP is called from Subroutine IDVOP each time an individual LRU output page is requested (INHIB=0).

A. This section of IPVOP contains the assignment of variable names to computer memory.

1. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5),
   REMARK(18),COSTIS(9),IPAGE
   This common block contains the information to print as a page header for each page of LOGAM output.

2. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains the tape or disc file assignments used by LOGAM. This subroutine uses K004FX as the output device to the printer. Normally K004FX is assigned the unit 6.

3. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,FMWO,
   OR,AYZIS,AYZOS
   This common block contains a miscellaneous list of variables that are used throughout the program. Only one of the variables, OR, is used in this subroutine. OR is the fractional part of AYZP that is used as the operational availability to be achieved.

4. COMMON/PROVQ/DELU,PDUE,PDUO,PDUI,PDUD,PQTU,
   PDME,PDMO,PDMI,PDMD,PQTM,
   PDPO,PDPI,PDPD,PQTP,
   PQTE,PQTO,PQTI,PQTD,PQT,
   QT,QTM,QTP
   This common block contains the provisioning quantities printed by this subroutine.

5. COMMON/INPUT/
   This common block contains the list of variables that are input through NAMELIST/L/. Refer to section A of the main program for the variable list.

B. This section of IPVOP prints initial provisions for LRUs, modules and parts at each supply echelon. There are three values printed for LRUs. For example, at the equipment level PDUE (computed or predetermined value), PQTE (required value based on a requested availability) and QTE (the distributed quantity for PQTE) are printed for comparative purposes.

1. CALL PAGE
   Print page header for the initial provisioning outputs.

2. WRITE(K004FX,4110)UNITIS,REMARK
This statement prints the two card description that precedes each individual LRU NAMELIST case.

3. WRITE(K004FX,4120)
WRITE(K004FX,4140)
These statements print titles for provisioning outputs.

4. WRITE(K004FX,4150)PDUE,PQTE,QTE,PDME,QTME,PDPE,
QTPE,EDS,H(1)
WRITE(K004FX,4160)PDUO,PQTO,QTO,PDMO,QTMO,PDPO,
QTPO,ODS,H(2)
WRITE(K004FX,4170)PDUI,PQTI,QTI,PDMI,QTMI,PDPI,
QTPI,DIS,H(3)
WRITE(K004FX,4180)PDUD,PQTD,QTD,PDMD,QTMD,PDPD,
QTPD,DDS,H(4)
These statements print the provisioning quantities at the Organizational, Direct Support, General Support and Depot supply points, respectively. PDUE, PDME, and PDPE are the computed provisions from either the maintenance rules, supply rules or predetermined values. QTE, QTME, and QTPE are the provisions after distributing the required values over the number of supply installations (EDS). PQTE is the required number of LRUs to meet a predetermined operational availability. The computed value (PDUE) is modified in Subroutine AVAIL to meet the desired availability when the fractional part of AYZP is greater than 0.5. The stock authorization at organizational level is determined by the value of H(1). When H(1)=1, stock is authorized; if H(1)=0, no stock is authorized. The description for the parameters at the other stock points are similar to the descriptions of the organizational level parameters.

5. WRITE (K004FX,4190) PQTU,PQT,QT,PQTM,QTM,PQTP,QTP
This statement prints the totals for initial provisioning of LRU's, modules and parts. PQTU is the total LRU provisioning required to meet the maintenance concept input. PQT is the LRU requirement to meet a predetermined operational availability. QT is the rounded total for either PQTU or PQT. PQTM, QTM, PQTP, and QTP are the computed and rounded values for modules and parts, respectively.

6. IF(OR.GT.0.5)WRITE(K004FX,4195)OR
This statement prints the value for a requested operational availability. When the desired availability is not greater than 50 percent, the logic to compute stock to meet availability is bypased.

7. WRITE(K004FX,4420)G

This statement prints the maintenance policy values.

8. RETURN
END
This statment terminates execution of IPVOP and transfers logic to Subroutine IDVOP.

4.10.4 SUBROUTINE TTLOP

Subroutine TTLOP is called from the main program to print costs for either case or grand totals. When NU is less than zero, TTLOP is called with IPRNT=0 to print a case total. If NU=-2 or -3 a second call is made to TTLOP with IPRNT=1 for grand total output. All of the output variables are in common block ZERO. The case totals for cost are accumulated in Subroutine LGCST. The grand total accumulations are made in the main program. When grand totals are to be printed, the case total parameters of common ZERO are restored in the main program with the grand total accumulations. This permits the printing of case and grand total outputs by using the same write statements.

A. This section of TTLOP contains the assignment of program variables to computer memory.

1. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block assigns all of the tape or disc files used by LOGAM. The output file K004FX is the only unit used in this subroutine.

2. COMMON/ZERO/CCET,CCTS,CCTSR,CCF,CCM,CCMF,CCMD,CCMFD,
   CTRF,CTRDEP,CTR,CIV,CIVREC,CRT,CWH,CSA,
   CSAREC,CSH,GCT,CTREC,PCD,CQTU,PCP,CQTM,PCR,
   CQTP,PCS,CQTUMP,PCGT,SEMPT,SEPC,SPCR,SEPV,
   SDEL,SPDEL,CAYZ(10),CAYZI(10)
   This common block contains the list of cost parameters that are accumulated in Subroutine LGCST as case totals. The only members of this list that are not printed here are the availability accumulators CAYZ and CAYZI. This common block will be reset with the values from the grand total accumulator after the case total is printed and a grand total output is requested.

3. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5),
   REMARK(18),COSTIS(9),IPAGE
   This common block contains information to print as a page header. COSTIS is the cost unit description and is the only variable of this list that is used in TTLOP.

B. This section of TTLOP prints the totals page. Case totals are printed when IPRNT=0 and grand totals when IPRNT=1.

1. CALL PAGE
   This call to page will print header information at the top of the output page.

2. IF(IPRNT.EQ.0) WRITE(K004FX,5041)

```
      IF(IPRNT.EQ.1) WRITE(K004FX,5042)
```

These statements print a title for either a case total (IPRNT=0) or a grand total (IPRNT=1).

3.
```
      WRITE(K004FX,2001)COSTIS,CCET,CCTS,CCTSR,
     &         CCF,CCF,CCM,CCMF,CCMD,CCMFD,CTRF,CTRDEP,
     &         CTR,CIV,CIVREC,CRT,CRT,CWH,CWH,CSA,CSAREC,
     &         CSH,CSH,CGT,CTREC,PCD,CQTU,PCP,CQTM,PCR,
     &         CQTP,PCS,CQTUMP,PCGT
      WRITE(K004FX,2)SEMPT,SEPC,SPCR,SEPV,SDEL,SPDEL
```

These statements print the costs for either case total or grand total accumulations. The parameters in these statements are the same parameters listed in common block ZERO.

4.
```
      RETURN
      END
```

These statements return logic to the main program.

4.10.5 SUBROUTINE EIGHT

This subroutine prints System Maintenance Support and System Operating and Support Costs in a format consistent with DA PAM 11-4. System Maintenance Support costs are evaluated in Subroutine LGCST using LOGAM maintenance factors. These costs are accumulated in LGCST into the SDA array of common block DAPAM. These accumulations are for each individual LRU case in a case total. Printout of these accumulations are made with a call to EIGHT from the main program (LOGAM) with the value of J=0. Intermittent printouts of these accumulations can be made before a case total is completed. By inputting NU=-1 in NAMELIST/L/ the case accumulations up to this point are printed but the case accumulator (SDA) array is not reinitialized. With NU=-2 the case accumulator array will be reset for the next case total accumulation. Resetting of the accumulator arrays is performed in LOGAM (main routine). When case totals are printed the value of IPRNT=0 is set in LOGAM, which causes the case total page header statement to be printed.

Grand total printout of System Maintenance Support costs can be selected when there is more than one case total concept. Grand totals are printed when NU=-2 or -3 is input. IPRNT=1 is set in LOGAM to print the grand total page header statement. The accumulation of grand totals is made into the SDAM array of the main program. Only System Maintenance Support costs can be output as a grand total; grand totals for System Operating and Support costs are not accumulated.

System Operating and Support costs include the System Maintenance Support costs and the costs from a TOE. TOE costs will be evaluated by Subroutine OPER after printing case totals for System Maintenance if IOPER=1 is set in NAMELIST/L/. The TOE costs are added to the LOGAM maintenance costs and printed as a case total for System Operations and Support. To print Operation and Support costs, Subroutine EIGHT is called from the main program with J=1 and IPRNT=0.

A. This section of EIGHT includes the assignement of program names to computer memory and a description of the subroutine arguments.

1. SUBROUTINE EIGHT(J,IPRNT)
   This subroutine is called from LOGAM. The use of the agurments of EIGHT and their values are:

   J=0 Print System Maintenance Support costs.
   J=1 Print System Operation and Support costs.
   IPRNT=0 Print CASE TOTAL statement.
   IPRNT=1 Print GRAND TOTAL statement.

2. COMMON/POUT/POUT(13),AMULT

This common block stores the output for the System Operation and Support costs. These costs are accumulated in the TOE Operation and Support routines (see OPER).

3. COMMON/DAPAM/SDA(10),WD(6)
This common block stores the System Maintenance and Support costs. These costs are accumulated in Subroutine LGCST. WD is accumulated in LGCST as SDA(11) through SDA(16).

4. DIMENSION PER(25)
This statement dimensions an array for storing individual support costs as a percentage of the total support cost.

B. This section of EIGHT determines the level of support costs to print and prints the page titles for the level selected. For the maintenance support level, certain LOGAM outputs are stored into the Operation and Support printout array (POUT).

1. CALL PAGE
This statement will cause the line printer to skip to the top of the next page and print a page header.

2. IF(IPRNT.EQ.0) WRITE(6,100)
IF(IPRNT.EQ.1) WRITE(6,101)
These statements print either a case total (IPRNT=0) title or a grand total (IPRNT=1) title.

3. IF(J.GT.0) GO TO 250
When J=0 the reference to EIGHT is made from LOGAM to print System Maintenance costs. With J>0, a transfer is made to statement 250 to print Operation and Support costs.

4. POUT(6)=SDA(10)
POUT(7)=SDA(8)
POUT(11)=SDA(9)
POUT(12)=0.
POUT(13)=SDA(8)+SDA(9)+SDA(10)
These statements store LOGAM maintenance support costs into the Operation and Support output array. The SDA values are the accumulated costs from Subroutine LGCST.

5. WRITE (6,161)
This statement prints the title for the LOGAM System Maintenance Support Cost outputs.

6. GO TO 251
This statement transfers to the logic to print costs in

the DA PAM 11-4 format.

7. WRITE (6,162)
This statement will print the title for System Operation and Support Costs.

C. The next section of code computes cost totals and percentages of totals for R&D, Investment, O&S, and grand total.

1.
```
250 CONTINUE
    TAI=0
    DO 40 I=1,7
40  TAI=TAI+SDA(I)
    X=POUT(13)-POUT(12)+TAI
```
These statements compute the total O&S costs. The SDA values are computed in LGCST. POUT(13) is either the maintenance support costs described in section B-3 above or the total support cost from the post processor Subroutine SEVEN. POUT(12) is zero for maintenance support outputs but is set to LGCST's in Subroutine SEVEN for operation and support outputs. CGT is the total maintenance support cost. Since the LOGAM maintenance cost total was added to the O&S costs in Subroutine SEVEN, CGT is subtracted when printing operation and support costs.

2. IF (X.EQ.0.) X=1.0
This statement sets the O&S phase cost to 1. in order that a division by zero will not occur when percentages of total cost are computed.

3. PER(1)=100.
This statement sets the percentage of R&D cost to 100 percent since there is only one cost element (Development Engineering) output for this cost phase.

4.
```
   XB=0.
   DO 85 I=2,6
85 XB=XB+WD(I)
```
These statements sum the individual Investment Costs into a total Investment. WD are the individual costs that are accumulated in LGCST as SDA(12) thru SDA(16).

5. IF (XB.EQ.0)XB=1.
This statement sets the total Investment Cost to 1. to prevent dividing by zero while computing individual cost percentages.

6. GT=X+XB+WD(1)
This statement sums the O&S costs (X), Investment Costs (XB), and Development Cost (WD(1)) into a grand total.

7.
```
      DO 86 I=2,5
   86 PER(I) = (WD(I)/XB)*100.0
      PER(25) = (WD(6)/XB)*100.0
```
These statements compute the percentage of individual Investment Costs to the total Investment Cost.

8.
```
PER(6)=(P(1)/X)*100.0
PER(7)=(SDA(1)/X)*100.0
PER(8)=(POUT(2)/X)*100.0
PER(9)=(POUT(3)/X)*100.0
```
These statements compute the percentage of individual O&S military personnel costs to the total O&S cost.

9.
```
PER(10)=(SDA(2)/X)*100.0
PER(11)=(POUT(4)/X)*100.0
PER(12)=(POUT(5)/X)*100.0
```
These statements compute the percentage of individual O&S Consumption Costs to the total O&S cost.

10.
```
      DO 87 I=13,17
   87 PER(I)=(SDA(I-10)/X)*100.0
      DO 88 I=18,23
   88 PER(I)=(POUT(I-12)/X)*100.0
```
These statements compute the percentage of the individual direct and indirect support operations cost to the total O&S cost.

D. The next section of EIGHT prints the maintenance and Operating Support Costs. Each line of output will include the WBS code, cost description, cost, and percentage of total cost. There are three levels of output data. Level 1 is the Research and Development cost with the WBS Code of "1.000." Investment costs are the second level having a WBS code of "2.000." The level three costs are for Operating and Support with a WBS code of "3,000." The variable PER in the write statements will be the cost percentage. The cost variables of the write statements will be described after each statement.

1.
```
      WRITE (6,74)
   74 FORMAT(82X,4HCOST4X,10HPERCENTAGE)
```
These statements print the column header for the following outputs.

2.
```
      WRITE (6,82)
   82 FORMAT(5X,5H1.000,5X,24HRESEARCH AND
             DEVELOPMENT)
```
These statements print the R&D cost title at WBS level 1.000.

3.
```
      WRITE (6,83)WD(1),PER(1)
```

```
      83 FORMAT (5X,5H1.010,14X,23HDEVELOPMENT
                 ENGINEERING19X,F20.2,4XF7.2)
```

These statements print the Development Engineering cost which was computed in LGCST as SDA(11).

```
4.       WRITE (6,84)WD(1),PER(1)
      84 FORMAT (5X,5HTOTAL56X,F20.2,4XF7.2/)
```

These statements print the total R&D cost.

```
5.       WRITE (6,89)
      89 FORMAT (5X,5H2.000,5X,15HINVESTMENT COST)
```

These statements print the Investment Cost Phase title at WBS level 2.000.

```
6.       WRITE (6,90)WD(2),PER(2)
      90 FORMAT (5X,5H2.010,9X,24HNON-RECURRING
                 INVESTMENT23X,F20.2,4XF7.2)
```

These statements print the non-recurring investment which was computed in LGCST as SDA(12).

```
7.       WRITE(6,91)WD(3),PER(3)
      91 FORMAT(5X,5H2.050,9X,4HDATA43X,F20.2,4XF7.2)
```

These statements print the cost of data during the Investment Phase. WD(3) is computed in LGCST as SDA(13).

```
8.       WRITE(6,92)WD(4),PER(4)
      92 FORMAT (5X,5H2.080,9X,31HTRAINING SERVICES
                AND EQUIPMENT16XF20.2,4XF7.2)
```

These statements print Investment Training Services and Equipment costs. WD(4) is computed in LGCST as SDA(14).

```
9.       WRITE(6,93)WD(5),PER(5)
      93 FORMAT(5X,5H2.090,9X,31HINITIAL SPARES AND
                REPAIR PARTS16XF20.2,4XF7.2)
```

These statements print investment cost of Initial Spares and Repair Parts. WD(5) is computed in LGCST as SDA(15).

```
10.      WRITE(6,200) WD(6),PER(25)
     200 FORMAT(5X,5H2.11 ,9X,5HOTHER,42X,F20.2,4X,F7.2)
```

These statements print Other Investment Costs. WD(6) is computed in LGCST as SDA(16), the procurement of test equipment.

```
11.      WRITE(6,94)XB,PER(1)
      94  FORMAT(5X,5HTOTAL56X,F20.2,4XF7.2/)
```

These statements print the Total Investment Cost. XB is summed in this subroutine.

```
12.      WRITE(6,10)
```

```
      10  FORMAT(5X,5H3.000,5X,26HOPERATING AND SUPPORT COST)
```

These statements print the title for the Operating and Support Costs at WBS level 3.000.

```
13.       WRITE(6,11)
      11  FORMAT(5X,5H3.010,5X,22HMILITARY PERSONNEL
```

These statements print the Military Personnel title for costs at level 3.010 of the O&S phase.

```
14.       WRITE(6,12)POUT(1),PER(6)
      12  FORMAT(5X,5H3.011,5X,32HCREW PAY AND
                 ALLOWANCES19X,F20.2,4XF7.2)
```

These statements print the military personnel cost for Crew Pay and Allowances. POUT(1) was computed Subroutine ONE.

```
15.       WRITE(6,13)SDA(1),PER(7)
      13  FORMAT(5X,5H3.012,5X,39HMAINTENANCE PAY AND
                 ALLOWANCES12X1,F20.2,4XF7.2)
```

These statements print the military personnel cost for Maintenance Pay and Allowances.

```
16.       WRITE(6,14)POUT(2),PER(8)
      14  FORMAT(5X,5H3.013,14X,27HINDIRECT PAY AND
                 ALLOWANCES15X,F20,2,4XF7.2)
```

These statements print Indirect Pay and Allowances for military personnel. POUT(2) is computed in Subroutine ONE.

```
17.       WRITE(6,15)POUT(3),PER(9)
      15  FORMAT(5X,5H3.014,14X,27HPERMANENT CHANGE OF
                 STATION15X,F20.2,4XF7.2)
```

These statements print the Permanent Change of Station for military personnel. POUT(3) is computed in Subroutine TWO.

```
18.       WRITE(6,16)
      16  FORMAT(5X,5H3.020,9X.11HCONSUMPTION)
```

These statements print a title for consumables during the O&S phase. The consumption costs are at WBS level 3.020.

```
19.       WRITE(6,17)SDA(2),PER(10)
      17  FORMAT(5X,5H3.021,14X,20HREPLENISHMENT SPARES
                 22X,F20.2,4XF7.2)
```

These statements print the consumable costs for Replenishment Spares. SDA(2) was computed in Subroutine LGCST.

```
20.       WRITE(6,18)POUT(4),PER(11)
      18  FORMAT(5X,5H3.022,14X,29HPETROLEUM,OIL
                 AND LUBRICANTS13X,F20.2,4XF7.2)
```

These statements print the consumable costs for POL. POUT(4) was computed in Subroutine THREE.

```
21.       WRITE(6,19)POUT(5),PER(12)
19    FORMAT(5X,5H3.023,14X,36HUNIT TRAINING
                AMMUNITION AND MISSILE6X,F20.2,4XF7.2)
```

These statements print the consumable costs for Unit Training Ammuniton and Missiles. POUT(5) was computed in Subroutine FOUR.

```
22.       WRITE(6,21)
21    FORMAT(5X,5H3.030,9X,17HDEPOT MAINTENANCE)
```

These statements print a title for Depot Maintenance costs during O&S.

```
23.       WRITE(6,22)SDA(3),PER(13)
22    FORMAT(5X,5H3.031,14X,5HLABOR37X,F20.2,4XF7.2)
```

These statements print the cost of Labor at Depot Maintenance.

```
24.       WRITE(6,23)SDA(4),PER(14)
23    FORMAT(5X,5H3.032,14X,8HMATERIEL34X,
                F20.2,4XF7.2)
```

These statements print the Depot Maintenance Materiel costs.

```
25.       WRITE(6,24)SDA(5),PER(15)
24    FORMAT(5X,5H3.033,14X,14HTRANSPORTATION
                28X,F20.2,4XF7.2)
```

These statements print the Depot Maintenance Transportation costs.

```
26.       WRITE(6,25)SDA(6),PER(16)
25    FORMAT(5X,5H3.040,9X,22HMODIFICATIONS
                MATERIAL25X,F20.2,4XF7.2)
```

These statements print the Modifications Material costs at WBS level 3.040.

```
27.       WRITE(6,26)
26    FORMAT(5X,5H3.050,9X,31HOTHER DIRECT
                SUPPORT OPERATIONS)
```

These statements print a title for Other Direct Support Operations cost at WBS level 3.050.

```
28.       WRITE(6,34)SDA(7),PER(17)
34    FORMAT(5X,5H3.051,14X,27HMAINTENANCE, CIVILIAN
                LABOR15X,F20.2,4XF7.2)
```

These statements print the cost of Maintenance, Civilian Labor at Other Direct Support Operations.

```
29.       WRITE(6,35)POUT(6),PER(18)
35    FORMAT(5X,5H3.052,14X,12HOTHER DIRECT
```

```
              30X,F20.2,4XF7.2)
```

These statements print other Direct Support Operations cost. POUT(6) is computed in Subroutine SEVEN when System Operations and support costs are to be printed. When system maintenance support costs are to be printed POUT(6) is set to SDA(10).

```
30.       WRITE(6,27)
27    FORMAT(5X,5H3.060,9X,27HINDIRECT
              SUPPORT OPERATIONS)
```

These statements print a title for O&S Indirect Support Operations.

```
31.       WRITE(6,28)POUT(7),PER(19)
28    FORMAT(5X,5H3.061,14X,21HPERSONNEL
              REPLACEMENT21X,F20.2,4XF7.2)
```

These statements print the Personnel Replacement cost. POUT(7) is computed in Subroutine SEVEN when System Operations and Support costs are printed. For System Maintenance Support costs POUT(7) is set to SDA(8) in this subroutine.

```
32.       WRITE(6,29)POUT(8)PER(20)
29    FORMAT(5X,5H3.062,14X,34HTRANSIENTS, PATIENTS
              AND PRISONERS8X,F20.2,4XF7.2).
```

These statements print the cost of Transients, Patients and Prisoners. POUT (8) is computed in Subroutine TWO.

```
33.       WRITE(6,30)POUR(9),PER(21)
30    FORMAT(5X,5H3.063,14X,35HQUARTERS, MAINTENANCE
              AND UTILITIES7X,F20.2,4XF7.2)
```

These statements print the cost of Quarters, Maintenance, and Utilities. POUT (9) is computed in Subroutine TWO.

```
34.       WRITE(6,31)POUT(10),PER(22)
31    FORMAT(5X,5H3.064,14X,15HMEDICAL
              SUPPORT27X,F20.2,4XF7.2)
```

These statements print the cost of Medical Support. POUT(10) is computed in Subroutine TWO.

```
35.       WRITE(6,32)POUT(11),PER(23)
32    FORMAT(5X,5H3.065,14X,14HOTHER
              INDIRECT28X,F20.2,4XF7.2)
```

These statements print Other Indirect Support Operations cost. POUT(11) is computed in Subroutine SEVEN for System Operations and Support Costs. For System Maintenance Support costs POUT(11) is set to SDA(9) in this subroutine.

```
36.       WRITE(6,95)X,PER(1)
95    FORMAT(5X,5HTOTAL56X,F20.2,4XF7.2/)
```

These statements print the total cost of the O&S lifecycle phase. The total O&S cost X is summed in this subroutine.

37. WRITE(6,96)GT
96 FORMAT(5X,11HGRAND TOTAL50X,F20.2)
These statements print the Grand Total of all costs in the systems lifecycle. GT is summed in this subroutine.

38. RETURN
END
These statements terminate execution of subroutine EIGHT and returns program control to LOGAM.

4.10.6 SUBROUTINE SUMRY(IL)

This subroutine summarizes the outputs from individual LRU cases for distinct groups (NDLRU) of LRUs. There are two scratch files used in the summarization process. Individual case outputs are stored into the "VV" array and added to the outputs (array "C") read from one of the scratch files from a previous LRU group. Files K001FX and K002FX are the scratch files used. The first group of LRU outputs are written to K001FX. The second group would read K001FX, add the data to the "VV" values and write the results to file K002FX. If a third group of LRU cases were input K002FX would be read for its summarized value and added to "VV" and the results written to K001FX. This alternating between K001FX and K002FX will continue for however many distinct groups (NDLRU) of LRUs input.

This subroutine is called from two different locations in the main program (LOGAM) when the summarization flag (IFLAG=0) is turned on in NAMELIST/L/. The subroutine is called first for each individual LRU case executed. When these calls are made the value of the calling argument IL=1. After a grand total output request (NU=-3) is made in the main program and the summarized outputs are printed this routine will be called with IL=2 to clear the accumulator array (C) and to reset the file counters.

The unit number for the last scratch file written is assigned to ITAP. Subroutine IDVOP will read file ITAP and print the results. Subroutine IDVOP is called from LOGAM when the value for NU=-3.

A. This section of SUMRY contains the assignment of program variables in computer memory.

1. COMMON/INPUT/
   This common block contains the list of NAMELIST/L/ input variables. Refer to section A of the main program for the listing of these variables.

2. DIMENSION G(20),H(4),OL(4),OST(4),SL(4),
   TAT(4),TAYZ(10),ZM(4),ZP(3),
   ZU(4),SENSY(266),T(2000)
   This statement contains all of the dimensioned variables in common block INPUT.

3. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5),
   REMARK(18),COSTIS(9),IPAGE
   This common block contains page header information. UNITIS is the only variable in this list that is used in SUMRY. UNITIS is the individual LRU case description that is written on a record of the scratch files that precedes the record for the LRU outputs.

SECTION 4 - DETAILED PROGRAM DESCRIPTION (SUMRY)

4. COMMON/COSTS/
This common block contains the list of individual LRU costs that are evaluated in Subroutine LGCST. Refer to LOGAM or LGCST for a listing of these parameters.

5. COMMON/MMHWF/
This common block contains the list of manpower and test equipment demands that are evaluated in Subroutine TEMWF for an individual LRU case. Refer to LOGAM or Subroutine TEMWF for listing of these paramenters.

6. COMMON/ZERO/
This common block contains the list of LRU cost acumulations (case totals) that are evaluated in Subroutine LGCST. Refer to LOGAM or Subroutine LGCST for a listing of these parameters.

7. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
This common block contains a list of the data files used by this program. K001FX,K002FX and ITAP are the file names used in this subroutine. The units for these files are set in the block data (BLKDAT) routine.

8. COMMON/SUMLRU/TLRU(15),NDLRU,ICN,ISET,IATE,IA7,IA8
This common block contains the summarized LRU description (TLRU), the number of distinct LRUs in a group (NDLRU), the current LRU case counter (ICN) and the scratch file control flags. TLRU and NDLRU are input on the eighth input card and the remaining parameters are initialized in the block data routine.

9. COMMON/REPSP/
This common block contains a list of the replenishment spares, consumed stock and residual stock that are evaluated in Subroutine REORDR. Refer to the main program for a listing of these parameters.

10. COMMON/MISC/AAIE,EDEE,QUA,YR8,SAVE,STI,STII,
FMWO,OR,AYZIS,AYZOS
This common block contains a list of miscellaneous parameters that are used in the various subroutines. EDEE (total LRUs installed) is the only parameter from this list that is used in this subroutine.

11. COMMON/PROVQ/
This common block contains the list of initial provisioning parameters that are evaluated by maintenance provisioning (MPROV), supply provisioning (BASIC,IOL) or predetermined (INPUT). Refer to the main program (LOGAM) for a listing of these parameters.

12. DIMENSION UNS(5),C(110),VV(110)

This statement dimensions the arrays used in the summarization of LRU cases. UNS is the individual LRU case discription (20 characters), VV is the values for the individual case outputs and C is the accumulation of outputs for the distinct LRU cases.

B. This section of SUMRY accumulates the outputs from an individual LRU case with outputs from an LRU case of a previous group. The LRU outputs are stored into the VV array. The VV array outputs are then accumulated with the previous accumulations read from a scratch file. The accumulations are made in the C array and written to one of the scratch files; either K001FX or K002FX. Writing and reading of accumulated data alternates betwen K001FX and K002FX for each group of LRU cases evaluated. The unit number for the latest scratch file written is stored in ITAP, where ITAP will be the unit read when outputting summarized data in Subroutine IDVOP.

1. IF(IL.EQ.2) GO TO 100
   This statement will transfer logic to statement 100 to reinitialize the accumulator array (C) after all distinct LRU groups have been evaluated. IL=1 is set in LOGAM for each LRU case evaluated which forces the logic through the following code to perform the summarization.

2. ICN=ICN+1
   This statement counts the number of individual LRU cases in a distinct group of LRU accumulations. When ICN becomes equal to NDLRU, ICN will be reset to zero to begin accumulations with a new group of LRUs.

3. VV(1) = PVGCT
   VV(2) = PCGT
   VV(3) = CET
   VV(4) = CTST
   VV(5) = CFT
   VV(6) = CMPT
   VV(7) = CIVT
   VV(8) = CROT
   VV(9) = CWHT
   VV(10) = CSAT
   VV(11) = CSHT
   VV(12) = GCT
   VV(13) = QT
   VV(14) = RQN
   VV(15) = QC
   VV(16) = QQC
   VV(17) = RU
   VV(18) = QTM
   VV(19) = RQMN

VV(20) = QCM
VV(21) = QQCM
VV(22) = RM
VV(23) = QTP
VV(24) = RQPN
VV(25) = QCP
VV(26) = QQCP
VV(27) = RP
VV(28) = AEY
VV(29) = AERY
VV(30) = AOY
VV(31) = AORY
VV(32) = AIY
VV(33) = AIRY
VV(34) = ADY
VV(35) = ADRY
VV(36) = SAEY
VV(37) = SAERY
VV(38) = SAOY
VV(39) = SAORY
VV(40) = SAIY
VV(41) = SAIRY
VV(42) = SADY
VV(43) = SADRY
VV(44) = CAEY
VV(45) = CAERY
VV(46) = CAOY
VV(47) = CAORY
VV(48) = CAIY
VV(49) = CAIRY
VV(50) = CADY
VV(51) = CADRY
VV(52) = ESUY
VV(53) = ESURY
VV(54) = ESU
VV(55) = ESUM
VV(56) = ESUR
VV(57) = DSUY
VV(58) = DSURY
VV(59) = DSU
VV(60) = DSUM
VV(61) = DSUR
VV(62) = GSUY
VV(63) = GSURY
VV(64) = GSU
VV(65) = GSUM
VV(66) = GSUR
VV(67) = DEPY
VV(68) = DPREY
VV(69) = DEP
VV(70) = DEPM
VV(71) = DEPR

VV(72) = DEPAIY
VV(73) = DEPARY
VV(74) = DEPAIE
VV(75) = DEPAIM
VV(76) = DEPAR
VV(77) = EPVGCT
VV(78) = SEPV
VV(79) = ECMPT
VV(80) = DELTA
VV(81) = PDELTA
VV(82) = QTE
VV(83) = QTO
VV(84) = QTI
VV(85) = QTD
VV(86) = QTME
VV(87) = QTMO
VV(88) = QTMI
VV(89) = QTMD
VV(90) = QTPO
VV(91) = QTPI
VV(92) = QTPD
VV(93) = CQTE
VV(94) = CQTO
VV(95) = CQTI
VV(96) = CQTD
VV(97) = CQTT
VV(98) = CRUT
VV(99) = CQTME
VV(100) = CQTMO
VV(101) = CQTMI
VV(102) = CQTMD
VV(103) = CQTMT
VV(104) = CRMT
VV(105) = CQTPO
VV(106) = CQTPI
VV(107) = CQTPD
VV(108) = CQTPT
VV(109) = CRPT
VV(110) = EDEE

These statements store the outputs from an individual LRU data case that will be accumulated with outputs from previous data cases. When the outputs are from the first group of LRUs, VV is written directly to a scratch file.

4. IF (ISET.EQ.1) GO TO 8503

This statement transfers logic to statement 8503 to read the summarized LRU data array "C" from disk unit K001FX (UNIT 17) and to write the present LRU data array (VV) to disk unit K002FX (UNIT 18). Initially, ISET=0 and will remain zero until all the LRUs in a summarization set are processed, i.e., ICN=NDLRU.

5. IF (IATE.EQ.1) GO TO 8603
This statement transfers logic to statement 8603 to read the summarized LRU data array "C" from disk unit K002FX (UNIT 18) and to write the present LRU data to disk unit K001FX (UNIT 17). Initially, IATE=0 and will remain zero until all LRUs in a summarization set are processed, i.e., IATE=NDLRU.

6. WRITE (K001FX) (UNITIS(I),I=1,5)
WRITE (K001FX) (VV(I),I=1,110)
These statements write the LRU data case description and the data for the individual case to disk unit K001FX, respectively. This logic is entered only on the first NDLRU passes of the LRUs in a data concept. For the following summarization logic to work properly, the user must have the LRUs grouped in sets of NDLRU LRUs.

7. IF (ICN.EQ.NDLRU) GO TO 8501
When the first NDLRU set of LRUs are completed a transfer is made to statement 8501 to set a flag (ISET) that will prevent logic flow through the previous three statements.

8. GO TO 8502
The evaluation of the first NDLRU set of LRUs in a data concept has not been completed, therefore, logic is transferred to statement 8502 which is outside the summarization logic.

9. 8501 ISET = 1
ICN=0
ITAP = K001FX
REWIND K001FX
REWIND K002FX
GO TO 8502
These statements are executed after the first set (NDLRU) of LRUs are completed. The data files are rewound for later use when LRU data will be read from the files, summarized, and written back to the files. ITAP is the file unit containing the data summarized from the previous set of LRUs. Transfer of logic is to statement 8502 to begin the evaluation of the next set of LRUs.

10. 8503 CONTINUE
READ (K001FX) (UNS(I), I = 1,5)
READ (K001FX) (C(I), I=1,110)
DO 8508 I=1,110
8508 C(I)=C(I)+VV(I)
WRITE (K002FX) (UNS(I), I = 1,5)
WRITE (K002FX) (C(I),I=1,110)

```
     IA8 = IA8+1
```

These statements are executed on even numbered sets of LRUs. LRU descriptive information (UNS) and data summarized from previous LRU sets are read from K001FX. The summarized data array is added to the values for the present LRU case (VV) and written to disk unit K002FX. IA8 counts the number of passes completed through the even numbered set of LRUs.

11. `IF (IA8-NDLRU) 8502, 8601, 8502`

When all NDLRU LRUs in the even numbered data set have been processed, logic transfer is made to statement 8601 to set flags for processing the next (odd) set of LRUs; otherwise, go to statement 8502 and complete the present set.

12.
```
8601 IA8 = 0
     IATE = 1
     ISET = 0
     ICN = 0
     ITAP = K002FX
     REWIND K001FX
     REWIND K002FX
     GO TO 8502
```

These statements are executed after all the LRUs (NDLRU) in an odd numbered LRU data set are processed. The odd number set counter (IA7) and the odd number set flag (IATE) are turned off and the LRU case counter (ICN) is reset. The even numbered set flag (ISET) is turned on, ITAP set to the current summarised data file, and files rewound for next data set.

16. `8502 CONTINUE`

This statement is the terminal point after all LRU cases are summarized and written to disc file.

17.
```
IF(ETI.EQ.1.0) STI=1.0
IF(ETII.EQ.1.0) STII=1.0
```

These statements set the flags to print maintenance demands for Type I and Type II test equipments when summarized cases are outpt. If demands are posted for any individual LRU case the respective flag will be set. These flags control the print in Subroutine IDVOP.

18. `GO TO 110`

This statement transfers logic to a return to the main program after the accumulations for an individual LRU case is completed.

C. This section of SUMRY reinitializes the accumulator array and

summarization flags. This code is entered once the outputs of a summarized case have been printed. When NU=-3 is input and the output routine IDVOP is called from the main program to print the summarizations, SUMRY is called with IL=2. With IL=2 a transfer is made to the next statement for the resetting of values.

1. 100 CONTINUE
   A transfer is made to this statement from the statement in section B-1. When IL=2 a jump is made here.

2. ```
   DO 8018 I=1,110
   8018 C(I)=0.0
   ```
   These statements reset the values in the summarized data array.

3. ```
   ISET=0
   REWIND K001FX
   IATE=-0
   IA7=0
   IA8=8
   REWIND K002FX
   ICN=0
   STI=0
   STII=0
   ```
   These statements reset the variables and rewind the files used in summarizing LRU data cases. After this logic the program is ready to summarize results for a sensitivity analyses or another LRU data concept.

4. ```
   110 CONTINUE
   RETURN
   END
   ```
   These statements return logic to the main program.
   ```

4.10.7 SUBROUTINE PAGE

This subroutine prints a header for a page of output. The header describes the logistics analysis performed, the date, and a page number. The information printed here is a once only input in LOGAM. This subroutine is referenced from all of the output routines.

1. COMMON/HEADER/DATE(5),TEXT(72),ANLYIS(5),UNITIS(5),
   REMARK(18),COSTIS(9),IPAGE
   This common block contains the output page header. Information for this list is read from the 8 input header cards and printed with each page of output.

2. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains the list of tape and disc files used by LOGAM. Only K004FX, the output file, is used in this subroutine.

3. IPAGE=IPAGE-1
   This statement decrements the page counter. The LOGAM program uses a negative page counter to achieve a particular output effect. As an example if the page counter is "-29-", the "-29" portion is the negative page counter.

4. WRITE (K004FX,1)IPAGE,TEXT,ANLYIS,DATE
   This statement will print header information at the top of an output page.

5. RETURN
   END
   These statements terminate Subroutine PAGE and returns control to the routine that called PAGE.

4.10.8 SUBROUTINE SENSIT

This subroutine is referenced by the main program, LOGAM, when the output pages that are being printed are for a sensitivity analysis case. SENSIT prints a page header that describes the parameters modified for a sensitivity analysis.

A. This section of SENSIT assigns computer memory locations to variables used internal to this routine and to variables that are transmitted by way of labled COMMON from the main program, LOGAM.

1. SUBROUTINE SENSIT (VAR)
   VAR is the list of NAMELIST inputs stored in common block INPUT.

2. COMMON/SENS/NRULE(12),NVAR(12),MODE,KPASS,NPASS
   This statement contains the list of variables used to perform a sensitivity analysis. NRULE are the input data modification rules and NVAR are the positions of the variables to be modified relative to the COMMON/INPUT/ list. MODE is the number of values that were modified for the present pass (KPASS) through the LRU cases. There is a maximum of 12 values that can be modified for any given pass. NPASS is the total number of passes to be made through the LRU cases. MODE can differ for each pass.

3. COMMON/LRU/LRU,NRU,ILRU,NLRU,ILRU2,NLRU2
   This common block contains the list of LRU case counters. NRU is the individual LRU case number currently under a sensitivity analysis. NRU is the only variable from this list that is used in this subroutine.

4. COMMON/FILES/K001FX,K002FX,K003FX,K004FX,ITAP,ND,K8
   This common block contains the list of tape and disc files used in the LOGAM program. These variables are initialized in the block data (BLKDAT) routine. The print file, K004FX, is the only variable of this list that is used in SENSIT.

5. DIMENSION VALUE(12)
   This statement sets the maximum size of the array that stores the modified inputs for a sensitivity analysis. The greatest number (MODE) of variables that can be changed per sensitivity case is 12.

6. DIMENSION VAR(326)
   This statement dimensions the input array to SENSIT as the number of words occupying common block INPUT of the main program. The NAMELIST input values are passed to

SENSIT with VAR as its calling argument.

B. The next section of code stores the value of the parameters modified for a sensitivity analysis and prints a page header for the next page of outputs.

1.
```
      I=0
10    I=I+1
      M=NVAR(I)
      VALUE(I)=VAR(M)
      IF(I.LT.MODE)GO TO 10
```
These statements loop through the number of input variables (MODE) that were modified, finds the variables location (M) in the input list (VAR), and stores the value (VALUE) for printing.

2.
```
WRITE(K004FX,1)KPASS,NRU,(NVAR(I),VALUE(I),I=1,MODE)
```
This statement prints the information for the current LRU data case for which a sensitivity analysis is being performed. The sensitivity logic is discussed in part H of LOGAM.

3.
```
RETURN
END
```
These statements return logic back to LOGAM.

SECTION 5

PROGRAM LOGAM INPUT DEFINITIONS

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

5. LOGAM Inputs

The inputs to LOGAM consists of page header controls, individual LRU case data, sensitivity data and TOE data. The header inputs use formatted inputs. All other inputs use NAMELIST/L/. A description of the input variables will be given in the following sections.

5.1 Program Controls - This section describes the LOGAM header control cards. These cards input information for use as output page headers, to define the cost output units, to control accumulation of data for summarized LRU case outputs, and to control the preprocessing of NAMELIST inputs.

1. TEXT (cards 1 through 4) - Each card may contain 72 columns of information to enable the analyst to print up to 4 lines of identifying information on each page of printed output. There must be four cards of input even if some are blank.

2. ANLYIS (card 5) - This card provides information in the first 20 columns to identify some specific information of the analysis to be printed on an output page.

3. DATE (card 6) - This card provides the first 20 columns to input a date for each page of output.

4. COSTIS (card 7) - This card contains 36 positions to input a description for the units that the cost will be output; i.e., "THOUSANDS OF DOLLARS".

5. AMULT (card 7) - Card 7 also contains the cost unit in column locations 42-51.

The five inputs above are read from the same FORTRAN statement. This statement is described in section B of LOGAM.

6. TLRU (card 8) - This input is used to print with page header, information that will describe common case concepts. The first 60 columns of card 8 are reserved for this input.

7. IRUN (card 8) - This input controls the NAMELIST/L/ input processor. IRUN=0 preprocesses all individual LRU data cases for errors but does not execute the data cases. IRUN=1 processes each data case for errors and executes the case (even if errors are found). Column 70 is reserved for this input.

8. NDLRU (card 8) - This input instructs LOGAM as to how many LRU input cases are to be summarized into a concept. For example if NDLRU=5, every fifth LRU case will be summed; i.e., LRU1+LRU6, LRU2+LRU7, etc. This input is right justified in columns 71-80.

The READ statement for the three inputs above are described in section B of LOGAM. All of the eight parameters just described are input only once per program execution. The following two statements will be input each time an LRU data case is input.

9. UNITIS (card 9) - This input describes the class and class number for the next LRU to be processed. The first 20 columns of information is used to print with each page of LRU data outputs. Section B of LOGAM describes the READ statement for this input.

10. REMARK (card 10) - This input provides a 72 column data field that describes the next LRU data case to be input. The READ statement for this input is described in section B of LOGAM.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

5.2 NAMELIST INPUTS - With the exception of the inputs just described in section 5.1, all other inputs are read through NAMELIST/L/. These inputs include the individual LRU case inputs, sensitivity analysis array (SENSY) and the Table of Organization and Equipment (TOE) data array (T). Each NAMELIST/L/ data case must be preceded by the UNITIS and REMARK cards described above.

This set of inputs will follow immediately after the REMARK card. The first NAMELIST card starts with $L in any card column except column 1. Column 1 must be blank for all NAMELIST inputs. The last input in the data set must be terminated by a $. In the designation of the variables for sensitivity testing, the LOGAM program is structured to reference the variables by their numbered positional location in COMMON block INPUT rather than by name. The numbers associated with the following input variable list are the numbered positions for the variables in COMMON/INPUT/. The list is in alphabetical order with the exception of SENSY and T. These two variables are at the end of the list. Refer to section 3 (LOGAM Features) for a description of the sensitivity testing procedure and the TOE option.

The default values for all the NAMELIST inputs are set in the block data routine (BLKDAT). The default values are listed with each variable description.

LOGAM has been modified to read the NAMELIST inputs using a preprocessor (FINLE) instead of the FORTRAN NAMELIST read statement. The list of NAMELIST variables are stored in a DATA statement in Subroutine FINLE and not in the usual NAMELIST statement. The preprocessor checks each individual LRU data case for errors and proceeds to the next case even if errors occur. The standard FORTRAN NAMELIST processor terminates execution once an error is found in the data case. With IRUN=1 set in the eighth data card (section 5.1) the individual LRU NAMELIST cases are processed for errors and the data case executed. If an input error is found, the value used will either be the default value or the value input from a previous case. With IRUN=0, the data cases are processed for errors but not executed.

The description for all NAMELIST/L/ inputs and their default values begin on the following page (section 5.2.1). Several of the inputs described in section 5.2.1 include a "footnote" label. These inputs are described in more detail in their respective footnotes. Since some of the footnotes take up to half of a page of print, the inclusion of the footnotes at the bottom of the description page could distract from the ease of using the descriptions. Therefore, the footnotes were included in a section of their own. For any input description in section 5.2.1 that has a footnote label, a further discussion of that input can be found in section 5.2.2.

5.2.1 NAMELIST/L/ VARIABLE DESCRIPTION

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
1.	ARA	Annual military manpower turnover fraction for field test and repair.	0.
2.	ARAD	Annual civilian manpower turnover fraction for depot test and repair.	0.
3.	AYZP	Control to specify the method for computing the initial provision quantities. It generally is input as a signed whole number as follows: AYZP = 1. Use LOGAM Maintenance Rule. AYZP = 0. Use LOGAM Supply Rule. AYZP = -1. Provision quantities are to be input (predetermined rule). AYZP may also contain a fractional part. The absolute value of the fraction is used to control override of stock to meet a specified availability. The absolute value of the fraction states the operational availability to be achieved. Example AYZP = 1.0 Use LOGAM Maintenance Rule. No force on availability. AYZP = 1.9 Use LOGAM Maintenance Rule. Force to get 90% operational availability. AYZP = 0.73 Use LOGAM Supply Rule. Force to get 73% operational availability.	1.
4.	CAD	Cost in dollars per year to retain an item (LRU, module, non-standard part) in the supply system.	0.
5.	CALMAN	Cost in dollars per year for a calibration man.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
6.	CALPUB	Cost in dollars for technical data for calibration/Type III test equipment. (CALPUB is set to zero within the program after use.)	0.
7.	CALSET	Number of calibration/Type III test sets and teams.	0.
8.	CCAL	Cost in dollars to develop calibration/Type III test equipment. (CCAL is set to zero within the program after use.)	0.
9.	CCALP	Cost in dollars to procure a calibration/Type III test set.	0.
10.	CCALR	Cost in dollars per year to support a calibration/Type III test set.	0.
11.	CCSP	Cost in dollars to develop contact support/Type IV test sets. (CCSP is set to zero within the program after use.)	0.
12.	CCSPP	Cost in dollars to procure a contact support/Type IV test set.	0.
13.	CCSPR	Cost in dollars per year to support a contact support/Type IV test set.	0.
14.	CDDI	Shipping from Depot to General Support (units as CDEO).	0.
15.	CDEO	Shipping from the installation to the Direct Support Activity. Input as dollars per item per pound per trip. Used in the computation of shipping and handling charges.	0.
16.	CDFD	Shipping for a one-way trip from a contractor to the government depot (units as CDEO). Applied to shipment of reprocured material.	0.
17.	CDID	Shipping from General Support to Depot (units as CDEO).	0.
18.	CDIO	Shipping from General to Direct Support (units as CDEO).	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
19.	CDIST	Cost in dollars per item per pound to distribute initial provision of LRUs, modules, and parts.	0.
20.	CDMAN	Cost in dollars per year for a test man at Direct Support.	0.
21.	CDOE	Shipping from Direct Support to the installation (units as CDEO).	0.
22.	CDOI	Shipping from Direct to General Support (units as CDEO).	0.
23.	CDPMAN	Cost in dollars per year for a test man at Depot.	0.
24.	CDPRMN	Cost in dollars per year for a repairman at Depot.	0.
25.	CDRMAN	Cost in dollars per year of a repairman at Direct Support.	0.
26.	CEMAN	Cost in dollars per year for a test man at the Equipment level.	0.
27.	CEN	Cost in dollars to enter a line item into the supply system.	0.
28.	CEND	Cost in dollars to develop an LRU. (CEND is set to zero within the program after use.)	0.
29.	CERMAN	Cost in dollars per year for a repairman at the Equipment level.	0.
30.	CFTD	Cost in dollars per square foot/month for floor space at Depot for test equipment.	0.
31.	CGMAN	Cost in dollars per year for a test man at General Support.	0.
32.	CGRMAN	Cost in dollars per year for a repairman at General Support.	0.
33.	CI	Cost in dollars to develop Type I test equipment. (CI is set to zero within the program after use.)	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
34.	CII	Cost in dollars to develop Type II test equipment. (CII is set to zero within the program after use.)	0.
35.	CKIT	Cost in dollars for a modification kit.	0.
36.	CKMD1	Safety stock coefficient for module stock at Depot.	0.
37.	CKME	Safety stock coefficient for module stock at equipment level.	0.
38.	CKMI	Safety stock coefficient for module stock at General Support.	0.
39.	CKMO	Safety stock coefficient for module stock at Direct Support.	0.
40.	CKPD	Safety stock coefficient for part stock at Depot.	0.
41.	CKPI	Safety stock coefficient for part stock at General Support.	0.
42.	CKPO	Safety stock coefficient for part stock at Direct Support	0.
43.	CKUD	Safety stock coefficient for LRU stock at Depot.	0.
44.	CKUE	Safety stock coefficient for LRU stock at Equipment level.	0.
45.	CKUI	Safety stock coefficient for LRU stock at General Support.	0.
46.	CKUO	Safety stock coefficient for LRU stock at Direct Support.	0.
47.	CLRUPG	Cost in dollars to program and provide technical data for Type I test equipment for LRU repair.	0.
48.	CMODPG	Cost in dollars to program and provide technical data for Type I test equipment for module repair for each module type.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
49.	CMP	Cost in dollars for spare or replacement module.	0.
50.	CONMAN	Cost in dollars per year/per man for the contact support team.	0.
51.	CONTCT	Number of contact support sets and teams.	0.
52.	CPE	Nonrecurring production cost in dollars for an LRU. (CPE is set to zero within the program after use.)	0.
53.	CPI	Cost in dollars to procure a Type I test set.	0.
54.	CPII	Cost in dollars to procure a Type II test equipment.	0.
55.	CPP	Average cost in dollars for a spare or replacement part.	0.
56.	CPUBII	Cost in dollars to program and provide technical data for Type II test equipment. (CPUBII is set to zero within the program after use.)	0.
57.	CPUBV	Cost in dollars for technical data for Type V test equipment. (CPUBV is set to zero after use).	0.
58.	CPV	Procurement cost in dollars for Type V test equipment.	0.
59.	CRI	Cost in dollars per year for materials to support a Type I test station.	0.
60.	CRII	Cost in dollars per year for material to support a Type II test station.	0.
61.	CRM	Cost in dollars per module reorder action.	0.
62.	CRP	Cost in dollars per part reorder action.	0.
63.	CRU	Cost in dollars per LRU reorder action.	0.
64.	CRV	Yearly cost in dollars to set up training programs for Type V test set.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
65.	CSDEP	Cost in dollars per cubic foot per month for material storage at Depot.	0.
66.	CSDSU	Cost in dollars per cubic foot per month for material storage at Direct Support.	0.
67.	CSESU	Cost in dollars per cubic foot per month for material storage at Equipment level.	0.
68.	CSGSU	Cost in dollars per cubic foot per month for material storage at General Support.	0.
69.	CTCPUB	Cost in dollars to program and provide technical data for contract support/Type IV test equipment. (CTCPUB is set to zero within the program after use.)	0.
70.	CTRA	Cost in dollars to train one man for field maintenance.	0.
71.	CTRAD	Cost in dollars to train one man for Depot maintenance.	0.
72.	CTRCAL	Nonrecurring cost in dollars to set up training program for calibration Type III test equipment teams.	0.
73.	CTRI	Nonrecurring cost in dollars to set up training program for Type I test equipment.	0.
74.	CTRII	Nonrecurring cost in dollars to set up training program for Type II test equipment.	0.
75.	CTRSPT	Nonrecurring cost in dollars to set up training program for contact support in Type IV test equipment.	0.
76.	CTRV	Non recurring cost in dollars to set up training programs for Type V test equipment.	0.
77.	CUBEM	Storage volume in cubic feet for a module.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
78.	CUPEP	Storage volume in cubic feet for a part.	0.
79.	CUBEU	Storage volume in cubic feet for an LRU.	0.
80.	CUCE[2]	Cost in dollars per year for equipment level manpower to provide preventive scheduled maintenance. Used in combination with SMF to model expected value manpower at the equipment level.	0.
81.	CUP	Cost in dollars for the LRU under analysis (deployment, replacement, and provision LRUs).	0.
82.	CV	Development cost in dollars for Type V test equipment.	0.
83.	DAOQL	Fraction of Depot workload that is good when delivered to the field stockage point. 1-DAOQL is recycled.	1.
84.	DD	Number of Depot level maintenance locations.	1.
85.	DDS[14]	Number of Depot level support points.	1.
86.	DI	Number of General Support maintenance locations.	1.
87.	DIS[14]	Number of General Support supply points.	0. ~~1.~~
88.	DTE	Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from the equipment level.	0.
89.	DTI	Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from General Support.	0.
90.	DTO	Pipeline in days for delays in handling repairable LRUs or modules being shipped rearward from Direct Support.	0.
91.	E	Failure rate per operating hour.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
92.	EACAL	Controls posting out one time costs for calibration/Type III test channels including manpower. Only the values zero and unity are permitted. EACAL = 0 no posting of costs. EACAL = 1 forces the posting of costs. EACAL is reset to zero after each use.	0.
93.	EACSP	Controls posting out one time costs for contact support/Type IV test equipment and manpower. Only the values zero and unity are permitted. EACSP = 0 no posting of costs. EACSP = 1 forces the posting of costs. EACSP is reset to zero after each use.	0.
94.	ED^3	Number of deployment installations.	1.
95.	EDS^{14}	Number of equipment level supply points.	1.
96.	EE	The number of material systems (LRUs) at each deployment installation.	1.
97.	EREI	Expected value flag for test and repair men on major items at the equipment level.	1.
98.	ETE	Controls posting out of accumulated work demands for men and Type V test equipment. ETE = 0 no posting of cost ETE = 1 forces the posting of cummulative demand into the cost totals and resets the demand accumulators.	1.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
99.	ETEI	Expected value flag for Type V test equipment on major items at equipment level.	1.
100.	ETI	Controls posting out accumulated work demands at service channels of Type I test equipment and their associated repair positions. Only the values zero and unity are permitted. ETI = 0 no posting of costs. ETI = 1 forces the posting of cumulative demand into the cost totals and reset the demand accumulators.	1.
101.	ETII	Controls posting out accumulated work demands for service channels at Depot of Type II test equipment. Only the values zero and unity are permitted. ETII = 0 no posting of costs. ETII = 1 forces the posting of cumulative demand into the cost totals and resets the demand accumulators.	1.
102.	EVDM[4]	Expected value flag for test manpower at Depot.	1.
103.	EVDR[4]	Expected value flag for repair manpower at Depot.	1.
104.	EVDT[4]	Expected value flag for test equipment at Depot.	1.
105.	EVEM[4]	Expected value flag for test manpower at equipment level.	1.
106.	EVER[4]	Expected value flag for repair manpower at equipment level.	1.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
107.	EVET[4]	Expected value flag for test equipment at equipment level.	1.
108.	EVIM[4]	Expected value flag for test manpower at General Support.	1.
109.	EVIR[4]	Expected value flag for repair manpower at General Support.	1.
110.	EVIT[4]	Expected value flag for test equipment at General Support.	1.
111.	EVOM[4]	Expected value flag for test manpower at Direct Support.	1.
112.	EVOR[4]	Expected value flag for repair manpower at Direct Support.	1.
113.	EVOT[4]	Expected value flag for test equipment at Direct Support.	1.
114.	FE	The fraction of Type V test equipment manpower added for self support.	0.
115.	FI	Fraction of Type I test equipment manpower demand that is added for self-support.	0.
116.	FII	Fraction of Type II test equipment manpower demand that is added for self-support.	0.
117.	FINT[5]	Yearly interest rate. Used in the computation of present value. It is the net rate between discount rate and inflation rate. Thus, if inflation exceeds discount, FINT may be input negative. Zero input gives net cost output without discount.	0.
118.	FLM	Factor of the calibration and contact support test equipment maintenance support costs for civilian maintenance labor.	~~1.~~ 0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
119.	FMD[6]	Fraction of modules that arrive at Depot that are repaired. Modules not repaired are scrapped.	1.
120.	FMI[6]	Module repair fraction at General Support.	1.
121.	FMO[6]	Module repair fraction at Direct Support.	1.
122.	FN[7]	Number of identical LRUs within a system whose failure does not detract from system availability. Used to model effect of equipment redundancy within the system.	0.
123.	FNGF[8]	Number to specify the ratio of false "no go" LRU demands to true failures.	0.
124.	FNSP[9]	Nonstandard part fraction related to the cost for supply administration.	1.
125.	FSA	Field supply administration cost. Dollars per year per line item type per field supply location.	0.
126.	FTI	Number of square feet of space required at Depot for Type I test equipment.	0.
127.	FTII	Number of square feet of space required at Depot for Type II test equipment.	0.
128.	FTM	Analogous to FTU but is for module reprocurement.	0.
129.	FTP	Analogous to FTU and FTM but is for parts reprocurement.	0.
130.	FTU	Time factor in weeks used in the computation of LRU Stock at Depot. FTU is the fixed time cycle associated with LRU reprocurement. Typically, this is the factory start-up time between placement of an order and delivery of the first LRU.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
131.	FUD^{10}	LRU repair fraction at Depot.	1.
132.	FUE^{10}	LRU repair fraction at equipment level.	1.
133.	FUI^{10}	LRU repair fraction at General Support.	1.
134.	FUO^{10}	LRU repair fraction at Direct Support.	1.
135.	G(1)	GA-Specifies a policy of discard at failure. There are no maintenance support activities. All failures, false "no go" indications, and attrition rate inputs result in LRU discard. Only LRUs are stocked in the supply system. There is no demand for modules or parts.	0.
136.	G(2)	GB-Similar to GA but here is a provision to detect false "no go's" at Direct Support and only failed and attrited LRUs are discarded. There is no demand for module or part stock. There is a demand for checkout service at Direct Support and the algebra uses Type I test equipment input data for this.	0.
137.	G(3)	GC-Specifies LRU repair at equipment level by removing and replacing a defective module. The defective module is discarded.	0.
138.	G(4)	GD-Specifies LRU repair at Direct Support by removing and replacing a defective module. The defective module is discarded.	0.
139.	G(5)	GE- Specifies LRU repair at General Support by removing and replacing a defective module. The defective module is discarded.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
140.	G(6)	GF-Specifies LRU repair at General Support with checkout performed at Direct Support to remove false "no go" LRUs before sending the work to General Support. LRU repair is by removal and replacement of a defective module and the defective module is discarded.	0.
141.	G(7)	GG-Specifies LRU repair at Depot. Defective modules are discarded.	0.
142.	G(8)	GH-Specifies LRU repair at Depot preceded by a checkout at Direct Support to screen false "no go's". Defective modules are discarded.	0.
143.	G(9)	GI-Specifies LRU repair at equipment level and module repair at Direct Support.	0.
144.	G(10)	GJ-Specifies LRU repair at equipment level and module repair at General Support.	0.
145.	G(11)	GK-Specifies LRU repair at equipment level and module repair at the Depot.	0.
146.	G(12)	GL-Specifies LRU and module repair at Direct Support.	0.
147.	G(13)	GM-Specifies LRU repair at Direct Support and module repair at General Support.	0.
148.	G(14)	GN-Specifies LRU repair at Direct Support and module repair at Depot.	0.
149.	G(15)	GO-Specifies checkout to catch false "no go's" at Direct Support followed by LRU and module repair at General Support.	0.
150.	G(16)	GP-Specifies checkout to catch false "no go's" at Direct Support followed by LRU repair at General Support and module repair at Depot.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
151.	G(17)	GQ-Specifies LRU checkout to catch false "no go's" at Direct Support followed by LRU and module repair at Depot.	0.
152.	G(18)	GR-Specifies LRU and module repair at General Support.	0.
153.	G(19)	GS-Specifies LRU repair at General Support and module repair at Depot.	0.
154.	G(20)	GT-Specifies LRU and module repair at Depot.	0.
155.	H(1)11	Stock authorization flag at organization.	1. 0.
156.	H(2)11	Stock authorization flag at Direct Support.	1.
157.	H(3)11	Stock authorization flag at General Support.	1.
158.	H(4)11	Stock authorization flag at Depot.	1.
159.	HPM	Discretionary procurement holding time in days for modules.	0.
160.	HPP	Discretionary procurement holding time in days for parts.	0.
161.	HPU	Discretionary procurement holding time in days for LRUs. No safety stock is applied to HPU, HPM, HPP, because it is a discretionary factor and may be waived if earlier procurement is indicated by field experience.	0.
162.	IBG	A debugging FLAG, which when set to 1, causes the printout of the current values of internal variables.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
163.	IFLAG	Flag for summarizing individual LRU cases between distinct groups of LRU cases. Used generally for summing outputs of identical LRU cases that are common in two or more theaters. For this process to work properly NDLRU must be input on the 8th header card (columns 71-80) as a positive non zero value. 1 Suppresses the summarization. 0 Summarizes and prints the results.	1.
164.	IMF	Selects the file number on tape or disc that contains data sorted by MOS for maintenance support positions. The data read by selecting IMF is sorted from the AR570-2 MACRIT data tape. IMF is used in conjuction with OPER to build the TOE tables for personnel related costs.	0.
165.	INHIB	An integer to control the printout of individual LRU output. Only the numbers 0 and 1 are permitted. INHIB = 0 prints the LRU output page. INHIB =1 inhibits the printout of LRU output.	0.
166.	IO	An integer to control printout of the input NAMELIST data. IO = 0 Inhibits NAMELIST printout. IO > 0 Entire sequence of input data for all LRUs printed out in alphabetical order.	0.
167.	IOPER	Selects the option to add TOE operational costs to the LOGAM output. IOPER = 1 Initiates the subroutine to compute the Operation and Support costs derived from a typical TOE structure. The O&S costs computed conform to DA PAM 11-4.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
168.	IS	An integer to control reset functions for maintenance concept fractions, case total accumulators, availability accumulators, workload accumulators, and recall of saved input values.	0.
		IS = 0 This is the default value and should be the value used on the first pass through LOGAM. With this value the accumulator arrays are initialized without going through the recall logic to store the input data array (SAV). If the recall array is used on the first pass the default values from BLKDAT will be erased.	
		IS = 1 Anticipatory control for the next LRU. All inputs used for an LRU case where NB=0 are recalled for use with the next LRU. Any values input for the next case will modify the recalled values. Availability and workload accumulators and case total accumulators are also reset. IS is automatically reset to 1 by the program if the user requests grand total outputs with NU<-1.	
		IS ≠ 2 Resets maintenance concept (G) fractions to zero.	
		IS = 2 Retains maintenance concept (G) fraction from one LRU to the next.	
		IS = 3 Neutralizes all reset actions. It is automatically set to 3 after a pass through the initialization section of LOGAM. This assures that the accumulators will not be reset until the user inputs IS=1 or NU<-1.	

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
169.	JTED	An integer control used to designate the type and location of test equipment. JTED = 1 Permits location of Type I test equipment at the Direct Support, General Support, and Depot sites. JTED = 2 Permits location of Type I test equipment as in JTED = 1 except only Type II test equipmentis at the Depot.	1.
170.	NA^{12}	An integer to control the number of system availability modes to be tallied for the case being run.	1.
171.	NB^{13}	An integer to control initialization of default values.	0.
172.	NU	An integer to control printout of case totals and grand totals pages, reset the grand total accumulators and provide the means for a positive program stop. NU = $\geq$ 0 Suppresses print of totals page. NU = -1 Prints the case totals page. This value may be used at any time to examine the contents of the totals accumulators. The printout of the case totals page is not accompanied by any change in the accumulators or any other program variable. NU = -2 Prints the case totals page as for NU = -1 and also prints a grand totals page following the case totals page. Reset of the case total accumulators is accomplished by the control IS. IS=1 is automatically set when NU<-1 to reset the case total accumulators after printout of the case totals pages.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
		NU = -3 Provides the same function as NU = -2, i.e., it prints out both the case total and the grand total pages. Additionally, it resets the grand total accumulators.	
		NU = -4 Provides a positive program stop; used in combination with a dummy REMARK card and a dummy UNITS card followed by a NAMELIST card with NU = -4.	
173.	OD	Number of Direct Support maintenance locations.	1. 0.
174.	ODS[14]	Number of Direct Support supply or stock transfer points.	1. 0.
175.	OL(1)[15]	An array of dimension four representing the operating level of supply in days for consumables at Organization, Direct, General and Depot supply points.	0.
176.	OL(2)		0.
177.	OL(3)		0.
178.	OL(4)		0.
179.	OST(1)[15]	An array of dimension four representing the order and ship time in days for Organization, Direct, General, and Depot supply points.	0.
180.	OST(2)		0.
181.	OST(3)		0.
182.	OST(4)		0.
183.	OTF	The fraction of real time that deployed equipment operates.	1.
184.	P	Number of module types per LRU used to cost supply administration.	1.
185.	PMR	Production rate for modules. This input, PRU and PPR are overridden by the program if the rates are insufficient to meet the demand.	0.
186.	PP	Number of part types per LRU used to cost supply administration.	1.
187.	PPR	Production rate for parts. Refer to PMR description.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
188.	PUR	Production rate for LRUs. Refer to PMR description.	0.
189.	QMM	The minimum reorder quantity for modules.	1.
190.	QMP	The minimum reorder quantity for parts.	1.
191.	QMU	The minimum reorder quantity for LRUs.	1.
192.	QTD[16]	Total Depot level LRU stock quantity for all DDS locations.	0.
193.	QTE[16]	Total organization level LRU stock quantity for all EDS locations.	0.
194.	QTI[16]	Total General Support level LRU stock quantity for all DIS locations.	0.
195.	QTMD[16]	Total Depot level module stock quantity for all DDS locations.	0.
196.	QTME[16]	Total Organizational level module stock quantity for all EDS stock locations.	0.
197.	QTMI[16]	Total General Support level module stock quantity for all DIS locations.	0.
198.	QTMO[16]	Total Direct Support level module stock quantity for all ODS locations.	0.
199.	QTO[16]	Total Direct Support level LRU stock quanitity for all ODS locations.	0.
200.	QTPD[16]	Total Depot level part stock quantity for all DDS locations.	0.
201.	QTPI[16]	Total General Support level part stock quantity for all DIS locations.	0.
202.	QTPO[16]	Total Direct Support level part stock quantity for all ODS locations.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
203.	RDD	Delay time in days between request for an LRU at a maintenance Depot and handling of the request by the supply point. Used in the computation of availability in reckoning down-time at the Depot.	0.
204.	REO	REO is similar to ROI but in this instance, it sets the days of supply at the equipment level for condemned modules.	0.
205.	REPEAT	Number of LRU types in a material system (EE). These are LRUs that are similar in their failure rates and maintenance concepts but because of their type difference they impact the cost of supply administration.	1.
206.	RF	The fraction of TRC that is devoted to LRU remove and replace time excluding fault isolate and retest time.	0.9
207.	RID	When using LOGAM supply rules, RID is input in days and is a specification use to distinguish between the supply allowance for condemned modules and parts and the number of days of supply for LRUs and for repaired modules at the General Support level. Within the program, RID is summed with the input TDI to form the term RIDT which sets the days of supply at General Support for condemned modules and parts.	0.
208.	ROI	Like RID, ROI is a specification used to distinguish between the supply allowance for condemned module and parts and the number of days of supply for LRUs and for repaired modules at the Direct Support level. Within the program, ROI is summed with the input TIO to form the term ROIT. ROIT sets the days of supply at Direct Support for condemned modules and parts.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
209.	SL(1)[15]	An array of dimension four representing	0.
210.	SL(2)	the safety level days of supply for	0.
211.	SL(3)	consumables at Organization, Direct,	0.
212.	SL(4)	General and Depot supply points (definition of OL).	0.
213.	SMD[17]	Module scrap fraction at Depot.	0.
214.	SME[17]	Module scrap fraction at Organization level.	0.
215.	SMF	Scheduled maintenance fraction (CUCE definition).	0.
216.	SMI[17]	Module scrap fraction at General Support.	0.
217.	SMO[17]	Module scrap fraction at Direct Support.	0.
218.	SPE[18]	Fraction for controlling the sunk portion of the prime equipment cost. Any fraction may be used for SPE, SPEV, and SPEVR. SPE = 0 charges zero (sinks) the cost of the deployed prime equipment. SPE = 1 charges full cost for deployed equipment.	0.
219.	SPEV[18]	Factor to control sinking of cost of the initial provision. SPEV = 0 no cost for the initial allowance. SPEV = 1 charges full cost.	1.
220.	SPEVR[18]	Factor to sink costs for reordered material. SPEVR = 0 charges zero cost. SPEVR = 1 charges full cost.	1.
221.	STAT	The depot pipe in days between the depot and the rear-most facility shipping LRUs and modules to the depot.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
222.	SUD[19]	LRU scrap fraction at Depot.	0.
223.	SUE[19]	LRU scrap fraction at equipment level.	0.
224.	SUI[19]	LRU scrap fraction at General Support level.	0.
225.	SUO[19]	LRU scrap fraction at the Direct Support level.	0.
226.	SVE[20]	Salvage fraction for cost of installed LRUs at the end of the life of the program.	0.
227.	SVR[20]	Salvage fraction of the cost for consumed material.	0.
228.	SVT[20]	Salvage fraction of the cost for test equipment.	0.
229.	SVV[20]	Salvage fraction of the cost for residual inventory.	0.
230.	TALMAN	Number of test men per calibration crew.	0.
231.	TAT(1)[15]	An array of dimension four representing maintenance turn-around Time in days at Organization, Direct, General, and Depot maintenance support levels.	0.
232.	TAT(2)		0.
233.	TAT(3)		0.
234.	TAT(4)		0.
235.	TATE[15]	The number of days required for stock to be obtained at the equipment level.	0.
236.	TAYZ(1)	An array of dimension ten to specify correspondence between model availabilities and the LRUs. See section 3.4 for an explanation of the use of TAYZ.	1.
245.	TAYZ(10)		1.
246.	TC	Mean test time in hours to checkout an LRU at any level for detection of false "no go" LRUs. Used to compute demand for test manpower.	0.
247.	TD	Test time in hours for LRU checkout at depot. Used to compute demand for test manpower.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
248.	TDI	Sums with TID to form variable TIDT which sets the number of days of supply for LRUs and for repaired modules at the General Support level. If stock of LRUs is not designated at General Support, then TIDT sums with TEOT and TOIT in computing down-time in the availability calculations (RID).	0.
249.	TDMAN[21]	Manpower productivity factor or number of men per test crew at Direct Support.	0.
250.	TDMW	The mean time in hours spent in the test position at Depot per test sequence. The program assumes that this time will be spent twice. Once before a modification is made and once after the modification. This is also true for TIMW and TOMW.	0.
251.	TDPMI[21]	Manpower productivity factor or number of men per test equipment crew at Depot (for Type I test equipment).	0.
252.	TDPMII[21]	Manpower productivity factor or number of men per test equipment crew at Depot (for Type II test equipment).	0.
253.	TDPRI[21]	Manpower productivity factor or the number of men per repair crew at Depot for Type I test equipment.	0.
254.	TDPRII[21]	Manpower productivity factor or the number of men per repair crew at Depot for Type II test equipment.	0.
255.	TDR	Repair time in hours to repair an LRU. Used to compute demand at Depot.	0.
256.	TDRMAN	Manpower productivity factor or number of men per repair crew at Direct Support.	0.
257.	TE	Test time in hours for an LRU at equipment level. Used to compute the demand for test manpower.	0.
258.	TEMAN	Manpower productivity factor or number of men per test crew at equipment level.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
259.	TENMAN	The men applied to MTTR effort at equipment level. This is a multiplier of the number of eight hour shifts needed to perform the work.	0.
260.	TEO22	Pipelength in hours between equipment level and Direct Support when using LOGAM Supply Rules or expedited time for obtaining a spare when using LOGAM Maintenance Rules (definition of OL).	0.
261.	TER	Repair time in manhours for an LRU at equipment level. Used to compute the demand for repair manpower.	0.
262.	TERMAN	Manpower productivity factor or number of repairman per repair crew at equipment level.	0.
263.	TF	Mean time in hours to test an LRU at Direct Support. It is the total time per service action in the test position and it is used to set the demand for test equipment and for test equipment men.	0.
264.	TFR	Repair time in hours for an LRU at Direct Support. Used to compute demand for repair manpower.	0.
265.	TGMAN21	Manpower productivity factor or number of men per test crew at General Support.	0.
266.	TGRMAN21	Manpower productivity factor or number of repairmen per repair crew at General Support.	0.
267.	TI	Test time in hours for an LRU at General Support. Used to compute demand for test manpower.	0.
268.	TID	Sums with TDI to form variable TIDT which sets the number of days of supply for LRUs and for repaired modules at the General Support level. If stock or LRUs is not designated at General Support, then TIDT sums with TEOT and TOIT in computing down-time in the availability calculations (RID).	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
269.	TIMW	The mean time in hours spent in the test position at General Support per test sequence. The program assumes that this time will be spent twice: once before a modification and once after the modification.	0.
270.	TIO	Sums with TOI to make the variable TOIT, TOIT states the number of days of supply at Direct Support for LRUs (repaired or condemned) and modules which will be repaired. If LRU stock is not designated at Direct, then TOIT also adds additional down-time to TEOT in the computation of availability (ROI).	0.
271.	TIR	Repair time in hours of an LRU at General Support. Used to compute demand for repair manpower.	0.
272.	TMD	Test time in hours for module checkout at Depot. Used to compute demand for test manpower.	0.
273.	TMDD	The time in hours for modification kit installation per repair crew at Depot.	0.
274.	TMDR	Repair time in hours for a module at Depot. Used to compute demand for repair manpower.	0.
275.	TMI	Mean test in hours for module checkout at General Support. Used to compute demand for test manpower.	0.
276.	TMID	The time in hours for modification kit installation per repair crew at General Support.	0.
277.	TMIR	Repair time in hours for a module at General Support. Used to compute demand for repair manpower.	0.
278.	TMO	Mean test time in hours for module checkout at Direct Support. Used to compute demand for test manpower.	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
279.	TMOD	The time in hours for modification kit installation per repair crew at Direct Support.	0.
280.	TMOR	Repair time in hours for a module at Direct Support. Used to compute demand for repair manpower.	0.
281.	TOE22	Pipelength in hours between Direct Support and equipment level when using LOGAM Supply Rules used with TOE.	0.
282.	TOI	Sums with TIO to make the variable TOIT, TOIT states the number of days of supply at Direct Support for LRUs (repaired or condemned) and modules which will be repaired. If LRU stock is not designated at Direct, then TOIT also adds down-time to TEOT in the computation of availability (ROI).	0.
283.	TOMW	The mean time in hours spent in the test position at Direct Support per test sequence. The program assumes that this time will be spent twice: Once before the modification is installed and once after the modification is installed.	0.
284.	TONMAN	Number of men per contact support crew (Type IV test equipment).	0.
285.	TRC	Down-time in hours per service demand at equipment level (equivalent to MTTR).	0.
286.	TUMD	Used in concepts GN, GP, GQ, GS, and GT which call for LRU and module repair at Depot. TUMD sets the supply allowance in hours for modules at Depot to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.	0.

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
287.	TUMI	Used in concepts GM, GO, and GR which call for LRU and module repair at General Support. TUMI sets the supply allowance in hours for modules at General Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.	0.
288.	TUMO	Used for maintenance concepts GL where both LRU and module repairs are performed at Direct Support. TUMO sets the supply allowance in hours for modules at Direct Support to cover the time between removal of a module from an LRU until the module is repaired and returned to service for servicing further LRUs.	0.
289.	WD[23]	The scheduled work week in hours for test equipment at Depot.	168.
290.	WDM	The scheduled work week in hours for test crews at Depot.	168.
291.	WDR	The scheduled work week in hours for repair crews at Depot.	168.
292.	WE[23]	Scheduled work week in hours for test equipment at Organization.	168.
293.	WEM	Scheduled work week in hours for test crews at Organization.	168.
294.	WER	Scheduled work week in hours for repair crews at Organization.	168.
295.	WI[23]	The scheduled work week in hours for test equipment at General Support.	168.
296.	WIM	The scheduled work week in hours for test crews at General Support.	168
297.	WIR	The scheduled work week in hours for repair crews at General Support.	168.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
298.	WM	The shipping weight in pounds of a module.	0.
299.	WMR	The work week in hours for repair men performing TRC work on major items.	48.
300.	WMT	The work week in hours for Type V test equipment.	48.
301.	WO23	The scheduled work week in hours for test equipment at Direct Support.	168.
302.	WOM	The scheduled work week in hours for test crews at Direct Support.	168.
303.	WOR	The scheduled work week in hours for repair crews at Direct Support.	168.0
304.	WP	The shipping weight in pounds of a part.	0.
305.	WTKIT	The shipping weight in pounds of mod kit.	0.
306.	WU	The shipping weight in pounds of an LRU.	0.
307.	YAT	The annual attrition fraction for LRUs. It represents an annual demand for reissue and reprocurement to replace attrited LRUs. It operates on the population of installed LRUs to determine the number to be replaced each year. Within the program YAT is converted to "A", an hourly attrition rate. "A", in turn, is multiplied by OTF to get the real time rate.	0.
308.	YD	The length of the development phase of the program in years. It is only used in computing present value of costs incurred during a development phase (definition for FINT).	0.

SECTION 5 - INPUT DEFINITIONS (NAMELIST/L/)

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
309.	YMWO	The number of MWOs per year per LRU. YMWO is input as a percent per year of MWOs expected to be performed in the life cycle, i.e., if two MWOs are expected in a life cycle of 10 years, YMWO = 0.2.	0.
310.	YP	The length of the production or acquisition phase in years. It is used in computing the present value of costs incurred during the production phase. It is also used in estimating the initial production rate which is used as a reference rate in the main program in the computation of reorder buy quantities.	1.
311.	YR	The duration of the operation and maintenance portion of the program in years. Many of the cost computations for support are directly proportional to this input. It is also used in computing present value of operation and maintenance expenditures.	10.
312.	YZ	Input in the dimension of years and may be positive or negative. It is used in the computation of present value of costs to change the zero print of reference at which present value is started. The program treats YD, YP, and YR as consecutive non-overlapping time intervals. Nominally, present value is computed for the end of the production phase and the start of the operation and maintenance phase. YZ shifts this point by as many years ahead of or after it. Thus, if YZ equals the negative of YP, then present value is stated at the start of the production phase. If YZ is positive, it moves the point so many years into the O&M period from it start. Shifting YZ from LRU to LRU in the input sequence of LRUs being analyzed and using sunk cost input controls can accomplish, at present value, a time phasing of program cost totals.	0.
313.	ZFL^{24}	Round-off rule used in computing service channel quantities when integer round-off is invoked.	.9999

SENSY NO.	NAME	DESCRIPTION	DEFAULT VALUE
314.	ZI	Fraction of MWOs installed at General Support.	0.
315.	ZM(1)	An array of dimension four to specify the round-off fractions for modules at Equipment Direct, General, and Depot supply points (ZFL).	.99999
316.	ZM(2)		.99999
317.	ZM(3)		.99999
318.	ZM(4)		.99999
319.	ZO	Fraction of MWOs installed at Direct Support.	0.
320.	ZP(1)	An array of dimension three to specify the round-off fractions for parts at Direct, General, and Depot supply points (ZFL).	.99999
321.	ZP(2)		.99999
322.	ZP(3)		.99999
323.	ZU(1)	An array of dimension four to specify round-off fractions for LRUs at Equipment, Direct, General, and Depot supply points (ZFL).	.99999
324.	ZU(2)		.99999
325.	ZU(3)		.99999
326.	ZU(4)		.99999
	SENSY	An array dimensioned by 266 organized in the NAMELIST/L/ format to conduct sensitivity runs. Refer to section 3.3 for a description of sensitivity analysis.	0.
	T^{25}	Table of Organization and Equipment (TOE) dimensioned by 2000. The contents of this array are not used unless IOPER=1 is input and case totals are selected with NU<-1.	0.

5.2.2 NAMELIST/L/ INPUT DESCRIPTION FOOTNOTES

1. The safety stock coefficients are used in computing initial provision quantities when using LOGAM supply rules (AYZP = 0). In this instance, stock computations are based on the sum of the mean demand quantity plus the safety stock coefficient times the square root of the mean demand quantity. This quantity is rounded off according to a rule governed by the fractional value input for the ZU, ZM, and ZP arrays.

2. SMF and CUCE - "E" Level "Manpower" - LOGAM includes an equation for the computation of expected value of manpower at the equipment level (CMANE) in addition to the provisions of the model for the computation of integer values of contact support manpower using inputs CONMAN, TONMAN, CONTCT, etc. (Both methods may be used simultaneously when desired.)

   Input CUCE is the cost in dollars per year per equipment level team where a team means the number of men to give round-the-clock coverage. For example, if two men are needed to work together on any given problem and round-the-clock coverage requires four such sets of two men, the CUCE is the cost per year for eight men and should include any associated burden costs.

   The cost for these men will be based on the maintenance rate (CUCE) for which scheduled maintenance is required. Input SMF is the scheduled maintenance fraction. For instance, if one hour per operating day is necessary for scheduled maintenance with 250 operating days per year, 250 hours per year are required for scheduled maintenance, SMF = 250/8766 = .0285. If two men are required per team for one hour per day at $24273.00 per manyear, CMANE becomes SMF X CUCE = .0285 X $24273 X 2 = $1385.56.

3. ED may represent installations of aircraft, missile systems, tanks, communications, etc. deployed to perform a specific military function or mission.

4. Expected value flags may have only the values zero and unity. When set to unity, they give expected value (shared prorata utilization) of the service demand. When set to zero, they give integer round off, as governed by the round off input ZFL.

5. Discounting relates to the time value of money. It refers to the application of a selected rate of interest FINT such that future cost is adjusted to the present time. It also

recognizes that a dollar today is worth more than future dollars because of the interest cost that is related to expenditures which occur over time. Discounting is a technique for converting costs occurring over time to equivalent amount at a common point in time to facilitate comparison of alternative investments. The common point in time is set by the input YZ in LOGAM.

6. FMD, FMI, and FMO are module repair fractions. They specify the fraction of modules arriving at Depot, General Support, and Direct Support, respectively, that are repaired at these levels. The workflow of modules relates to the maintenance policy (G fraction). Modules not repaired at a lower echelon may be sent to a higher echelon if the maintenance policy allows it, otherwise they are scrapped.

7. FN is used in computing availability. It is a statement of the number of permissible failures at an installation before down-time counts. For redundant items, for example, FN = 1. If all failures count, FN = 0. The availability computation forgives FN failures, so to speak, before counting down-time. For a given LRU, input EE (a whole number) states the number of times a given LRU is replicated (used) per ED location. Input FN states the number of LRUs within EE whose equipment failure does not detract from system availability. For example, if there are two radio receivers per tank for the sake of redundancy of the system and one receiver out of service is permitted, then EE = 2 and FN = 1. When FN = EE, all LRUs are permitted out of service and the LRU has no role in the system availability. EE and FN must be input as whole real numbers.

8. If FNGF equals unity, the false no-go rate will equal the true failure rate specified by E. In use in the program, FNGF is commutative with the input OTF. Thus OTF modifies the false no-go rate so that it is a real time rate.

9. The purpose of FNSP is to account for supply administration cost for new items to be introduced in the supply system which are unique to the system under analysis. Inputs CEN and CAD are charged for FNSP times input PP as supply administration. FNSP=1 charges all parts to supply administration and FNSP=0 would delete the cost. Intermediate values act directly.

10. FUD, FUE, FUI and FUO are the LRU repair fractions. They specify the fraction of LRUs arriving at Equipment Level, Direct Support, General Support, and Depot, respectively, that are repaired at these levels. Like the module repair fraction FMO, FMI, and FMD, the workflow of LRUs relates to the maintenance policy (G fraction). LRUs not repaired at a maintenance echelon are scrapped.

In all cases, the term LRU repair includes detection of false no-go items unless it has been preceded by a checkout at the Direct Support level. Also, whenever module repair succeeds LRU repair, any LRUs not repaired at the lower level will also go to the module repair facility for LRU repair. Thus, for example, for maintenance policy GP any LRUs not repaired at the General Support will go to Depot for repair. In all cases, the degree of repair performed at any level either on LRUs or on modules will be set by other input repair fractions. Whenever LRU repair is indicated, the program computes the module stock required to perform LRU repair by module replacement. Similarly, whenever module repair is indicated, the program computes the part stock required to perform module repair. Parts are always nonrepairable and are discarded.

The specification of a maintenance concept, input by the "G" fractions, may be mixed in any proportion summing to unity to represent the flow of work demand. For example, if:

GL = 0.6, GR = 0.25, GT = 0.15

Then, 60% of the LRU removals would be sent to Direct Support for repair, 25% would be sent to General Support for repair, and the remaining 15% would go to Depot. If FUO = .8 then eighty percent of the total LRUs arriving at Direct Support would be repaired and twenty percent scrapped. Similarly FUI = .8 and FUD = .8 would act for General Support and Depot.

11. LOGAM permits four levels of LRU supply. In the program these are the equipment, Direct, General, and Depot support locations. Array H is used to specify for each level whether or not LRU spares are permitted. Any combination from no supply locations to all four is permitted. Array H has dimension four. The first element is for the E level. The second element is for the Direct Support level. The third element is for the General Support level. The fourth element is for the Depot level.

Zero means not authorized. Unity means "yes" stock is authorized. Only zero or unity are to be used.

H is input via a NAMELIST/L/. Typically,

H = 0.,3*1.,

allows LRU supply at Direct Support, General Support, and Depot levels only. The input

H = 1.,3*0.,

allows LRU spares at the equipment (E level) and denies them to the Direct Support, General Support, and Depot levels.

The program will inspect the inputs QTE, QTO, QTI, and QTD to see if stock quantities have been input. If they have been input, the corresponding H element will be set to unity even if input as zero. This change to H, if made, is permanent until H is again input with some subsequent LRU.

12. NA is used in combination with the input TAYZ (the availability tally control). In LOGAM, there are ten availability accumulators, therefore, it is possible to take up to ten availability products for different sets of LRUs. NA is the input which specifies how many of the ten accumulators are active. Section 3 contains a discussion of the availability accumulator.

13. In LOGAM, all program inputs obtain initial values in a BLOCK DATA sub-program. All inputs are stored in an array immediately after the read of input data for the first LRU either from BLOCK DATA or NAMELIST/L/. Input of IS=1 with one LRU recalls the list of saved values prior to the read of the next LRU. Thus, the set of inputs for the first LRU including the standard values not input via NAMELIST/L/ become the "reset" standard values. These "reset" standard values may be redefined at any time throughout the program by use of the control NB. NB is in NAMELIST/L/ and may be input with any LRU. The exact value NB = 0 (an integer) will force the storing of the current data set for that LRU as the new set of "reset" values. NB is set to 1 during this storing and the input need not be turned off by the user.

14. DDS, DIS, EDS, and ODS determine the number of stock locations at each echelon. These inputs work with the H Array to designate if stock is permitted at a location. If DIS = 3, meaning three General Support stock locations, and H(3) = 1 stock will be located at all three General Support locations. If H(3) = 0 stock will be denied. If DIS = 3 and H(3) = 0 stock will be denied. The objective of this is that an alternate policy of no stock at that location. Simply changing the H Array permits quick turn-around.

15. The input AYZP = 1 activates the use of the LOGAM maintenance rules. When using the LOGAM maintenance rules, four maintenance turn-around times are provided.

TATE - Used for time required to obtain an LRU or module (based on maintenance policy) from "ED" stock or support facility.
TAT - Array of dimension four used for maintenance turn-

around-time at Organization, Direct Support, General Support and Depot, respectively.

TATE and TAT are input in days. According to Array H setting authorized supply points, the contents of the repairable pipelines are computed using these maintenance turn-around times. Consumables are supplied according to the Operating Level (OL), Safety Level (SL), and Order Ship Times (OST). (The last is also used at the Depot level for repairables.)

OL, SL, and OST are arrays of dimension four. The order of each array designates the days of supply allowance for Organization, Direct Support, General Support, and Depot, respectively. The total content of the repairable and consumable pipelines is computed for LRUs, modules, and parts.

The program attempts to pass this quantity out to the authorized supply points beginning with the forward-most location. It integerizes at each location using the round-off rules. After each point, a test is made to see if the entire demand has been equaled or exceeded. If it has been met, no further quantities are computed. This prevents oversupply of stock on top of stock. The concept is that all stock is under the control of the NICP and that stock will be directed to where needed from where stocked.

Note that when the LOGAM maintenance rules are used to compute the initial provision, the inputs TEO, TOE, TOI, TIO, etc. pipeline times are used for the computation of availability. These times specify the "down-times" consequence of a stock outage. The time should be the maximum time, as the model will adjust the time for the fullness of the pipeline.

Thus, TAT, OL, SL, OST, STAT, DTE, DTI, and DTO never enter directly into the availability calculation. The effect of these times is the computation of an integer number of spares. The number of spares enters into the computation of availability. In this way the user of the model may input policy times for setting the supply levels and input expedited times for the consequences of supply outage on availability.

16. QTE, QTO, QTI, QTD, QTME, QTMO, QTMI, QTMD, QTPO, QTPI, QTPD - Stock Quantities: In LOGAM, these stock quantities may be input at any time regardless of the value of AYZP. LOGAM sets each of these to zero just prior to the read of the input NAMELIST. If any one is input, it will be used as input instead of being computed. It is the responsibility of the user to input values compatible with his concept (GA

through GT), i.e., unless Direct Support is performing repair to the piece part level, it would be meaningless to input a value for QTPO. However, such an erroneous input would be accepted and used by the program.

After the read of the input NAMELIST, THE LRU stock quantities are inspected to see which are non-zero. If any are non-zero and this is inconsistent with the input Array H (Page), the corresponding values of Array H are altered. For example, if Array H has been input to prohibit LRU spares at Direct and QTO is input giving LRU spares to Direct, then Array H is permanent until altered by some subsequent input of Array H via NAMELIST with some subsequent LRU. When AYZP has a fractional part to call for a force on availability, the forward-most LRU stockpile will be increased if necessary to try to meet the specified availability. In the event that the initial quantity for the forward-most supply has been input, it will be subject to revision upwards. When the fractional part calls for a decrease in availability, the rearward-most supply will be revised.

17. The scrap fractions, SME, SMO, SMI, and SMD are applied to the work flow sent to Organization, DS, GS, and Depot by the maintenance policy "G" fractions prior to any application of the repair fractions FMO, FMI, and FMD. Thus, the total module scrap is the flow arriving at a maintenance point times the scrap fraction plus the remainder that are not scrapped but are not repaired as set by the repair fraction (if the latter are not sent on to a higher maintenance level).

18. LOGAM includes equations for the cost of prime equipment CEP, the cost of supply material CIVP, and the cost of consumed material CIVR. The input factors SPE, SPEV, and SPEVR appear in these equations as multipliers. Assigning values to the inputs less than unity, therefore, reduces the value of the cost equations or in effect sinks some portion of the cost which would otherwise be charged for materials.

19. Module scrap fractions are for SMD, SME, SMI, and SMO. The same definitions apply for the LRU scrap fractions; however, it is also noted that the scrap fractions apply only to failure flow and not to false no-go flow.

20. Within the LOGAM program, a salvage computation is made on four types of equipment: installed LRUs, consumed material, test equipment, and residual inventory. The salvage fractions are used as multipliers in functions that are signed negative to reflect the sense of credit. Thus, the salvage terms are taken as some fraction of the costs for the various types of equipment.

21. In LOGAM, manpower may be input as shared or dedicated according to the value input for the expected value flags (EVDM). When shared manpower is used, the inputs such as TDMAN represent the manpower productivity to account for less than full time utilization of the maintenance manpower. Factors greater than one are input which in effect act as multipliers on the cost for manpower. For Depot and General Support, TGMAN, TGRMAN, TDPMI, TDPMII, TDPRI and TDPRII may be adjusted to account for maintenance civilian labor costs (Item 3.051 and DA PAM 11-4).

22. When LOGAM supply rules are used (AYZP = 0), TEO is used in conjunction with the input TOE to set the down-time per failure or false no-go or attrited item returned from an installation to Direct Support. (The return may be for repair, supply, or material transfer.) This down time is used to compute one of the terms in the LOGAM availability formulation and as a minimum at least this much down time is occasioned at each support demand by a unit. It is the sum of TEO, and TOE that is used in the program; they are never used separately. In particular, TEO might represent the time for a contract support team to go to an installation.

23. The work weeks are set in accordance with the corresponding expected value controls. When the expected value control is set to zero, the program acts in an integer round off computation mode for the service channel requirements. In this mode, one is cautioned that excessively long work weeks can lead to queues which are not computed within this program, i.e., if indeed there were a work demand requiring work 168 hours per week and the work week input as 168 hours per week, i.e., repair rate equals demand rate, then the queue would, in general, be long. When the expected value control for manpower is set to one, the work week should correspond to the manpower salary scale, i.e., if the salary scale is on the basis of a 40 hour week, then the work week should be input as 40 hours.

24. The round-off rules ZFL, ZM, ZP, and ZU all act in the same manner. The values input are added to the demands computed by the program and then the fractional part is dropped and the whole number is retained. This done to avoid acquisition of fractional portions of test equipment, LRU, modules and parts.

25. TOE data are input into the T(2000) array. The array is dimensioned to store data for 200 line items with a maximum of 10 input values per item. The meaning of each value is determined by the position of the input in the string and the values of the first (and in some cases the second) item in each 10 element field. The possible inputs for the first

two values and a general description of the use of the associated data field is shown on Table 4. The first value dictates which post processor routine is called to evaluate the inputs in that 10 element data field. For example a value of 3 will cause a reference to Subroutine THREE from Subroutine OPER to compute Equipment/Fuel usage costs. The second value (when applicable) determines the section of logic within a post-processor routine that will be used in the evaluation.

Table 5 includes a list of the possible input values and the mnemonics for the 10 element input fields. The mnemonics are not used in the NAMELIST inputs but are used as a reference only. In the fields where there are no mnemonics, no inputs are required. Reference Table 6 for a description of the mnemonics in Table 5.

The tables referenced above as well as a description of TOE inputs and cost evaluations can be found in section 3.6.

SECTION 6

PROGRAM LOGAM OUTPUT DEFINITIONS

SECTION 6 - OUTPUT DEFINITIONS (INITIAL PROVISIONS)

6.1 LOGAM Provisioning Output

This section describes the initial provisioning outputs for LRUs, modules and parts. These outputs are printed by Subroutine IPVOP which in turn is called from Subroutine IDVOP. IDVOP is referenced by LOGAM (main program) if the individual LRU case print flag (INHIB=0) is input through NAMELIST/L/.

The supply echelons for which provisions are output are Organizational (ED), Direct Support (OD), General Support (DI), and Depot (DD). For LRUs there are three columns of output. Column one is the provisioning computed as a function of the maintenance flows and their respective time charges. Column two is the required provisioning to meet a predetermined operational availability. Columns one and two are identical if an operational availability is not requested. The sample output for this section (8.1) does not show it, but the requested availability would be printed on this page if a request had been made. Column three is the required LRU provisioning after distributing the stock evenly over the number of stock points at a supply echelon. Only LRUs are effected by the availability request, therefore only computed and distributed values for module and part provisions are output.

The number of stock points, stock authorization flags and maintenance policies are additional outputs to this page. The reference numbers for the following variables correspond to numbers marked on the sample output page of section 8.1.

The following outputs are page header descriptions that are printed from subroutine PAGE.

1. IPAGE — Output page number.

2. TEXT — Four cards of input that describe the logistics scenario.

4. ANLYIS — Input description for the type of analysis.

6. DATE — Input date.

The next 2 outputs are page header descriptions also but they are printed by IPVOP rather than PAGE.

3. UNITIS — Input description for the individual LRU class and number.

5. REMARK — Input remark for the individual LRU case.

The following outputs are LRU provisions. The column labled "COMPUTED" are outputs for LRUs computed as a functions of the

maintenance flows and their respective time charges. The "REQUIRED" column outputs are the computed values after adjustments are made for an operational availability requirement. The DISTRIBUTED column outputs are the LRUs required after distributing evenly over all supply points and rounding to the nearest integer as dictated by the "Z" round-off factors.

7. PDUE — Computed LRU provisioning at the organizational level.

8. PQTE — Required LRU provisioning at the organizational level.

9. QTE — Distributed LRU provisioning at the organizational level.

16. PDUO — Computed LRU provisioning at the Direct Support level.

17. PQTO — Required LRU provisioning at the Direct Support level.

18. QTO — Distributed LRU provisioning at the Direct Support level.

25. PDUI — Computed LRU provisioning at the General Support level.

26. PQTI — Required LRU provisioning at the General Support level.

27. QTI — Distributed LRU provisioning at the General Support level.

34. PDUD — Computed LRU provisioning at the Depot level.

35. PQTD — Required LRU provisioning at the Depot level.

36. QTD — Distributed LRU provisioning at the Depot level.

43. PQTU — Total of computed LRUs for all supply echelons.

44. PQT — Total of required LRUs for all supply echelons.

45. QT — Total of distributed LRUs for all supply echelons.

The following outputs are the module provisions. The "COMPUTED" column are outputs computed as a function of the maintenance flows and their respective time charges. The DISTRIBUTED column are outputs for modules distributed evenly over the supply points

at an echelon.

10. PDME Computed module provisioning at the Organizational level.

11. QTME Distributed module provisioning at the Organizational level.

19. PDMO Computed module provisioning at the Direct Supply level.

20. QTMO Distributed module provisioning at the Direct Supply level.

28. PDMI Computed module provisioning at the General Support level.

29. QTMI Distributed module provisioning at the General Support level.

37. PDMD Computed module provisioning at the Depot level.

38. QTMD Distributed module provisioning at the Depot level.

46. PQTM Total of computed modules for all supply echelons.

47. QTM Total of distributed modules for all supply echelons.

The following outputs are the part stock provisions. The "COMPUTED" column are outputs computed as a function of maintenance flows and their respective time charges. The DISTRIBUTED column are outputs for parts distributed evenly over all supply points at an echelon.

12. PDPE LOGAM does not compute stock for parts at the Organizational level. This is a dummy variable used to fit the formats for LRUs and modules.

13. QTPE LOGAM does not compute stock for parts at the Organizational level. This is a dummy variable used to fit the formats for LRUs and modules.

21. PDPO Computed part provisioning at the Direct Support level.

22. QTPO Distributed part provisioning at the Direct Support level.

SECTION 6 - OUTPUT DEFINITIONS (INITIAL PROVISIONS)

30. PDPI — Computed part provisioning at the General Support level.

31. QTPI — Distributed part provisioning at the General Support level.

39. PDPD — Computed part provisioning at the Depot level.

40. QTPD — Distributed part provisioning at the Depot level.

48. PQTP — Total of computed parts for all supply echelons.

49. QTP — Total of distributed parts for all supply echelons.

The following outputs are the stock points at each echelon and the stock authorization flags.

14. EDS — Number of supply points at the Organizational level.

23. ODS — Number of supply points at the Direct Support level.

32. DIS — Number of supply points at the General Support level.

41. DDS — Number of supply points at the Depot level.

15. H(1)
24. H(2)
33. H(3)
42. H(4) — Stock authorization flags at Organization, Direct Support, General Support and Depot, respectiely. The values are always unity for authorization and zero if stock is not authorized.

The following outputs are the maintenance policies. It is these flags that determine the values entered under the "COMPUTED" columns of output described above. The sum of the values for all maintenance policies must equal to unity (100 percent).

50. G — Twenty maintenance policies that are equivalenced to variables GA through GT.

The following output is printed only if an Operational Requirement (OR) for availability has been requested with the AYZP flag. The fractional part of AYZP is stored into OR and OR becomes the operational availability to be obtained by either adding or substracting LRUs from the initial provisions. If OR is greater than 0.5 the value of OR will be printed.

51.	OR	Over-Ride factor for operatinal availability. This is the fractional part of input AYZP. Refer to sample output section 8.7.2 for an example of this output.

SECTION 6 - OUTPUT DEFINITIONS (INDIVIDUAL LRUs)

6.2 Individual LRU Outputs

This section describes the outputs for an individual LRU case. These outputs are printed when Subroutine IDVOP is called from the main program (LOGAM). IDVOP is called when INHIB=0 is input through NAMELIST/L/. The outputs in this section are for costs, test and repair manpower, test equipment demands and comparions between present value and expected value costs. The reference numbers for the following outputs variables correspond to the numbers marked on the sample output page in section 8.2.

The following variables are page header descriptions that are printed when Subroutine PAGE is called.

1. IPAGE — Output page number.
2. TEXT — Four cards of input that describe generally the Logistics scenario.
3. ANLYIS — Input description for the type of analysis.
4. DATE — Input date.

The next 2 outputs are page header descriptions also but they are printed by IDVOP rather than PAGE.

5. UNITIS — Input description for the individual LRU class and number.
6. REMARK — Input remark for the individual LRU case.

The following output variables are the individual LRU case costs and availabilities. The availabilities are computed in Subroutine AVAIL and the costs are outputs from Subroutine LGCST.

7. COSTIS — Thirty six characters of input information describing the cost units.
8. AYZOS — Operational availability.
9. AYZIS — Inherent availability.
10. CET — Development cost for prime equipment.
11. CTST — Development cost of test equipment.
12. CFT — Cost of housing test equipment.
13. CMPT — Manpower cost for test and repair personnel, including training.

SECTION 6 - OUTPUT DEFINITIONS (INDIVIDUAL LRUs)

14. CIVT — Acquisition cost of initial and replenishment spares.

15. CROT — Cost of reordering spares.

16. CWHT — Cost of storing spares.

17. CSAT — Cost of entering and maintaining line items in supply system.

18. CSHT — Cost of shipping spares and mod kits.

19. GCT — Grand total cost.

The following output variables include the quantities for initial buys, reorders, consumables and residuals. The initial provisions are the distributed totals from Subroutine STKDS. Reorders, consumables and residuals are evaluated in Subroutine REORDR.

20. EDEE — Equipment (LRUs) installed at all ED locations.

21. QT — Initial LRU stock provisions.

22. RQN — Number of LRU reorder transactions.

23. QB — LRU reorder lot size.

24. QC — Quantity of LRUs reordered during O&S phase.

25. QQC — Quantity of LRUs consumed during O&S phase.

26. RU — Residual LRUs at end of O&S phase.

27. QTM — Initial module stock provisions.

28. RQMN — Number of module reorder transactions.

29. QBM — Module reorder lot size.

30. QCM — Quantity of modules reordered during O&S phase.

31. QQCM — Quantity of modules consumed during O&S phase.

32. RM — Residual modules at end of O&S phase.

33. QTP — Initial part stock provisions.

34. RQPN — Number of part stock reorder transactions.

35. QBP — Part reorder lot size.

SECTION 6 - OUTPUT DEFINITIONS (INDIVIDUAL LRUs)

36. QCP — Quantity of parts reordered during O&S phase.

37. QQCP — Quantity of parts consumed during O&S phase.

38. RP — Residual parts stock at end of O&S phase.

The following output variables are test and repair manhour demands for each individual LRU case. These outputs represent the manhours for the effective number of installations supported by one maintenance facility. These outputs are evaluated in Subroutine TEMWF.

39. AEY
40. AERY — Test and repair manhour demands for an Organizational service channel.

41. AOY
42. AORY — Test and repair manhour demands for a Direct Support service channel.

43. AIY
44. AIRY — Test and repair manhour demands for a General Support service channel.

45. ADY
46. ADRY — Test and repair manhour demands for a Depot service channel.

The following output variables are the cumulative values for the individual LRU case manhour demands described above. These outputs are accumulated for groups of LRU cases whose manpower is to be shared. The LRU groups are determined by the Type I test equipment posting flag (ETI). While ETI=0, the manhour demands are accumulated. Once ETI=1 is input the cumulative values are posted and the values are reset to zero to begin a new accumulation with the next LRU case. These outputs are evaluated in Subroutine TEMWF.

47. ILRU
48. NLRU — The first and last individual LRU case number for the group of LRU cases whose manpower is being shared.

49. SAEY
50. SAERY — Cumulative test and repair manhour demands at an Organizational service channel.

51. SAOY
52. SAORY — Cumulative test and repair manhour demands at a Direct Support service channel.

53. SAIY
54. SADRY — Cumulative test and repair manhour demands at a General Support service channel.

55. SADY
56. SADRY — Cumulative test and repair manhour demands at a Depot service channel.

SECTION 6 - OUTPUT DEFINITIONS (INDIVIDUAL LRUs)

The following output variables are cumulative values of the manhour demands for all individual LRU cases in a case concept. These outputs differ from the accumulations above in that the ones above can be in groups for a case concept. If the flag ETI=1 is input with the last LRU case in a concept then these variables will have the same value as the variables for the previous accumulations. A case concept is completed by inputting IS=1 with the last individual LRU case for that concept. These outputs were accumulated in Subroutine TEMWF.

57. CAEY
58. CAERY — Cumulative test and repair manhour demand for an Organizational service channel.

59. CAOY
60. CAORY — Cumulative test and repair manhour demand for a Direct Support service channel.

61. CAIY
62. CAIRY — Cumulative test and repair manhour demand for a General Support service channel.

63. CADY
64. CADRY — Cumulative test and repair manhour demand for a Depot service channel.

The following variables are the outputs for Type I test equipment manpower and test equipment demands. This output is available only when the posting (ETI=1) of Type I test equipment is requested. The values here represent the demands for the effective number of installations supported by one maintenance facility. Test manpower demands include the factor (FI) for test equipment self-support. These outputs are evaluated in Subroutine TEMWF.

65. ILRU
66. NLRU — The first and last individual LRU case numbers for the group of LRU cases whose manpower is being shared using Type I test equipment.

67. ESUY
68. ESURY — Posted cumulative test (AEY) and repair (AERY) manhour demands at one Organizational service channel.

69. ESU
70. ESUM
71. ESUR — Demands for Type I test equipment, test manpower, and repair manpower at one Organizational service channel.

72. DSUY
73. DSURY — Posted cumulative test (AOY) and repair (AORY) manhour demands at one Direct Support service channel.

74. DSU
75. DSUM
76. DSUR — Demands for Type I test equipment, test manpower, and repair manpower at one Direct Support service channel.

77. GSUY
78. GSURY — Posted cumulative test (AIY) and repair (AIRY) manhour demands at one General Support service channel.

79. GSU
80. GSUM
81. GSUR — Demands for Type I test equipment, test manpower, and repair manpower at one General Support service channel.

82. DEPY
83. DEPRY — Posted cumulative test (ADY) and repair (ADRY) manhour demands at one Depot service channel.

84. DEP
85. DEPM
86. DEPR — Demands for Type I test equipment, test manpower, and repair manpower at one Depot service channel.

The following variables are the outputs for manpower and test equipment demands at Depot service channels having Type II test equipment. The factor (FII) for Type II test equipment self-support is included with the cumulative repair manhours (SADY). This output is available only if the posting flag ETII=1. The values printed here can either be the expected values or integer rounded. The expected value flags input in namelist /L/ determine if the values are rounded according to the ZFL parameter.

87. ILRU2
88. NLRU2 — The first and last individual LRU case numbers for the group of LRU cases whose manpower is being shared at the Depot using Type II test equipment.

89. DEPAIY
90. DEPARY — Posted cumulative test (SADY) and repair (SADRY) manhour demands for Type II test equipment at Depot service channel.

91. DEPAIE
92. DEPAIM
93. DEPAR — Demands for Type II test equipment, test manpower, and repair manpower at Depot service channel.

The following variables are the outputs for the cost of initial provisioning and the salvage value for unconsumed provisions at end of O&S phase. Provisioning costs are provided at each supply echelon.

94. CQTE — Cost of initial LRU provisions at Equipment level.

95. CQTO — Cost of initial LRU provisions at Direct Support level.

96. CQTI — Cost of initial LRU provisions at General

Support level.

97. CQTD — Cost of initial LRU provisions at Depot level.

98. CQTT — Total cost of initial LRU provisioning.

99. CRUT — Salvage value for the LRU provisions that are unconsumed at end of O&M.

100. CQTME — Cost of initial module provisions at the Equipment level.

101. CQTMO — Cost of initial Module provisions at Direct Support level.

102. CQTMI — Cost of initial Module provisions at General Support level

103. CQTMD — Cost of initial Module provisions at Depot level.

104. CQTMT — Total cost of initial Module provisioning.

105. CRMT — Salvage value for the Module provisions that are unconsumed at end of O&M.

106. CQTPO — Cost of initial Parts provisions at Direct Support level.

107. CQTPI — Cost of initial Parts provisions at General Support level.

108. CQTPD — Cost of initial parts provisions at Depot level.

109. CQTPT — Total cost of initial parts provisioning.

110. CRPT — Salvage value for the initial Part provisions that are unconsumed at end of O&M.

The following variables are the outputs for present and expected value cost totals and the deltas between expected cost and dedicated manpower costs.

111. PVCGT
115. EPVGCT — Present and expected value cost totals for an LRU case.

112. PCGT
116. SEPV — Cumulative present and expected value cost totals for all LRU cases currently evaluated.

113. CMPT
117. ECMPT — Present and expected value manpower costs.

SECTION 6 - OUTPUT DEFINITIONS (INDIVIDUAL LRUs)

114. PDELTA The present value for DELTA.

118. DELTA The difference between dedicated cost value and expected cost value.

SECTION 6 - OUTPUT DEFINITIONS (SUMMARIZED CASES)

6.3 Summarized LRU Case Outputs

These outputs use essentially the same format statements used in the individual LRU outputs. The initial provisions at the supply echelons are printed on the same page as the costs and not on a separate page as the individual LRU outputs are. These outputs are printed by Subroutine IDVOP when IDVOP is called from LOGAM with the option flag IL=2. The outputs printed here were summarized into the "C" array in Subroutine SUMRY. The order of individual LRU cases that make up a summarized case total is determined by the NDLRU input parameter from LOGAM. As an example, if NDLRU=11, then the 1st and 12th, 2nd and 13th, etc., LRU cases are summarized for the outputs here. The input flag IFLAG must be equal to zero for the values to be summarized or printed. The numbers assigned to the variables below correspond to the numbers marked on the sample output page of section 8.3. The first 93 numbers correspond to the numbers for the individual LRU case descriptions in section 6.1. Because of printing inital provisions on this page, the numbers 105 through 129 of this output will equate to numbers 94 through 118 of the individual LRU outputs. There will be one page of output printed for each NDLRU summarizations.

The following variables are page header descriptions printed by Subroutine PAGE.

1. IPAGE — Output page number that is printed from Subroutine PAGE after a call from LOGAM.

2. TEXT — Four cards of input that describe generally the Logistics scenario.

4. ANLYIS — Input description for the type of analysis.

6. DATE — Input date.

The following variables are the LRU unit description and a summarization description. These outputs are page headers that are printed by IDVOP and not by PAGE. The unit description is the description (UNITIS) input for the first LRU case in a summarization. The summarization title is input along with NDLRU on the 8th input record.

3. UNS — Input description for the individual LRU class and number.

5. TLRU — Up to 60 characters of input data to label output for summarized LRU's.

7. COSTIS — Thirty six characters of information describing the cost units.

8. Not used.

9. Not used.

10. C(3) Summed values for development cost of prime equipment (CET).

11. C(4) Summed values for development cost of test equipment (CTST).

12. C(5) Summed values for cost of housing test equipment (CFT).

13. C(6) Summed values for test and repair manpower cost (CMPT).

14. C(7) Summed values for cost of spares (CIVT).

15. C(8) Summed values for cost of reordering spares (CROT).

16. C(9) Summed values for cost of storing spares (CWHT).

17. C(10) Summed values for cost of entering and maintaining line items (CSAT).

18. C(11) Summed values for cost of shipping spares (CSHT).

19. C(12) Summed values for grand total cost (GCT).

The following outputs include the summarized quantities of spares, consumed material and residuals.

20. C(110) Summed values for installed LRUs (EDEE).

21. C(13) Summed values for initial LRU provisions (QT).

22. C(14) Summed values for the number of LRU reorder actions (RQN).

23. QB LRU reorder lot size. Has the value from the last individual LRU case.

24. C(15) Summed values for LRU replenishment spares (QC).

25. C(16) Summed values for the consumed LRUs during O&S (QQC).

26. C(17) Summed values for the residual LRUs (RU).

27. C(18) Summed values for initial Module provisions (QTM).

28. C(19) Summed values for the number of Module reorder actions (RQMN).

29. QBM Module reorder lot size. Has the value from the last individual LRU case.

30. C(20) Summed values for the Module replenishment spares (QCM).

31. C(21) Summed values for the Modules consumed during O&S (QQCM).

32. C(22) Summed values for the residual Modules (RM).

33. C(23) Summed values for initial part provisions (QTP).

34. C(24) Summed values for the number of part reorder actions (RQPN).

35. QBP Part stock reorder lot size. Has the value from the last individual LRU case.

36. C(25) Summed values for part replenishment spares (QCP).

37. C(26) Summed values for parts consumed during O&S (QQCP).

38. C(27) Summed values for the part residuals (RP).

The following outputs are summarized test and repair manhour demands. These outputs represent the number of manhours required at one maintenance facility to support all equipment installatins (ED).

39. C(28)
40. C(29) Summed values for test (AEY) and repair (AERY) manhour demands at an Organizaitnal service channel.

41. C(30)
42. C(31) Summed values for test (AOY) and repair (AORY) manhour demands at a Direct Support service channel.

43. C(32)
44. C(33) Summed values for test (AIY) and repair (AIRY) manhour demands at a General Support service channel.

45. C(34)
46. C(35) Summed values for test (ADY) and repair (ADRY) manhour demands at a Depot service channel.

The following outputs are the summarized cumulative values for the individual LRU case manhour demands. These demands are accumulated for groups of LRU cases whose manpower is to be shared. The posting of these demands is controlled by ETI, the Type I test equipment posting flag.

47. KKK
48. KKK Order of the LRU cases summarized.

49. C(36)
50. C(37) Summed values for cumulative test (SAEY) and repair (SAERY) manhour demands at an Organization service channel.

51. C(38)
52. C(39) Summed values for cumulative test (SAOY) and repair (SAORY) manhour demands at a Direct Support service channel.

53. C(40)
54. C(41) Summed values for cumulative test (SAIY) and repair (SAIRY) manhour demands at a General Support service channel.

55. C(42)
56. C(43) Summed values for cumulative test (SADY) and repair (SADRY) manhour demands at a Depot service channel.

The following outputs are the summarized cumulative values for all individual LRU cases. These outputs differ from the ones above in that the ones above can be accumulated for groups of LRUs by using the posting flag (ETI). If ETI=1 is set for the first time in the last LRU case, the values for these two output sections will be identical.

57. C(44)
58. C(45) Summed values for cumulative test (CAEY) and repair (CAERY) manhour demands at an Organization service channel.

59. C(46)
60. C(47) Summed values for cumulative test (CAOY) and repair (CAORY) manhour demands at a Direct Support service channel.

61. C(48)
62. C(49) Summed values for cumulative test (CAIY) and repair (CAIRY) manhour demands at a General Support service channel.

63. C(50)
64. C(51) Summed values for cumulative test (CADY) and repair (CADRY) manhour demands at a Depot service channel.

The following outputs are the summarized cumulative values for

Type I test equipment manpower and test equipment demand. This output is printed only when the posting (ETI=1) of Type I test equipment data is requested for any one of the individual LRU cases. Whenever ETI=1 is input, STI=1 is set in LOGAM which flags IDVOP to print these outputs.

65.	ILRU	The first and last individual LRU case numbers
66.	NLRU	for the group of LRU cases whose manpower is being shared using Type I test equipment.
67.	C(52)	Summed values for posted cumulative test (ESUY)
68.	C(53)	and repair (ESURY) manhour demands at an Organizational service channel.
69.	C(54)	Summed values for posted cumulative test
70.	C(55)	equipment demand (ESU), test (ESUM) and repair
71.	C(56)	(ESUR) manpower at an Organizational service channel.
72.	C(57)	Summed values for posted cumulative test (DSUY)
73.	C(58)	and repair (DSURY) manhour demand at one Direct Support service channel.
74.	C(59)	Summed values for posted cumulative test
75.	C(60)	equipment demand (DSU), test (DSUM) and repair
76.	C(61)	(DSUR) manpower at a Direct Support service channel.
77.	C(62)	Summed values for posted cumulative test (GSUY)
78.	C(63)	and repair manhour demand at a General Support service channel.
79.	C(64)	Summed values for posted cumulative test
80.	C(65)	equipment demand (GSU), test (GSUM) and repair
81.	C(66)	(GSUR) manpower at a General Support service channel.
82.	C(67)	Summed values for posted cumulative test (DEPY)
83.	C(68)	and repair (DEPRY) manhour demand at a Depot service channel.
84.	C(69)	Summed values for posted cumulative test
85.	C(70)	equipment demand (DEP), test (DEPM) and repair
86.	C(71)	(DEPR) manpower at a Depot service channel.

The following outputs are the summarized cumulative values for Type II test equipment at the Depot. This output is printed only if the posting flag ETII=1 is input with an LRU case. Whenever ETII=1 is input, STII=1 is set by LOGAM to flag IDVOP to print these outputs.

89. C(72)
90. C(73) Summed values for posted cumulative test (DEPAIY) and repair (DEPARY) manhour demand at a Depot service channel.

91. C(74)
92. C(75)
93. C(76) Summed values for posted cumulative test equipment demand (DEPAIE), test (DEPAIM) and repair (DEPAR) manpower at a Depot service channel.

The following outputs are the summarized provisioning quantities at each supply echelon. This output is the only difference between the format used for individual LRU cases and summarized cases. For individual cases the initial provisions are printed on a separate page.

94. C(82)
95. C(83)
96. C(84)
97. C(85) Summed values for initial LRU provisions at Equipment (QTE), Direct Support (QTO), General Support (QTI) and Depot (QTD).

98. C(86)
99. C(87)
100. C(88)
101. C(89) Summed values for initial Module provisions at Equipment (QTME), Direct Support (QTMO), General Support (QTMI) and Depot (QTMD).

102. C(90)
103. C(91)
104. C(92) Summed values for initial part provisions at Direct Support (QTPO), General Support (QTPI) and Depot (QTPD).

The following outputs are summarized costs for initial provisions and salvage value of residuals.

105. C(93)
106. C(94)
107. C(95)
108. C(96) Summed values for cost of initial LRU provisions at Equipment (CQTE), Direct Support (CQTO), General Support (CQTI) and Depot (CQTD).

109. C(97) Summed values for total initial LRU provisioning cost (CQTT).

110. C(98) Summed values for salvage value of residual LRUs (CRUT).

111. C(99)
112. C(100)
113. C(101)
114. C(102) Summed values for cost of initial Module provisions at Equipment (CQTME), Direct Support (CQTMO), General Support (CQTMI) and Depot (CQTMD).

115. C(103) Summed values for total initial Module provisioning cost (CQTMT).

116. C(104) Summed values for salvage value of residual

Modules (CRMT).

117. C(105)
118. C(106)
119. C(107) Summed values for cost of initial part provisioning at Direct Support (CQTPO), General Support (CQTPI) and Depot (CQTPD).

120. C(108) Summed values for total part provisioning cost (CQTPT).

121. C(109) Summed values for salvage value of residual parts (CRPT).

The following outputs are summarized expected and present value cost totals.

122. C(1)
126. C(77) Summed values for present (PVGCT) and expected (EPVGCT) value cost totals.

123. C(2)
127. C(78) Summed values for cumulative present (PCGT) and expected (SEPV) cost totals.

124. C(6)
128. C(79) Summed values for present (CMPT) and expected (ECMPT) manpower costs.

125. C(20) Summed value for differences between dedicated cost values and expected cost (DELTA).

129. C(81) Summed value for present value cost delta (PDELTA).

SECTION 6 - OUTPUT DEFINITIONS (COST TOTALS)

6.4 Cost Totals Output

Cost total outputs are printed by Subroutine TTLOP for CASE TOTALS and GRAND TOTALS. The same WRITE and FORMAT statements are used for both printouts. The user controls these outputs with the NU input flag. NU=-1 will print the values accumulated from the individual LRU cases as a CASE TOTAL but will not reset the accumulator array. With NU=-2 or -3 a case total is printed and also the accumulated values for all LRU cases evaluated will be printed as a GRAND TOTAL. When NU=-2 the grand total accumulator is not reinitialized but the case total accumulator is. NU=-3 will reinitialize the grand total accumulators and any additional individual LRU case input will start a new grand total. The NU printout selection is entered along with the input data for the last individual LRU case that makes up a case total.

6.4.1 Case Totals - The case total printout is shown on the sample output page of section 8.4.1. This page is printed when the user inputs NU<0 with the last individual LRU case in a case total concept. The numbers assigned to the output variable definitions correspond to the numbers marked on the sample page. The page header information for this output is the same as was discussed for the "INDIVIDUAL LRU OUTPUT" of section 6.1. The parameters enclosed in parenthesis are those evaluated for an individual LRU case. The sample case for this output is shown in section 8.4.1 of this document.

1. COSTIS — Cost output units.
2. CCET — Development and Procurement cost of prime (installed) equipment less the salvage value (CET).
3. CCTS — Development, Procurement, Maintenance, and Software cost of test equipment less salvage value (CTST).
4. CCTSR — Test equipment support cost (CTSR).
5. CCF — Test equipment housing cost (CFT).
6. CCF — Depot space/utilities cost (same as 5 above).
7. CCM — Manpower cost for test and repair personnel including training (CMPT).
8. CCMF — Field maintenance cost.
9. CCMD — Depot maintenance cost.
10. CCMFD — Total maintenance cost.

SECTION 6 - OUTPUT DEFINITIONS (COST TOTALS)

11. CTRF — Field training cost for test and repair personnel.

12. CTRDEP — Depot training cost for test and repair personnel.

13. CTR — Total training cost for test and repair personnel.

14. CIV — Cost of initial and replenishment spares less the salvage value (CIVT).

15. CIVREC — Cost of replenishment spares (CIVR).

16. CRT — Cost of reordering material (CROR).

17. CRT — Cost of reordering material (CROR). Same as 16.

18. CWH — Cost of storage for spares (CWHT).

19. CWH — Cost of storage for spares (CWHT). Same as 18.

20. CSA — Cost of entering and maintaining line items in supply system (CSAT).

21. CSAREC — Cost to maintain (inventory management) supply in the system (CSAR).

22. CSH — Cost of shipping spares (CSHT).

23. CSH — Cost of shipping spares (CSHT). Same as 22.

24. CGT — Case total cost (GCT).

25. CTREC — Total recurring costs.

26. PCD — Total development cost. Includes the cost to develop prime and test equipment plus test equipment software (PVCD).

27. CQTU — Cost of initial LRU provisions (CQTT).

28. PCP — Total acquisition cost. Includes procurement of prime equipment, initial provisions, nonrecurring training costs, and the cost to enter items in inventory (PVCP).

29. CQTM — Cost of initial Module provisions (CQTMT).

30. PCR — Total cost of Operation and Maintenance of

prime equipment, initial provisions, and test equipment (PVCR).

31. CQTP — Cost of initial Parts provisions (CQTPT).

32. PCS — The end of life salvage value (PVCS).

33. CQTUMP — Total initial provisioning cost.

34. PCGT — Present value case total cost (PVGCT).

35. SEMPT — Expected value cost for maintenance manpower (ECMPT).

36. SEPC — Expected value case total cost (EPCGT).

37. SPCR — Present value Operation and Support cost.

38. SEPV — Present value grand total cost (EPVCGT).

39. SDEL — The expected value cost difference (DELTA).

40. SPDEL — The present value cost difference (PDELTA).

6.4.2 Grand Total Output - The grand total printout is shown on the sample output page of section 8.4.2. This page was printed when the user input NU=-2 or -3 with the last individual LRU case in a case total concept. The grand total accumulator is reinitialized only after NU=-3 is input.

The grand totals are accumulated in LOGAM into the array SUM. At the time to print the grand totals the contents of SUM are stored into the CUM array which is equivalenced to the variable names listed in section 6.4.1 above. Therefore, printing this list of names again will give the grand total outputs. Since the variable list was described in section 6.4.1 they will not be included here.

SECTION 6 - OUTPUT DEFINITIONS (MAINTENANCE DEMANDS)

6.5 LOGAM Maintenance Manpower Outputs

Maintenance manpower outputs are printed from Subroutine PERSON when case total outputs are selected. The outputs here are manhours and men for test and repair at the maintenance locations. The outputs are shown on the sample output page section 8.5. The numbers assigned to the output variables correspond to the numbers marked on the sample page.

In addition to the maintenance outputs system and subsystem availabilities are printed.

The following outputs are the case totals that were accumulated from the individual LRU cases in Subroutine PERSON.

1. WPD(1,1) Test manhours/year at all equipment installations.

2. WPD(2,1) Test manhours/year at all Organizational maintenance facilities.

3. WPD(3,1) Test manhours/year at all Direct Support maintenance facilities.

4. WPD(4,1) Test manhours/year at all General Support maintenance facilities.

5. WPD(5,1) Test manhours/year at all Depot maintenance facilities.

6. WPD(1,2) Repair manhours/year at all Equipment installations.

7. WPD(2,2) Repair manhours/year at all Organizational maintenance facilities.

8. WPD(3,2) Repair manhours/year at all Direct Support maintenance facilities.

9. WPD(4,2) Repair manhours/year at all General Support maintenance facilities.

10. WPD(5,2) Repair manhours/year at all Depot maintenance facilities.

11. PERS(1,1) Test manpower at all equipment installations.

12. PERS(2,1) Test manpower at all Equipment maintenance facilities.

13. PERS(3,1) Test manpower at all Direct Support

maintenance facilities.

14. PERS(4,1) Test manpower at all General Support maintenance facilities.

15. PERS(5,1) Test manpower at all Depot maintenance facilities.

16. PERS(1,2) Repair manpower at all equipment installations.

17. PERS(2,2) Repair manpower at all equipment maintenance facilities.

18. PERS(3,2) Repair manpower at all Direct Support maintenance facilities.

19. PERS(4,2) Repair manpower at all General Support maintenance facilities.

20. PERS(5,2) Repair manpower at all Depot maintenance facilities.

21. WPDL(1,1) Test manhours/year per equipment installation.

22. WPDL(2,1) Test manhours/year per Organizational maintenance facility.

23. WPDL(3,1) Test manhours/year per Direct Support maintenance facility.

24. WPDL(4,1) Test manhours/year per General Support maintenance facility.

25. WPDL(5,1) Test manhours/year per Depot maintenance facility.

26. WPDL(1,2) Repair manhours/year per equipment installation.

27. WPDL(2,2) Repair manhours/year per Organizational maintenance facility.

28. WPDL(3,2) Repair manhours/year per Direct Support maintenance facility.

29. WPDL(4,2) Repair manhours/year per General Support maintenance facility.

30. WPDL(5,2) Repair manhours/year per Depot maintenance facility.

31. PERL(1,1) Test manpower per equipment installation.

32. PERL(2,1) Test manpower per Organizational maintenance facility.

33. PERL(3,1) Test manpower per Direct Support maintenance facility.

34. PERL(4,1) Test manpower per General Support maintenance facility.

35. PERL(5,1) Test manpower per Depot maintenance facility.

36. PERL(1,2) Repair manpower per equipment installation.

37. PERL(2,2) Repair manpower per Organizational maintenance facility.

38. PERL(3,2) Repair manpower per Direct Support maintenance facility.

39. PERL(4,2) Repair manpower per General Support maintenance facility.

40. PERL(5,2) Repair manpower per Depot maintenance facility.

The following two outputs are evaluated in LOGAM. The number (NA) of values printed per array name is controled by the number of system availability modes input by the user. The first value of each array is the system availability and the subsequent values are subsystem availabilities. The control for system availability computations is determined by the TAYZ input array. Refer to section 3.2 for a detailed description of the TAYZ inputs.

41. CAYZ Operational availabilities.

42. CAYZI Inherent availabilities.

SECTION 6 - OUTPUT DEFINITIONS (SYSTEM SUPPORT COSTS)

6.6 System Support Costs Output

There are two levels of system support costs that can be output. LOGAM System Maintenance costs that are accumulated in Subroutine LGCST for case totals are printed by Subroutine EIGHT using the DA PAM 11-4 format. These outputs are printed when NU<0 is input and the accumulators are reset using the same logic described in section 6.4 Both case and grand totals can be output for LOGAM system support costs but for the second level of support costs only case totals are printed. The second level is the addition of costs for a TOE to the LOGAM system maintenance costs. TOE costs are evaluated by Subroutine OPER and included with the LOGAM costs for System Operation and Support costs output.

6.6.1 LOGAM System Maintenance Cost Case Totals - These costs were accumulated in Subroutine LGCST and will be printed by Subroutine EIGHT. Subroutine EIGHT is called from the main program when NU<0 is input. The numbers assigned to the output descriptions correspond to the number marked on sample output page of section 8.6.1. Only those outputs that are evaluated in LOGAM will be listed in this section. The variables enclosed in parentheses are the names used in LOGAM to compute the cost at the individual LRU case level.

1. WD(1) Development Engineering costs (CED+CTSD).
2. PER(1) Percent of R&D cost for WD(1)
3. WD(1) Total research and Engineering cost.
4. PER(1) Percent of R&D cost for ED(1) (100%).
5. WD(2) Investment costs that include recurring costs (CEP + CEV + CTSV) and nonrecurring cost (CPE).
6. PER(2) Percent of Total Investment cost for WD(1) (100%).
7. WD(3) Cost of Data (CTSOFT).
8. PER(3) Percent of Total Investment cost for WD(3).
9. WD(4) Cost of Training Services and Equipment (CMPPY).
10. PER(4) Percent of Total Investment cost for WD(4).
11. WD(5) Cost of Initial Spares and Repair Parts (CIVP).
12. PER(5) Percent of Total Investment cost for WD(5).

SECTION 6 - OUTPUT DEFINITIONS (SYSTEM SUPPORT COSTS)

13. WD(6) Other Investment costs (CTSD). This is the procurement costs (CTSP) for test equipment.

14. PER(6) Percent of Total Investment cost for WD(6).

15. XB Total Investment cost.

16. PER(1) Percent of Total Investment cost for XB (100%).

19. SDA(1) Maintenance pay and allowances (PA).

20. PER(7) Percent of O&S cost for SDA(1).

25. SDA(2) Cost of Replenishment spares (REPSF).

26. PER(10) Percent of O&S cost for SDA(2).

31. SDA(3) Cost of Labor (DML).

32. PER(13) Percent of O&S cost for SDA(3).

33. SDA(4) Cost of materials (RFPSD).

34. PER(14) Percent of O&S cost for SDA(4).

35. SDA(5) Cost of Transportation (CSHTD).

36. PER(15) Percent of O&S cost for SDA(5).

37. SDA(6) Cost of modification materials (DMM).

38. PER(16) Percent of O&S cost for SDA(6).

39. SDA(7) Cost of maintenance civilian labor (CLS).

40. PER(17) Percent of O&S Cost of SDA(7).

41. POUT(6) Other Direct O&S costs (U17). Same as SDA(10).

42. PER(18) Percent of O&S cost for POUT(6).

43. POUT(7) Personnel Replacement cost (U12). Same as SDA(8).

44. PER(19) Percent of O&S cost for POUT(7).

51. POUT(11) Other Indirect cost (CSHTF + CFT). Same as SDA(9).

52. PER(23) Percent of O&S Cost for POUT(11).

53. X Total Operating and Support Cost.

54. PER(1) Percent of O&S cost for X (100%).

55. GT Grand Total cost.

6.6.2 System Operations and Support Cost Case Totals - These costs were evaluated by Subroutine OPER. To get System Operation and Support costs TOE data must be input with the NAMELIST/L/ inputs. Also IOPER=1 must be input or Subroutine OPER will not be called. Under these conditions and NU<-1 these costs will be evaluated and printed. Subroutine EIGHT is called from LOGAM to print the results. The System Operation Support costs includes the LOGAM costs described in section 6.6.1 and the TOE costs. Only case totals are output for System Operation and Support whereas both case and grand totals are printed for System Maintenance Support costs. Only those outputs that are evaluated in the post processor will be listed here. All other outputs are the LOGAM case totals described in section 6.6.1. The numbers assigned to the output list correspond to the numbers marked on the sample output page of section 8.6.2.

17. POUT(1) Crew pay and allowances.

18. PER(6) Percent of O&S cost for POUT(1).

21. POUT(2) Indirect pay and allowance.

22. PER(8) Percent of O&S cost for POUT(2).

23. POUT(3) Permanent change of Station costs.

24. PER(9) Percent of O&S for POUT(3).

27. POUT(4) Cost of petroleum, oil and lubricants.

28. PER(1) Percent of O&S for POUT(4).

29. POUT(5) Cost of Unit Training Ammunition and Missiles.

30. PER(12) Percent of O&S for POUT(5).

41. POUT(6) Other Direct Costs (includes SDA(10) from LOGAM).

42. PER(18) Percent of O&S for POUT(6).

43. POUT(7) Cost of Personnel Replacement (includes SDA(8) from LOGAM).

44. PER(19) Percent of O&S cost for POUT(7).

SECTION 6 - OUTPUT DEFINITIONS (SYSTEM SUPPORT COSTS)

45. POUT(8) Cost of Transients, patients and prisioners.

46. PER(20) Percent of O&S cost for POUT(8).

47. POUT(9) Cost of quarters, maintenance and utilities.

48. PER(21) Percent of O&S cost for POUT(9).

49. POUT(10)Cost of Medical Support.

50. PER(22) Percent of O&S cost for POUT(10).

51. POUT(11)Cost of other Indirect O&S costs (includes SDA(9) from LOGAM).

52. PER(23) Percent of O&S cost for POUT(11).

6.6.3 LOGAM System Maintenance Grand Total Costs = These costs are accumulated in the main routine (LOGAM) by adding the contents of the case total accumulators. When the user selects a case total output with NU<-1, the grand total accumulation is made. When the user selects grand total outputs with NU=-3, Subroutine EIGHT is called to print the grand total accumulations for the LOGAM System Maintenance costs. Grand totals are not accumulated for System Operations and Support Cost which include the costs from a TOE.

The description for the outputs on sample output page 8.6.3 are the same as was described in section 6.6.1.

SECTION 6 - OUTPUT DEFINITIONS (SENSITIVITY)

6.7 Sensitivity Outputs

The sensitivity output uses the same program formats as described earlier in section 6.1. An additional line of print is included with the Individual LRU output pages that describe the sensitivity analysis being performed. The Individual LRU outputs can be inhibited by inputting INHIB=1. In this instance only the description of the sensitivity will be output per individual LRU case. Section 6.7.1 describes the sensitivity case where INHIB=1. Section 6.7.2 and section 6.7.3 describes the Individual LRU case output. Case totals, grand totals, and all other output forms are printed as described earlier for the baseline case.

6.7.1 Sensitivity with Inhibited Print - The additional print given when a sensitivity analysis is being performed is shown on the sample output pages of sections 8.7.1, 8.7.2 and 8.7.3. The numbers assigned to the following descriptions correspond to the numbers marked on the sample outputs.

1. UNITIS — This output normally describes the class and class number of an individual LRU case. It can be used when inputting SENSY data to describe the sensitivity analysis. Line item 9 of section 5.1.1 describes the UNITIS input.

2. REMARK — A 72 column input field that describes an individual LRU case; in this instance it is used to describe a sensitivity analysis. Line item 10 of section 5.1.1 describes this input.

3. KPASS — The sensitivity pass number. An LRU input value can be varied one or more times using the SENSY input array. KPASS will count each variation to the input value. In the example of section 8.7.1 KPASS is equal to 1. This is the second pass through the LRU cases where the value for the third input variable (AYZP) is modified. For this pass the value of AYZP is 1.75. All LRU cases input will be executed with the value of AYZP=1.75.

4. NRU — The individual LRU case number. The case number is the order in which the LRU case was read from the NAMELIST/L/ data deck.

5. NVAR — The modified variables position within the list of variables in COMMON/INPUT/. In the sample output of section 8.7.1 the variable modified is AYZP which is in position 3 of COMMON/INPUT/. Refer to input section 5.1.2 for a list of the variable that can be modified

by SENSY and their positions.

6. VALUE — The modified value for the variable stored in position NVAR of COMMON/INPUT/. The sample output of section 8.7.1 shows only one variable per case being modified. There can be several different variable modifications per case if sc desired.

6.7.2 Initial Provisioning Sensitivity - The outputs when a sensitivity analysis is being performed and INHIB=0 are the same as discussed in sections 6.1 through 6.6 except that the sensitivity input descriptions are included on the individual LRU output page. Section 8.7.2 is a sample provisioning output page for an individual LRU with the sensitivity data. The numbers assigned to the following statements correspond to the numbers marked on the output page of section 8.7.2.

7. UNITIS — This is the class and class number description that was input with the baseline case.

8. REMARK — This is the 72 column description of on LRU that was input with the baseline case.

9. KPASS — This variable and the ones following are printed in Subroutine SENSIT. KPASS is the same as was discussed in section 6.7.1.

10. NRU — Same as in section 6.7.1.

11. NVAR — Same in section 6.7.1.

12. VALUE — Same as in section 6.7.1.

51. OR — Operational availability Over-Ride factor. This value is the fractional portion of the input for VALUE of item 12 above. AYZP, the third input value, was selected as the variable to be modified for a sensitivity run using an Over-Ride factor. The OR factor is printed in Subroutine IPVOP.

The additional outputs on the sample output page of section 8.7.2 are discussed in section 6.1 for Individual LRU Provisioning Outputs. A comparison between the baseline case and the sensitivity case can be made from sample output pages 8.7.2.

6.7.3 Individual LRU Case Sensitivity Outputs

The outputs for this section are the same as discussed in section

6.7.1 and section 6.2. Section 6.7.1 discusses those parameters pertaining to the sensitivity run. Section 6.2 discusses the LOGAM output variables. For a comparison of a baseline case and a sensitivity case refer to sample outputs of section 8.2 and 8.7.3.

SECTION 6 - OUTPUT DEFINITIONS ($L INPUTS)

6.8 Output of Logam Inputs ($L)

Subroutine INPOP is called from the main program (LOGAM) to print the values input through NAMELIST/L/. The Namelist variable names are stored in a DATA statement in Subroutine FINLE and are in the exact order as the list of COMMON/INPUT/. The input variable list is in alphabetical order with the exception of SENSY and T(TOE data array) which are the last 2 items in the list. These 2 items are the only variables of COMMON/INPUT/ whose values are not output when INPOP is called. The remaining variables are output in alphabetical order.

To select the output of input values the user sets the IO input flag to a positive non-zero value. As a matter of convention IO=1 should be used. IO is one of the NAMELIST/L/ input variables. Which input case the user sets IO depends on when the user wants to review his inputs. Ordinarily the flag would be set in the last case to get a listing that is grouped for all LRU cases input.

Each input case is written to a disc file (ND) in the main program. When an output request (IO=1) is made, file ND is read (from INPOP) once for each LRU case input and a line of output for each of the cases is printed. ND is rewound to read the input cases again and print the next group of input values. This procedure is repeated until the values for all variable names are printed.

A sample output for the following output descriptions can be found in section 8.8. This sample case lists only a few of the many input variables but is sufficient to illustrate the format for the outputs. The numbers for the output descriptions below correspond to numbers annotated on the sample output page.

1. I — File ND is read in a "DO LOOP" ranging from 1 to LRU, where LRU is the number of cases that have been input. I is then the index for the case being read from ND.

2. UNITIS — Each LRU case input is preceded by 2 description cards. UNITIS is the first of these 2 cards and is a 20 character description for the class of LRU input.

3. ARA — This is the first variable name listed in COMMON/INPUT/. When INPOP reads ND the values are read into the SAVV array. Since SAVV(1) is equivalenced to ARA, all 326 values of /INPUT/ will be restored each time ND is read.

4. CCSP — This is the last variable to be printed on the current print line. After printing CCSP the

input file (ND) is read again to get the inputs for the next LRU case.

5. I — This is the index for the last LRU case on ND. Once the line of outputs for this case are printed ND is rewound and the first LRU case will be read again. "I" will now become 1.

6. CCSPP — This variable begins a new line of output variables. The procedure described in items 1 through 5 above are repeated for each new line of output variables until all variables (through ZU) of COMMON/INPUT/ are printed.

SECTION 7

SAMPLE INPUTS

SECTION 7 - SAMPLE INPUTS

7.1 LOGAM Sample Input Case

```
            MISSILE SYSTEM
                    CASE 1(SENSY)
                              FY83 COSTS

SAMPLE CASE
OCT 83
THOUSANDS OF DOLLARS                          .001
   TOTAL                                                    1          10
LRU 1
USAREUR
 $L ARA=.4,ARAD=.2,AYZP=1,CAD=348,CALMAN=25526,
 FNSP=.49,FNGF=.2,
 CALSET=2,CDDI=.3,CDEO=.05,CDFD=.21,CDID=.3,CDIO=.1,
 CDIST=.05,CDOE=.05,CDOI=.1,CDMAN=25526,CDPMAN=61409,
 CDPRMN=61409,CDRMAN=25526,CERMAN=25526,CEMAN=25526,
 CFTD=1,CGMAN=25526,CGRMAN=25526,
 CKMD=.85,CKME=.85,CKMI=.85,CKMO=.85,CKPD=.85,CKPI=.85,
 CKPO=.85,CKUD=.85,CKUE=.85,CKUI=.85,CKUO=.85,
 CONMAN=25526,CRM=1254,CRP=1254,CRU=1254,CSDEP=1,
 CSDSU=.25,CSESU=.25,CSGSU=.25,CTRA=11332,CTRAD=2471,
 CUCE=43736,DADQL=.95,DTE=10,DTI=30,DTO=30,
 EVDM=1,EVEM=1,EVDR=1,EVDT=1,
 EVER=1,EVET=1,EVIM=1,EVIR=1,EVIT=1,EVOM=1,EVOR=1,
 EVOT=1,FINT=0,FTM=38,FTP=20,FTU=64,HPM=30,
 HPP=30,HPU=30,JTED=2,OL=5,15,15,30,OST=6,6,10,90,
 OTF=.0342,PMR=0,PPR=0,PUR=0,QMM=1,QMP=1,QMU=1,
 QTD=0,QTE=0,QTO=0,QTD=0,QTME=0,QTMO=0,QTMI=0,
 QTMD=0,QTPO=0,QTPI=0,QTPD=0,
 RDD=127,RID=15,REO=5,ROI=15,SL=5,15,15,30,SPE=0,
 SPEV=1,SPEVR=1,STAT=60,TAT=10,30,30,127,TATE=7,
 TDMAN=2,TDRMAN=2,TEO=7,TEMAN=2,TERMAN=2,TEVMAN=2,
 TGMAN=2,TGRMAN=2,TID=15,TIO=15,TOE=7,TOI=15,TOI=15,
 TUMD=168,TUMI=168,TUMO=168,WD=40,WDM=40,WDR=40,
 WE=44,WER=44,WEM=44,WI=44,WIM=44,WIR=44,
        WO=44,WOM=44,WOR=44,YAT=.001,YO=5,YP=7,
 ETE=0,ETEI=0,EREI=0,WMR=44,WMT=44,RF=0,
 YR=20,YZ=0,ZFL=1,ZI=0,
 EACSP=1,EACAL=1,
 ED=10,OD=10,DI=6,DD=1,EDS=10,ODS=10,DIS=6,DDS=1,OTF=.0342,EE=4,
 TRC=1.0,TC=.1,TE=2.0,TER=2.5,TF=2.0,TFR=2.5,TI=2.0,TIR=2.5,
 TD=8,TDR=8,TMD=4,TMDR=12,
 CUP=104911,CMP=3782,CPP=39,CUBEU=1,CUBEM=.07,CUBEP=.005,WU=40,WM=2,WP=.1,
 P=12,PP=50,CKIT=5773,WTKIT=10,ZO=.5,REPEAT=1,
 YMWO=.05,TAYZ=1,9*0,E=.00259,
 G(11)=.25,G(14)=.5,G(19)=.15,G(20)=.1,4*1,1,1,1,1,
 CAD=985,CEN=1616,
 FUD=.95,FUE=.98,FUO=.98,FUI=.98,FMO=.95,FMI=.98,FMD=.95,
 SUE=.01,SUO=.01,SUI=.01,SUD=.05,SME=.01,SMO=.01,SMI=.01,SMD=.08,
 CRI=17058,CI=68230,CLRUPG=25587,CMODPG=21322,CPI=42644,
 CTRI=15352,ETI=1,FTI=0,CPII=37882,CRII=23677,CII=65294,
 CPUBII=13259,CTRII=23677,TDPMII=1.4,TDPRII=1.4,ETII=1,
 FTII=200,CALPUB=11940,CCAL=34115,CCALP=25587,CCALR=17057,CTRCAL=11940,
 EACAL=1,CCSP=63966,CCSPP=32409,CCSPR=16205,CTCPJB=14499,CTRSPT=14499,
 CONTCT=2,EACSP=1,TDNMAN=2,FI=.02,FII=.02,TALMAN=2,
 IFLAG=0,IOPER=1,
 $
LRU 2
USAREUR
 $L E=.000447,
 CUP=2142,CMP=1089,CPP=2,CUBEU=.5,CUBEM=.05,CUBEP=.005,WU=.75,WM=.1,WP=.05,
 P=2,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,
 TRC=.75,TC=.3,TE=1.25,TER=2.75,TF=1.25,TFR=2.75,TI=1.25,TIR=2.75,
 TD=4,TDR=8,TMD=2,TMDR=5,SMF=.0046,
```

7.1 LOGAM Sample Input Case (Continued)

```
  I=2946,CI=11769,CLRUPG=4413,CMODPG=3878,CPI=7355,
 ,TRI=2648,ETI=1,CPII=6538,CRII=4086,CII=11442,CPUBII=2288,
 CTRII=4086,ETII=1,CALPUB=2059,CCAL=5884,CCALP=4413,CCALR=2943,CCSPR=2795,
 CTRCAL=2059,EACAL=1,CCSP=5590,CCSPP=5590,CTCPUB=2501,CTRSPT=2501,EACSP=1,
 $
LRU 3
USAREUR
 $L
 CUP=2142,CMP=1089,CPP=2,CUBEU=.5,CUBEM=.05,CUBEP=.005,WU=.75,WM=.1,WP=.05,
 P=2,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,
 TRC=2.75,TC=.3,TE=1,TER=3,TF=1,TFR=3,TI=1,TIR=3,
 TD=4,TDR=8,TMD=2,TMDR=5,SMF=.0046,E=.0018,FNGF=.11,
 TMO=2,TMOR=4,TMI=2,TMIR=4,
 G(9)=.5,G(15)=.25,G(20)=.25,H=1,1,1,1,
 CRI=44784,CI=179137,CLRUPG=67176,CMODPG=55980,CPI=111960,CTRI=40306,
 EACAL=1,CCSP=63966,CCSPP=32409,CCSPR=16205,CTCPUB=14499,CTRSPT=14499,
 ETII=1,CALPUB=31349,CCAL=89568,CCALP=67176,CCALR=44784,CTRCAL=31349,
 EACAL=1,CCSP=167941,CCSPP=85090,CCSPR=42545,CTCPUB=38066,CTRSPT=38066,EACSP=1,
 CII=174052,CPUBII=34819,
 $
LRU 4
USAREUR
 $L E=.000747,
 CUP=35012,CMP=2239,CPP=34,CUBEU=.7,CUBEM=.08,CUBEP=.005,WU=26,WM=1.5,WP=.8,
 P=9,PP=40,CKIT=1716,WTKIT=10,ZO=.5,REPEAT=1,
 TRC=.8 ,TC=.1,TE=1.8,TER=2.5,TF=1.8,TFR=2.5,TI=1.8,TIR=2.5,
 TD=8,TDR=6,TMD=5,TMDR=13,
 SMF=.0046,TMO=3,TMOR=5,TMI=3,TMIR=3,
 G(1)=.1,G(9)=.3,G(10)=.2,G(20)=.4,H=1,1,1,1,
 FNGF=.39,
 CRI=4920,CI=19680,CLRUPG=7380,CMODPG=6150,CPI=12299,CTRI=4428,
 ETI=1,CPII=0,CRII=0,CII=0,CPUBII=0,CTRII=0,ETII=0,
 CALPUB=3444,CCAL=9839,CCALP=7380,CCALR=4921,CTRCAL=3444,
 EACAL=1,CCSP=18450,CCSPP=9347,CCSPR=4674,CTRSPT=4182,EACSP=1,
 CV=100000,CPV=75000,CRV=10000,CPUBV=10000,CTRV=15000,FE=.02,
 ETE=1,ETEI=1,EREI=1,WMR=44,WMT=44,RF=.05,
 CTCPUB=4182,NU=-1,
 $
LRU 5
USAREUR
 $L E=.000097,
 CUP=4500,CMP=2277,CPP=5.1,CUBEU=.2,CUBEM=.02,CUBEP=.005,WU=4.5,WM=.2,WP=.1,
 P=1,PP=30,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,EE=4,
 TRC=.5 ,TC=.1,TE=.25,TER=.75,TF=.25,TFR=.75,TI=.25,TIR=.75,
 TD=4,TDR=8,TMD=2,TMDR=5,
 TMO=0,TMOR=0,
 SMF=.0046,
 G(4)=.5,G(8)=.5,H=0,1,0,1,
 EACAL=1,NU=1,
 FNGF=.42,
 CRI=640,CI=2555,CLRUPG=958,CMODPG=798,CPI=1597,CTRI=575,ETI=1,
 CPII=1418,CRII=886,CII=2482,CPUBII=496,CTRII=886,ETII=1,
 CALPUB=447,CCAL=1296,CCALP=958,CCALR=639,CTRCAL=447,EACAL=1,
 CCSP=2395,CCSPP=1213,CCSPR=607,CTCPUB=543,CTRSPT=543,EACSP=1,
 CV=0,CPV=0,CRV=0,CPUBV=0,CTRV=0,FE=0,
 ETE=0,ETEI=0,EREI=0,WMR=44,WMT=44,RF=0,
 $
LRU 6
USAREUR
 $L E=.000763,
 CUP=91623,CMP=7298,CPP=48,CUBEU=1,CUBEM=.3,CUBEP=.02,WU=36,WM=2,WP=.75,
 P=8,PP=40,CKIT=1883,WTKIT=10,ZO=.5,REPEAT=1,EE=4,
 TRC=1.2,TC=.1,TE=0,TER=0,TF=.50,TFR=1.5,TI=.50,TIR=1.5,
 TMO=1.5,TMOR=4,TMI=1.5,TMIR=4,
 TD=8,TDR=6,TMD=6,TMDR=16,
```

7.1 LOGAM Sample Input Case (Continued)

```
 (12)=.5,G(13)=.5,H=0,1,1,0,
 FNGF=.49,
 CRI=5026,CI=20100,CLRUPG=7538,CMODPG=6282,CPI=12563,CTRI=4523,ETI=1,
 CPII=0,CRII=0,CII=0,CPUBII=0,CTRII=0,ETII=1,
 CALPUB=3518,CCAL=10050,CCALP=7538,CCALR=5025,CTRCAL=3518,EACAL=1,
 CCSP=18845,CCSPP=9548,CCSPR=4774,CTCPUB=4271,CTRSPT=4271,EACSP=1,
 $
LRU 7
USAREUR
 $L E=.00005,
 CUP=37455,CMP=7489,CPP=18,CUBEU=2,CUBEM=.5,CUBEP=0,WU=5,WM=.5,WP=.01,
 P=1,PP=40,CKIT=500,WTKIT=10,ZO=.5,REPEAT=1,
 TRC=1.5,TC=.1,TE=1.0,TER=1.8,TF=0,TFR=0,TI=1.0,TIR=1.8,
 TD=4,TDR=6,TMD=4,TMDR=6,
 TMO=0,TMOR=0,TMI=.8,TMIR=2.5,
 SMF=.0046,
 G(11)=.4,G(15)=.4,G(20)=.2,H=1,1,1,1,
 FNGF=.33,
 CRI=331,CI=1319,CLRUPG=494,CMODPG=412,CPI=824,CTRI=297,ETI=1,
 CPII=732,CRII=457,CII=1280,CPUBII=257,CTRII=457,ETII=1,
 CALPUB=231,CCAL=659,CCALP=494,CCALR=330,CTRCAL=231,EACAL=1,
 CCSP=1236,CCSPP=626,CCSPR=313,CTCPUB=280,CTRSPT=280,EACSP=1,
 $
LRU 8
USAREUR
 $L E=.000193,
 CUP=6627,CMP=3354,CPP=21,CUBEU=.5,CUBEM=.05,CUBEP=.01,WU=3,WM=.75,WP=.05,
 P=44,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,EE=4,
 TRC=.50,TC=.1,TE=.25,TER=.75,TF=.25,TFR=.75,TI=.25,TIR=.75,
 TD=4,TDR=8,TMD=2,TMDR=5,
 TMO=0,TMOR=0,
 G(20)=1,H=0,0,0,1,
 FNGF=.2,
 CRI=1271,CI=5085,CLRUPG=1907,CMODPG=1589,CPI=3178,CTRI=1144,ETI=1,
 CPII=2822,CRII=1764,CII=4938,CPUBII=988,CTRII=1764,ETII=1,
 CALPUB=890,CCAL=2542,CCALP=1907,CCALR=1271,CTRCAL=890,EACAL=1,
 CCSP=4768,CCSPP=2415,CCSPR=1208,CTCPUB=1081,CTRSPT=1081,EACSP=1,
 $
LRU 9
USAREUR
 $L E=.001097,
 TRC=2.4,TC=.1,TE=1,TER=5,TF=1,TFR=5,TI=1.5,TIR=5.1,TMI=2,TMIR=6,
 TD=8,TDR=20,TMD=8,TMDR=20, TMO=2,TMOR=6,
 CUP=139755,CMP=4641,CPP=17,CUBEU=15,CUBEM=.5,CUBEP=.05,WU=150,WM=15,WP=.5,
 P=13,PP=40,CKIT=7226,WTKIT=30,REPEAT=1,
 G(9)=.5,G(18)=.38,G(20)=.12,H=1,1,1,1,
 FNGF=.2,
 CRI=7226,CI=28900,CLRUPG=10838,CMODPG=9031,CPI=18062,CTRI=6502,ETI=1,
 CPII=16046,CRII=28080,CPUBII=5616,CTRII=10029,ETII=1,
 CALPUB=5057,CCAL=14449,CCALP=10838,CCALR=7225,CTRCAL=5057,EACAL=1,
 CCSP=27094,CCSPP=13727,CCSPR=6863,CTCPUB=6141,CTRSPT=6141,EACSP=1,
 $
LRU 10
USAREUR
 $L
 E=.0024,
 TRC=2.4,TC=.1,TE=0,TER=0,TF=0,TFR=0,TI=1.5,TIR=5.1,TMI=1.5,TMIR=4.5,
 TD=8,TDR=20,TMD=8,TMDR=20,
 CUP=139755,CMP=4641,CPP=17,CUBEU=15,CUBEM=.5,CUBEP=.05,WU=150,WM=15,WP=.5,
 P=13,PP=40,CKIT=7226,WTKIT=30,ZO=.7,REPEAT=1,
 SMF=.0046,
 FNGF=.2,
 G(14)=.25,G(19)=.25,G(20)=.5,H=1,1,1,1,
 CRI=15801,CI=63225,CLRUPG=23709,CMODPG=19758,CPI=39516,CTRI=14225,ETI=1,
 CPII=35104,CRII=21940,CII=61432,CPUBII=12286,CTRII=21940,
```

7.1 LOGAM Sample Input Case (Continued)

```
 CAL=31612,CCALP=23709,CCSP=59274,CII=28080,CTCPUB=13435,
CTRCAL=11064,EACAL=1,EACSP=1,CALPUB=11064,
 T=1.,1.,0.,5.,1.,0.,0.,0.,0.,29804.,
 1.,10.,0.,4.,1.,0.,0.,0.,0.,29804.,
 1.,13.,0.,3.,1.,0.,0.,0.,0.,18718.,
 1.,16.,0.,3.,0.,0.,1.,0.,0.,14038.5,
 1.,8.,0.,2.,1.,0.,0.,0.,0.,21057.75,
 1.,1.,0.,2.,0.,0.,0.,0.,1.,18718.,
 1.,16.,0.,2.,0.,0.,1.,0.,0.,18718.,
 1.,16.,0.,.5,0.,0.,0.,0.,1.,18080.,
 1.,3.,0.,.5,1.,0.,0.,0.,0.,18080.,
 1.,1.,1.,9.,1.,0.,0.,0.,0.,16759.,
 1.,4.,1.,8.,1.,0.,0.,0.,0.,16765.75,
 1.,4.,1.,8.,0.,0.,1.,0.,0.,16765.75,
 1.,3.,1.,8.,0.,0.,0.,0.,1.,16759.,
 1.,6.,1.,7.,1.,0.,0.,0.,0.,16759.,
 1.,20.,1.,7.,0.,0.,1.,0.,0.,16759.,
 1.,15.,1.,7.,0.,0.,0.,1.,0.,16759.,
 1.,16.,1.,7.,0.,0.,0.,0.,1.,16759.,
 1.,8.,1.,6.,1.,0.,0.,0.,0.,10469.,
 1.,32.,1.,6.,0.,0.,1.,0.,0.,10469.,
 1.,22.,1.,6.,0.,0.,0.,1.,0.,10469.,
 1.,6.,1.,6.,0.,0.,0.,0.,1.,10469.,
 1.,2.,1.,5.,0.,1.,0.,0.,0.,10469.,
 1.,148.,1.,5.,0.,0.,1.,0.,0.,10469.,
 1.,77.,1.,5.,0.,0.,0.,1.,0.,10469.,
 1.,216.,1.,4.,0.,0.,1.,0.,0.,10469.,
 1.,135.,1.,4.,0.,0.,0.,1.,0.,10469.,
 1.,224.,1.,3.,0.,0.,1.,0.,0.,7500.,
 1.,127.,1.,3.,0.,0.,0.,1.,0.,7500.,
 2,1,.48,1769,247732,.295,8283,.295,7388.7,80052,
 2,2,.112,17068,.112,17068,.036,.036,687,741,
 2,3,.46,12011,0,0,0,0,0,0,
 3,0,0,600,5,1.32,.0001,0,0,200,
 3,0,0,400,4,1.32,.0001,0,0,62,
 3,0,0,400,4,1.12,.0001,0,0,53,
 3,0,0,324,138,1.50,.001,0,0,4,
 4,16000,303360,5000,50000,0,0,0,0,0,
 6,1,1000,50000,5000,10000,10000,0,0,3,
 6,2,10000,50000,10000,15000,10000,4000,0,3,
 7.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
 8.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
 $
LRU 1
CONUS
 $L
 ED=6,EDS=6,OD=2,ODS=2,DI=2,DIS=2,DTF=.025,EE=4,
 TRC=1.0,TC=.1,TE=2.0,TER=2.5,TF=2.0,TFR=2.5,TI=2.0,TIR=2.5,
 TD=8,TDR=8,TMD=4,TMDR=12,
 CUP=104911,CMP=3782,CPP=39,CUBEU=1,CUBEM=.07,CUBEP=.005,WU=40,WM=2,WP=.1,
 P=12,PP=50,CKIT=5773,WTKIT=10,ZD=.5,REPEAT=1,
 YMWD=.05,TAYZ=1,9*0,E=.00259,
 G(11)=.25,G(14)=.5,G(19)=.15,G(20)=.1,H=1,1,1,1,
 CAD=591,CEN=969,
 CDDI=.21,CDID=.21,STAT=20,
 FUD=.95,FUE=.98,FUO=.98,FUI=.98,FMO=.95,FMI=.98,FMD=.95,
 SUE=.01,SUO=.01,SUI=.01,SUD=.05,SME=.01,SMD=.01,SMI=.01,SMO=.08,
 CRI=17058,CI=68230,CLRUPG=25587,CMODPG=21322,CPI=42644,
 CTRI=15352,ETI=1,FTI=0,CPII=37882,CRII=23677,CII=66294,
 CPUBII=13259,CTRII=23677,TDPMII=1.4,TDPRII=1.4,ETII=1,
 FTII=200,CALPUB=11940,CCAL=34115,CCALP=25587,CCALR=17057,CTRCAL=11940,
 EACAL=1,CCSP=63966,CCSPP=32409,CCSPR=16205,CTCPUB=14499,CTRSPT=14499,
 CONTCT=2,EACSP=1,TONMAN=2,FI=.02,FII=.02,TALMAN=2,
 $
LRU 2
```

7.1 LOGAM Sample Input Case (Continued)

```
 $L E=.000447,
 CUP=2142,CMP=1089,CPP=2,CUBEU=.5,CUBEM=.05,CUBEP=.005,WU=.75,WM=.1,WP=.05,
 P=2,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,
 TRC=.75,TC=.3,TE=1.25,TER=1.75,TF=1.25,TFR=1.75,TI=1.25,TIR=1.75,
 TD=4,TDR=8,TMD=2,TMDR=5,SMF=.0046,
 TMO=2,TMOR=4,TMI=2,TMIR=4,
 G(3)=.25,G(9)=.25,G(10)=.25,G(11)=.25,H=1,1,1,1,
 EE=4,EACAL=1,FNGF=.11,
 CRI=2946,CI=11769,CLRUPG=4413,CMODPG=3878,CPI=7355,
 CTRI=2648,ETI=1,CPII=6538,CRII=4086,CII=11442,CPUBII=2288,
 CTRII=4086,ETII=1,CALPUB=2059,CCAL=5884,CCALP=4413,CCALR=2943,CCSPR=2795,
 CTRCAL=2059,EACAL=1,CCSP=5590,CCSPP=5590,CTCPUB=2501,CTRSPT=2501,EACSP=1,
 $
LRU 3
CONUS
 $L
 CUP=2142,CMP=1089,CPP=2,CUBEU=.5,CUBEM=.05,CUBEP=.005,WU=.75,WM=.1,WP=.05,
 P=2,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,
 TRC=2.75,TC=.3,TE=1,TER=3,TF=1,TFR=3,TI=1,TIR=3,
 TD=4,TDR=8,TMD=2,TMDR=5,SMF=.0046,E=.0018,FNGF=.11,
 TMO=2,TMOR=4,TMI=2,TMIR=4,
 G(9)=.5,G(15)=.25,G(20)=.25,H=1,1,1,1,
 CRI=44784,CI=179137,CLRUPG=67176,CMODPG=55980,CPI=111960,CTRI=40306,
 EACAL=1,CCSP=63966,CCSPP=32409,CCSPR=16205,CTCPUB=14499,CTRSPT=14499,
 ETII=1,CALPUB=31349,CCAL=89568,CCALP=67176,CCALR=44784,CTRCAL=31349,
 EACAL=1,CCSP=167941,CCSPP=85090,CCSPR=42545,CTCPUB=38066,CTRSPT=38066,EACSP=1,
 CII=174052,CPUBII=34819,
 $
LRU 4
CONUS
 $L E=.000747,
 CUP=35012,CMP=2239,CPP=34,CUBEU=.7,CUBEM=.08,CUBEP=.005,WU=26,WM=1.5,WP=.8,
 P=9,PP=40,CKIT=1716,WTKIT=10,ZO=.5,REPEAT=1,
 TRC=.8 ,TC=.1,TE=1.8,TER=2.5,TF=1.8,TFR=2.5,TI=1.8,TIR=2.5,
 TD=8,TDR=6,TMD=5,TMDR=13,
 SMF=.0046,TMO=3,TMOR=5,TMI=3,TMIR=5,
 G(1)=.1,G(9)=.3,G(10)=.2,G(20)=.4,H=1,1,1,1,
 FNGF=.39,
 CRI=4920,CI=19680,CLRUPG=7380,CMODPG=6150,CPI=12299,CTRI=4428,
 ETI=1,CPII=0,CRII=0,CII=0,CPUBII=0,CTRII=0,ETII=0,
 CALPUB=3444,CCAL=9839,CCALP=7380,CCALR=4921,CTRCAL=3444,
 EACAL=1,CCSP=18450,CCSPP=9347,CCSPR=4674,CTRSPT=4182,EACSP=1,
 CV=100000,CPV=75000,CRV=10000,CPUBV=10000,CTRV=15000,FE=.02,
 ETE=1,ETEI=1,EREI=1,WMR=44,WMT=44,RF=.05,
 CTCPUB=4182,NU=-1,
 $
LRU 5
CONUS
 $L E=.000097,
 CUP=4500,CMP=2277,CPP=5.1,CUBEU=.2,CUBEM=.02,CUBEP=.005,WU=4.5,WM=.2,WP=.1,
 P=1,PP=30,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,EE=4,
 TRC=.5 ,TC=.1,TE=.25,TER=.75,TF=.25,TFR=.75,TI=.25,TIR=.75,
 TD=4,TDR=8,TMD=2,TMDR=5,
 TMO=0,TMOR=0,
 SMF=.0046,
 G(4)=.5,G(8)=.5,H=0,1,0,1,
 EACAL=1,NU=1,
 FNGF=.42,
 CRI=640,CI=2555,CLRUPG=958,CMODPG=798,CPI=1597,CTRI=575,ETI=1,
 CPII=1418,CRII=886,CII=2482,CPUBII=496,CTRII=886,ETII=1,
 CALPUB=447,CCAL=1296,CCALP=958,CCALR=638,CTRCAL=447,EACAL=1,
 CCSP=2395,CCSPP=1213,CCSPR=607,CTCPUB=543,CTRSPT=543,EACSP=1,
 CV=0,CPV=0,CRV=0,CPUBV=0,CTRV=0,FE=0,
 ETE=0,ETEI=0,EREI=0,WMR=44,WMT=44,RF=0,
 $
```

SECTION 7 - SAMPLE INPUTS

7.2 Post Processor Sample Input

```
 NUS
 $L E=.000763,
 CUP=91623,CMP=7298,CPP=48,CUBEU=1,CUBEM=.3,CUBEP=.32,WU=36,WM=2,WP=.75,
 P=8,PP=40,CKIT=1883,WTKIT=10,ZO=.5,REPEAT=1,EE=4,
 TRC=1.2,TC=.1,TE=0,TER=0,TF=.50,TFR=1.5,TI=.50,TIR=1.5,
 TMD=1.5,TMDR=4,TMI=1.5,TMIR=4,
 TD=8,TDR=6,TMD=6,TMDR=16,
 SMF=.0046,
 G(12)=.5,G(13)=.5,H=0,1,1,0,
 FNGF=.49,
 CRI=5026,CI=20100,CLRUPG=7538,CMODPG=6282,CPI=12563,CTRI=4523,ETI=1,
 CPII=0,CRII=0,CII=0,CPUBII=0,CTRII=0,ETII=1,
 CALPUB=3518,CCAL=10050,CCALP=7538,CCALR=5025,CTRCAL=3518,EACAL=1,
 CCSP=18845,CCSPP=9548,CCSPR=4774,CTCPUB=4271,CTRSPT=4271,EACSP=1,
 $
LRU 7
CONUS
 $L E=.00005,
 CUP=37455,CMP=7489,CPP=18,CUBEU=2,CUBEM=.5,CUBEP=0,WU=5,WM=.5,WP=.01,
 P=1,PP=40,CKIT=500,WTKIT=10,ZO=.5,REPEAT=1,
 TRC=1.5,TC=.1,TE=1.0,TER=1.8,TF=0,TFR=0,TI=1.0,TIR=1.8,
 TD=4,TDR=6,TMD=4,TMDR=6,
 TMD=0,TMDR=0,TMI=.8,TMIR=2.5,
 SMF=.0046,
 G(11)=.4,G(15)=.4,G(20)=.2,H=1,1,1,1,
 FNGF=.33,
 CRI=331,CI=1319,CLRUPG=494,CMODPG=412,CPI=824,CTRI=297,ETI=1,
 CPII=732,CRII=457,CII=1280,CPUBII=257,CTRII=457,ETII=1,
 CALPUB=231,CCAL=659,CCALP=494,CCALR=330,CTRCAL=231,EACAL=1,
 CCSP=1236,CCSPP=626,CCSPR=313,CTCPUB=280,CTRSPT=280,EACSP=1,
 $
LRU 8
CONUS
 $L E=.000193,
 CUP=6627,CMP=3354,CPP=21,CUBEU=.5,CUBEM=.05,CUBEP=.01,WU=3,WM=.75,WP=.05,
 P=44,PP=20,CKIT=148,WTKIT=1,ZO=0,REPEAT=1,EE=4,
 TRC=.50,TC=.1,TE=.25,TER=.75,TF=.25,TFR=.75,TI=.25,TIR=.75,
 TD=4,TDR=8,TMD=2,TMDR=5,
 TMD=0,TMDR=0,
 G(20)=1,H=0,0,0,1,
 FNGF=.2,
 CRI=1271,CI=5085,CLRUPG=1907,CMODPG=1589,CPI=3178,CTRI=1144,ETI=1,
 CPII=2822,CRII=1764,CII=4938,CPUBII=988,CTRII=1764,ETII=1,
 CALPUB=890,CCAL=2542,CCALP=1907,CCALR=1271,CTRCAL=890,EACAL=1,
 CCSP=4768,CCSPP=2415,CCSPR=1208,CTCPUB=1081,CTRSPT=1081,EACSP=1,
 $
LRU 9
CONUS
 $L E=.001097,
 TRC=2.4,TC=.1,TE=0,TER=0,TF=0,TFR=0,TI=1.5,TIR=5.1,TMI=1.5,TMIR=4.5,
 TD=8,TDR=20,TMD=8,TMDR=20,
 CUP=139755,CMP=4641,CPP=17,CUBEU=15,CUBEM=.5,CUBEP=.05,WU=150,WM=15,WP=.5,
 P=13,PP=40,CKIT=7226,WTKIT=30,REPEAT=1,
 G(18)=.88,G(20)=.12,H=1,0,1,1,
 FNGF=.2,
 CRI=7226,CI=28900,CLRUPG=10838,CMODPG=9031,CPI=18052,CTRI=6502,ETI=1,
 CPII=16046,CRII=28080,CPUBII=5616,CTRII=10029,ETII=1,
 CALPUB=5057,CCAL=14449,CCALP=10838,CCALR=7225,CTRCAL=5057,EACAL=1,
 CCSP=27094,CCSPP=13727,CCSPR=6863,CTCPUB=6141,CTRSPT=6141,EACSP=1,
 $
LRU 10
CONUS
 $L
 E=.0024,
 TRC=2.4,TC=.1,TE=0,TER=0,TF=0,TFR=0,TI=1.5,TIR=5.1,TMI=1.5,TMIR=4.5,
```

7.2 Post Processor Sample Input (Continued)

```
 JP=139755,CMP=4641,CPP=17,CUBEU=15,CUBEM=.5,CUBEP=.05,dU=150,WM=15,dP=.5,
P=13,PP=40,CKIT=7226,WTKIT=30,ZO=.7,REPEAT=1,
SMF=.0046,
FNGF=.2,
G(14)=.25,G(19)=.25,G(20)=.5,M=1,1,1,1,
CRI=15801,CI=63225,CLRUPG=23709,CMODPG=19758,CPI=39516,CTRI=14225,ETI=1,
CPII=35104,CRII=21940,CII=61432,CPUBII=12286,CTRII=21940,
NU=-3,IO=3,IS=1,IFLAG=1,
CCAL=31612,CCALP=23709,CCSP=59274,CII=29080,CTCPUB=13435,
CTRCAL=11064,EACAL=1,EACSP=1,CALPUB=11064,
 T=1.,1.,0.,5.,1.,0.,0.,0.,0.,29804.,
 1.,6. ,0.,4.,1.,0.,0.,0.,0.,29804.,
 1.,8. ,0.,3.,1.,0.,0.,0.,0.,18718.,
 1.,10.,0.,3.,0.,0.,1.,0.,0.,14038.5,
 1.,5.,0.,2.,1.,0.,0.,0.,0.,21057.75,
 1.,1.,0.,2.,0.,0.,0.,0.,1.,18718.,
 1.,10.,0.,2.,0.,0.,1.,0.,0.,18718.,
 1.,10.,0.,.5,0.,0.,0.,0.,1.,18080.,
 1.,1.,0.,.5,1.,0.,0.,0.,0.,18080.,
 1.,1.,1.,9.,1.,0.,0.,0.,0.,16759.,
 1.,2.,1.,8.,1.,0.,0.,0.,0.,16765.75,
 1.,2.,1.,8.,0.,0.,1.,0.,0.,16765.75,
 1.,1.,1.,8.,0.,0.,0.,0.,1.,16759.,
 1.,3.,1.,7.,1.,0.,0.,0.,0.,16759.,
 1.,12.,1.,7.,0.,0.,1.,0.,0.,16759.,
 1.,9. ,1.,7.,0.,0.,0.,1.,0.,16759.,
 1.,10.,1.,7.,0.,0.,0.,0.,1.,16759.,
 1.,5.,1.,6.,1.,0.,0.,0.,0.,10469.,
 1.,19.,1.,6.,0.,0.,1.,0.,0.,10469.,
 1.,13.,1.,6.,0.,0.,0.,1.,0.,10469.,
 1.,3.,1.,6.,0.,0.,0.,0.,1.,10469.,
 1.,1.,1.,5.,0.,1.,0.,0.,0.,10469.,
 1.,88. ,1.,5.,0.,0.,1.,0.,0.,10469.,
 1.,48.,1.,5.,0.,0.,0.,1.,0.,10469.,
 1.,130.,1.,4.,0.,0.,1.,0.,0.,10469.,
 1.,81. ,1.,4.,0.,0.,0.,1.,0.,10469.,
 1.,134.,1.,3.,0.,0.,1.,0.,0.,7500.,
 1.,76. ,1.,3.,0.,0.,0.,1.,0.,7500.,
 2,1,.48,1769,247732,.295,8283,.295,7388.7,80052,
 2,2,.112,17068,.112,17068,.036,.036,687.741,
 2,3,.46,12011,0,0,0,0,0,0,
 3,0,0,600,5,1.32,.0001,0,0,200,
 3,0,0,400,4,1.32,.0001,0,0,62,
 3,0,0,400,4,1.12,.0001,0,0,53,
 3,0,0,324,138,1.50,.001,0,0,4,
 4,16000,303360,5000,50000,0,0,0,0,0,
 6,1,1000,50000,5000,10000,10000,0,0,3,
 6,2,10000,50000,10000,15000,10000,4000,0,3,
 7.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
 8.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
 $
SENSY ON OPERATION AVAILABILITY
AVAILABILITY DESIRED .6,.75,.9,.92
 $L SENSY=1,4,1,3,1.6,1.75,1.9,1.92,258*0,IFLAG=1 $
STOP 1
STOP 2
 $L NU=-4 $
```

SECTION 8

SAMPLE OUTPUTS

(1)
-30-
(2) MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE (3)

DATE - OCT 83 (4)

UNIT - LRU 1 (5)
CONUS (6)

LOGAM PROVISIONING QUANTITIES

	LRUS COMPUTED	LRUS REQUIRED	LRUS DISTRIBUTED	MODULES COMPUTED	MODULES DISTRIBUTED	PARTS COMPUTED	PARTS DISTRIBUTED	STOCK POINTS	M
ED.....	(7) 1.12	(8) 1.12	(9) 6.00	(10) .19	(11) 6.00	(12) 0.00	(13) 0.00	(14) 5.0	(15) 1.
OD.....	(16) .70	(17) .70	(18) 0.00	(19) 1.14	(20) 0.00	(21) 0.00	(22) 0.00	(23) 2.0	(24) 1.
DI.....	(25) .21	(26) .21	(27) 0.00	(28) .34	(29) 0.00	(30) 0.00	(31) 0.00	(32) 2.0	(33) 1.
DD.....	(34) .86	(35) .86	(36) 0.00	(37) 6.30	(38) 2.00	(39) 4.98	(40) 5.00	(41) 1.0	(42) 1.
TOTALS.....	(43) 2.89	(44) 2.89	(45) 6.00	(46) 7.98	(47) 8.00	(48) 4.98	(49) 5.00		

MAINTENANCE POLICIES

	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT
(50)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	.25	0.00	0.00	.50	0.00	0.00	0.00	0.00	.15	.10

(1)
-31-

(2) MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE (3) DATE - OCT 83 (4)

UNIT - LRU 1 (5)
CONUS (6)

PRESENT VALUE COST TOTALS IN (7) THOUSANDS OF DOLLARS AVAILABILITY= (8) .973590 INHERENT= (9) .999689

(10) PRIME	(11) T.E.	(12) TE SPACE	(13) MANPOWER	(14) SUPPLY	(15) ORDERING	(16) STORAGE	(17) S.ADMIN	(18) SHIPPING	(19) TOTAL
0.	2021.	1.	12294.	1056.	48.	0.	480.	3.	15903.

	INSTALLED EQUIPMENT	*INITIAL** PROVISIONS	*REORDER** ACTIONS	*REORDER** LOTS	*REORDER** QUANITY	*CONSUMED* STOCK	*RESIDUAL* STOCK
UNITS.....	(20) 24.	(21) 6.	(22) 1.	(23) 1.	(24) 1.	(25) 6.	(26) 1.
MODULES...		(27) 8.	(28)15.	(29) 2.	(30) 30.	(31) 36.	(32) 2.
PARTS.....		(33) 5.	(34)22.	(35) 11.	(36)242.	(37)236.	(38) 11.

** TEST EQP AND REPAIR CHANNEL MMH,S ** ** PER HOUR PER MAINTENANCE LOCATION **	***ORGANIZATION**		******DIRECT*****		*****GENERAL*****		*******DEPOT*****	
	TEST	REPAIR	TEST	REPAIR	TEST	REPAIR	TEST	REPAIR
EACH LRU CASE (47) (48)	(39).0002 (40) .0002		(41) .0008	(42) .0009	(43) .0002	(44) .0003	(45).0068	(46) .0175
CUM FOR LRU CASES 11-11	(49).0002 (50) .0002		(51) .0008	(52) .0009	(53) .0002	(54) .0003	(55).0068	(56) .0175
CUM FOR ALL LRU CASES	(57).0002 (58) .0002		(59) .0008	(60) .0009	(61) .0002	(62) .0003	(63).0068	(64) .0175

(65) (66) TYPE I TEST EQP POSTED FOR LRUS 11-11	***TEST HRS***	**REPAIR HRS**	***TEST EQP***	***TEST MEN***	**REPAIR MEN**
ORGANIZATION	(67) .0002	(68) .0002	(69) .0036	(70) .0036	(71) .0035
DIRECT SUPPORT	(72) .0008	(73) .0009	(74) .0061	(75) .0061	(76) .0070
GENERAL SUPPORT	(77) .0002	(78) .0003	(79) .0018	(80) .0018	(81) .0021
(87) DEPOT	(82) 0.0000	(83) 0.0000	(84) 0.0000	(85) 0.0000	(86) 0.0000
TYPE II TEST EQP POSTED FOR LRUS 11-11(88)					
DEPOT	(89) .0069	(90) .0175	(91) .0289	(92) .0289	(93) .0735

COST OF INITIAL PROVISION

	EQPT.	DIRECT	GENERAL	DEPOT	TOTAL	RESIDUAL
UNIT	(94) 629.	(95) 0.	(96) 0.	(97) 0.	(98) 629.	(99) 102.
MODULE	(100) 23.	(101) 0.	(102) 0.	(103) 8.	(104) 30.	(105) 6.
PART		(106) 0.	(107) 0.	(108) 0.	(109) 0.	(110) 0.

	TOTALS	CUM	MANPOWER	DELTA
PRESENT VALUE COSTS	(111)15903.	(112)15903.	(113)12294.	(114) 0.
EXPECTED VALUE COSTS	(115)15903.	(116)15903.	(117)12294.	(118) 0.

(1)
-59-

(2) MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE (4) DATE - OCT 83 (6)

UNIT - LRU 1 (3)
TOTAL (5)

PRESENT VALUE COST TOTAL THOUSANDS OF DOLLARS (7)

PRIME	T.E.	TE SPACE	MANPOWER	SUPPLY	ORDERING	STORAGE	S.ADMIN	SHIPPING	TOTAL
(10) 0.	(11) 4069.	(12) 5.	(13) 29673.	(14) 3154.	(15) 97.	(16) 1.	(17) 1279.	(18) 11.	(19) 38288.

	INSTALLED EQUIPMENT	*INITIAL** PROVISIONS	*REORDER** ACTIONS	*REORDER** LOTS	*REORDER** QUANITY	*CONSUMED* STOCK	*RESIDUAL* STOCK
UNITS.....	(20) 64.	(21) 16.	(22) 5.	(23) 1.	(24) 5.	(25) 20.	(26) 1.
MODULES...		(27) 30.	(28) 28.	(29) 1.	(30) 95.	(31) 121.	(32) 4.
PARTS.....		(33) 20.	(34) 44.	(35) 10.	(36) 792.	(37) 789.	(38) 23.

** TEST EQP AND REPAIR CHANNEL MMH,S ** ** PER HOUR PER MAINTENANCE LOCATION **	***ORGANIZATION** TEST	REPAIR	******DIRECT***** TEST	REPAIR	*****GENERAL***** TEST	REPAIR	*******DEPOT***** TEST	REPAIR
EACH LRU CASE (47) (48)	(39) .0004	(40) .0004	(41) .0011	(42) .0013	(43) .0004	(44) .0005	(45) .0225	(46) .0585
CUM FOR LRU CASES 1- 1	(49) .0004	(50) .0004	(51) .0011	(52) .0013	(53) .0004	(54) .0005	(55) .0225	(56) .0585
CUM FOR ALL LRU CASES	(57) .0004	(58) .0004	(59) .0011	(60) .0013	(61) .0004	(62) .0005	(63) .0225	(64) .0585

(65) (66)

TYPE I TEST EQP POSTED FOR LRUS 1- 1	***TEST HRS***	**REPAIR HRS**	***TEST EQP***	***TEST MEN***	**REPAIR MEN**
ORGANIZATION	(67) .0004	(68) .0004	(69) .0119	(70) .0119	(71) .0117
DIRECT SUPPORT	(72) .0012	(73) .0013	(74) .0203	(75) .0203	(76) .0234
GENERAL SUPPORT	(77) .0004	(78) .0005	(79) .0061	(80) .0061	(81) .0070
DEPOT	(82) 0.0000	(83) 0.0000	(84) 0.0000	(85) 0.0000	(86) 0.0000
TYPE II TEST EQP POSTED FOR LRUS 1- 1 (88)					
(87) DEPOT	(89) .0230	(90) .0595	(91) .0966	(92) .0966	(93) .2455

QUANTITY OF INITIAL PROVISIONS

	EQPT.	DIRECT	GENERAL	DEPOT
UNIT	(94) 16.	(95) 0.	(96) 0.	97) 0.
MODULE	(98) 16.	(99) 0.	(100) 0.	(101) 14.
PART		(102) 0.	(103) 0.	(104) 20.

COST OF INITIAL PROVISION

	EQPT.	DIRECT	GENERAL	DEPOT	TOTAL	RESIDUAL
UNIT	(105) 1679.	(106) 0.	(107) 0.	(108) 0.	(109) 1679.	(110) 121.
MODULE	(111) 51.	(112) 0.	(113) 0.	(114) 53.	(115) 113.	(116) 14.
PART		(117) 0.	(118) 0.	(119) 1.	(120) 1.	(121) 1.

	TOTALS	CUM	MANPOWER	DELTA
PRESENT VALUE COSTS	(122) 38288.	(123) 38288.	(124) 29673.	(125) 0.
EXPECTED VALUE COSTS	(126(38288.	(127) 38288.	(128) 29673.	(129) 0.

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

(1)	CASE TOTAL
COST TOTALS, COST IN THOUSANDS OF DOLLARS	
INSTALLED EQUIPMENT	0. (2)
TEST EQUIPMENT	10528. (4)
TEST EQUIPMENT SPACE	5. (6)
MAINTENANCE MANPOWER	115024. (7)
SUPPLY MATERIAL	5711. (14)
REORDERING	298. (16)
MATERIAL STORAGE	14. (18)
SUPPLY ADMINISTRATION	3413. (20)
SHIPPING AND HANDLING	34. (22)
GRAND TOTAL COST	135028. (24)
PRESENT VALUE	
DEVELOPMENT	2670. (26)
ACQUISITION	4367. (28)
OPERATION AND MAINTENANCE	127991. (30)
END LIFE SALVAGE	0. (32)
GRAND TOTAL	135028. (34)

FIELD (8)	96988.
TRAINING FIELD (11)	17969.

RECURRING COSTS	
T.E. MAINTENANCE	7232. (3)
DEPOT SPACE/UTILITIES	5. (5)
DEPOT (9)	642. TOTAL 97533. (10)
DEPOT (12)	5. TOTAL 17374. (13)
SUPPLIES	2579. (15)
REORDERING	298. (17)
MATERIAL STORAGE	14. (19)
INVENTORY MANAGEMENT	3155. (21)
SHIPPING	34. (23)
TOTAL RECURRING	127991. (25)

COST OF INITIAL PROVISION	
UNITS	2955. (27)
MODULES	155. (29)
PARTS	1. (31)
TOTAL PROVISION	3132. (33)

EXPECTED VALUE MANPOWER AT EQUIPMENT DIRECT AND GENERAL

MAINTENANCE MANPOWER	115023. (35)
GRAND TOTAL COST	135027. (36)
PRESENT VALUE	
OPERATION AND MAINTENANCE	127990. (37)
GRAND TOTAL	135027. (38)

DELTA 1.(39) PV DELTA 1.(40)

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

GRAND TOTAL

COST TOTALS, COST IN THOUSANDS OF DOLLARS (1)				RECURRING COSTS	
INSTALLED EQUIPMENT	0.(2)			T.E. MAINTENANCE	14488.(3)
TEST EQUIPMENT	21146.(4)			DEPOT SPACE/UTILITIES	18.(5)
TEST EQUIPMENT SPACE	18.(6)			DEPOT (9) 2156. TOTAL	233531.(10)
MAINTENANCE MANPOWER	274946.(7)	FIELD)(8)231444.		DEPOT (12) 17. TOTAL	40706.(13)
		TRAINING FIELD)(11) 40688.		SUPPLIES	9477.(15)
SUPPLY MATERIAL	18336.(14)			REORDERING	631.(17)
REORDERING	631.(16)			MATERIAL STORAGE	56.(19)
MATERIAL STORAGE	56.(18)			INVENTORY MANAGEMENT	8413.(21)
SUPPLY ADMINISTRATION	9103.(20)			SHIPPING	123.(23)
SHIPPING AND HANDLING	123.(22)				
GRAND TOTAL COST	324358.(24)			TOTAL RECURRING	305512.(25)
PRESENT VALUE				COST OF INITIAL PROVISION	
DEVELOPMENT	5339.(26)			UNITS	9251.(27)
ACQUISITION	12506.(28)			MODULES	595.(29)
OPERATION AND MAINTENANCE	306512.(30)			PARTS	3.(31)
END LIFE SALVAGE	0.(32)			TOTAL PROVISION	9859.(33)
GRAND TOTAL	324358.(34)				

EXPECTED VALUE MANPOWER AT EQUIPMENT DIRECT AND GENERAL

MAINTENANCE MANPOWER	274943.(35)				
GRAND TOTAL COST	324355.(36)				
PRESENT VALUE					
OPERATION AND MAINTENANCE	306510.(37)				
GRAND TOTAL	324355.(38)	DELTA	3.(39)	PV DELTA	3.(40)

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

CASE TOTAL

**** TEST AND REPAIR MANPOWER REQUIREMENTS

	MAN-E	EQUIP	DIRECT	GENERAL	DEPOT
HRS PER YR ALL MAINT LOC					
TEST EQUIPMENT	183.200(1)	21.553(2)	28.468(3)	25.854(4)	232.556(5)
REPAIR	17420.114(6)	23.135(7)	44.206(8)	62.688(9)	551.049(10)
NO OF MEN PER ANY TIME UNIT ALL MAINT LOC					
TEST EQUIPMENT	6.000(11)	.019(12)	.025(13)	.023(14)	.156(15)
REPAIR	12.000(16)	.020(17)	.039(18)	.055(19)	.370(20)
HRS PER YR PER MAINT LOC					
TEST EQUIPMENT	30.533(21)	3.592(22)	14.234(23)	12.927(24)	232.556(25)
REPAIR	2903.352(26)	3.856(27)	22.103(28)	31.344(29)	551.049(30)
NO OF MEN PER ANY TIME UNIT PER MAINT LOC					
TEST EQUIPMENT	1.000(31)	.003(32)	.012(33)	.011(34)	.156(35)
REPAIR	2.000(36)	.003(37)	.019(38)	.027(39)	.370(40)

****SYSTEM/SUBSYSTEM AVAILABILITIES ****

CAYZ= .122988 (41)

CAYZI= .960982 (42)

-55-

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

CASE TOTAL

SYSTEM MAINTENANCE SUPPORT COSTS

		COST	PERCENTAGE
1.000	RESEARCH AND DEVELOPMENT		
1.010	DEVELOPMENT ENGINEERING	1358.14(1)	100.00(2)
TOTAL		1358.14(3)	100.00(4)
2.000	INVESTMENT COST		
2.010	NON-RECURRING INVESTMENT	0.00(5)	0.00(6)
2.050	DATA	1311.58(7)	24.20(8)
2.080	TRAINING SERVICES AND EQUIPMENT	319.63(9)	5.90(10)
2.090	INITIAL SPARES AND REPAIR PARTS	3132.41(11)	57.83(12)
2.11	OTHER	655.99(13)	12.10(14)
TOTAL		5419.61(15)	100.00(16)
3.000	OPERATING AND SUPPORT COST		
3.010	MILITARY PERSONNEL		
3.011	CREW PAY AND ALLOWANCES	0.00(17)	0.00(18)
3.012	MAINTENANCE PAY AND ALLOWANCES	96986.31(19)	75.62(20)
3.013	INDIRECT PAY AND ALLOWANCES	0.00(21)	0.00(22)
3.014	PERMANENT CHANGE OF STATION	0.00(23)	0.00(24)
3.020	CONSUMPTION		
3.021	REPLENISHMENT SPARES	7481.09(25)	5.83(26)
3.022	PETROLEUM, OIL AND LUBRICANTS	0.00(27)	0.00(28)
3.023	UNIT TRAINING AMMUNITION AND MISSILE	0.00(29)	0.00(30)
3.030	DEPOT MAINTENANCE		
3.031	LABOR	649.31(31)	.51(32)
3.032	MATERIEL	1555.69(33)	1.21(34)
3.033	TRANSPORTATION	1.00(35)	.00(36)
3.040	MODIFICATIONS MATERIAL	738.62(37)	.58(38)
3.050	OTHER DIRECT SUPPORT OPERATIONS		
3.051	MAINTENANCE, CIVILIAN LABOR	0.00(39)	0.00(40)
3.052	OTHER DIRECT	3725.74(41)	2.91(42)
3.060	INDIRECT SUPPORT OPERATIONS		
3.061	PERSONNEL REPLACEMENT	17073.95(43)	13.31(44)
3.062	TRANSIENTS, PATIENTS AND PRISONERS	0.00(45)	0.00(46)
3.063	QUARTERS, MAINTENANCE AND UTILITIES	0.00(47)	0.00(48)
3.064	MEDICAL SUPPORT	0.00(49)	0.00(50)
3.065	OTHER INDIRECT	38.26(51)	.03(52)
TOTAL		128249.98(53)	100.00(54)
GRAND TOTAL		135027.73(55)	

-56-

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

CASE TOTAL

SYSTEM OPERATIONS AND SUPPORT COSTS

		COST	PERCENTAGE
1.000	RESEARCH AND DEVELOPMENT		
1.010	DEVELOPMENT ENGINEERING	1358.14 (1)	100.00 (2)
TOTAL		1358.14 (3)	100.00 (4)
2.000	INVESTMENT COST		
2.010	NON-RECURRING INVESTMENT	0.00 (5)	0.00 (6)
2.050	DATA	1311.58 (7)	24.20 (8)
2.080	TRAINING SERVICES AND EQUIPMENT	319.63 (9)	5.90 (10)
2.090	INITIAL SPARES AND REPAIR PARTS	3132.41 (11)	57.80 (12)
2.11	OTHER	655.99 (13)	12.10 (14)
TOTAL		5419.61 (15)	100.00 (16)
3.000	OPERATING AND SUPPORT COST		
3.010	MILITARY PERSONNEL		
3.011	CREW PAY AND ALLOWANCES	80967.15 (17)	26.67 (18)
3.012	MAINTENANCE PAY AND ALLOWANCES	96986.31 (19)	31.95 (20)
3.013	INDIRECT PAY AND ALLOWANCES	17940.30 (21)	5.91 (22)
3.014	PERMANENT CHANGE OF STATION	16651.81 (23)	5.48 (24)
3.020	CONSUMPTION		
3.021	REPLENISHMENT SPARES	7481.09 (25)	2.46 (26)
3.022	PETROLEUM, OIL AND LUBRICANTS	7.40 (27)	.00 (28)
3.023	UNIT TRAINING AMMUNITION AND MISSILE	13987.20 (29)	4.61 (30)
3.030	DEPOT MAINTENANCE		
3.031	LABOR	649.31 (31)	.21 (32)
3.032	MATERIEL	1555.69 (33)	.51 (34)
3.033	TRANSPORTATION	1.00 (35)	.00 (36)
3.040	MODIFICATIONS MATERIAL	738.62 (37)	.24 (38)
3.050	OTHER DIRECT SUPPORT OPERATIONS		
3.051	MAINTENANCE, CIVILIAN LABOR	0.00 (39)	0.00 (40)
3.052	OTHER DIRECT	8684.11 (41)	2.86 (42)
3.060	INDIRECT SUPPORT OPERATIONS		
3.061	PERSONNEL REPLACEMENT	48312.75 (43)	15.91 (44)
3.062	TRANSIENTS, PATIENTS AND PRISONERS	7284.31 (45)	2.40 (46)
3.063	QUARTERS, MAINTENANCE AND UTILITIES	714.48 (47)	.24 (48)
3.064	MEDICAL SUPPORT	0.00 (49)	0.00 (50)
3.065	OTHER INDIRECT	1639.34 (51)	.54 (52)
TOTAL		303600.87 (53)	100.00 (54)
GRAND TOTAL		310378.62 (55)	

-58-

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

GRAND TOTAL

SYSTEM MAINTENANCE SUPPORT COSTS

		COST	PERCENTAGE
1.000	RESEARCH AND DEVELOPMENT		
1.010	DEVELOPMENT ENGINEERING	2716.28(1)	100.00 (2)
TOTAL		2716.28(3)	100.00 (4)
2.000	INVESTMENT COST		
2.010	NON-RECURRING INVESTMENT	0.00(5)	0.00 (6)
2.050	DATA	2623.15(7)	18.17 (8)
2.080	TRAINING SERVICES AND EQUIPMENT	639.26(9)	4.43 (10)
2.090	INITIAL SPARES AND REPAIR PARTS	9858.68(11)	68.28 (12)
2.11	OTHER	1318.48(13)	9.13 (14)
TOTAL		14439.57(15)	100.00 (16)
3.000	OPERATING AND SUPPORT COST		
3.010	MILITARY PERSONNEL		
3.011	CREW PAY AND ALLOWANCES	0.00(17)	0.00 (18)
3.012	MAINTENANCE PAY AND ALLOWANCES	231435.82(19)	75.34 (20)
3.013	INDIRECT PAY AND ALLOWANCES	0.00(21)	0.00 (22)
3.014	PERMANENT CHANGE OF STATION	0.00(23)	0.00 (24)
3.020	CONSUMPTION		
3.021	REPLENISHMENT SPARES	15708.80(25)	5.11 (26)
3.022	PETROLEUM, OIL AND LUBRICANTS	0.00(27)	0.00 (28)
3.023	UNIT TRAINING AMMUNITION AND MISSILE	0.00(29)	0.00 (30)
3.030	DEPOT MAINTENANCE		
3.031	LABOR	2181.73(31)	.71 (32)
3.032	MATERIEL	5220.18(33)	1.70 (34)
3.033	TRANSPORTATION	2.93(35)	.00 (36)
3.040	MODIFICATIONS MATERIAL	2019.36(37)	.66 (38)
3.050	OTHER DIRECT SUPPORT OPERATIONS		
3.051	MAINTENANCE, CIVILIAN LABOR	0.00(39)	0.00 (40)
3.052	OTHER DIRECT	9789.83(41)	3.19 (42)
3.060	INDIRECT SUPPORT OPERATIONS		
3.061	PERSONNEL REPLACEMENT	40705.54(43)	13.25 (44)
3.062	TRANSIENTS, PATIENTS AND PRISONERS	0.00(45)	0.00 (46)
3.063	QUARTERS, MAINTENANCE AND UTILITIES	0.00(47)	0.00 (48)
3.064	MEDICAL SUPPORT	0.00(49)	0.00 (50)
3.065	OTHER INDIRECT	138.16(51)	.04 (52)
TOTAL		307202.36(53)	100.00 (54)
GRAND TOTAL		324358.22(55)	

SECTION 8 - SAMPLE OUTPUTS

8.7.1 Sensitivity Output (Inhibited print)

DATE - OCT 83

-19-

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

	(3)		(4)		(5)	(6)	
SENSITIVITY PASS	1	LRU	1	VAR(	3)=	.1600000E+01	VAR(
SENSITIVITY PASS	1	LRU	2	VAR(	3)=	.1600000E+01	VAR(
SENSITIVITY PASS	1	LRU	3	VAR(	3)=	.1600000E+01	VAR(
SENSITIVITY PASS	1	LRU	4	VAR(	3)=	.1600000E+01	VAR(

-146-

MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE

DATE - OCT 83

UNIT - LRU 1 (1)
CONUS (2)

(3) (4) (5) (6)

SENSITIVITY PASS 2 LRU 11 VAR(3)= .1750000E+01 VAR(

LOGAM PROVISIONING QUANTITIES

	LRUS COMPUTED	LRUS REQUIRED	LRUS DISTRIBUTED	MODULES COMPUTED	MODULES DISTRIBUTED	PARTS COMPUTED	PARTS DISTRIBUTED	STOCK POINTS	
ED.....	1.12	.00	0.00	.19	6.00	0.00	0.00	6.0	1.
OD.....	.70	0.00	0.00	1.14	0.00	0.00	0.00	2.0	1.
DI.....	.21	0.00	0.00	.34	0.00	0.00	0.00	2.0	1.
DD.....	.86	0.00	0.00	6.30	2.00	4.98	5.00	1.0	1.
TOTALS.....	2.89	.00	0.00	7.98	8.00	4.98	5.00		

*** COLUMN 2 ABOVE IS THE QTY OF LRUS REQUIRED TO MEET AN OPER. AVAIL. OF .750000 ***

(51)

MAINTENANCE POLICIES

GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	.25	0.00	0.00	.50	0.00	0.00	0.00	0.00	.15	.15

-147-
MISSILE SYSTEM
CASE 1(SENSY)
FY83 COSTS

ANALYSIS - SAMPLE CASE DATE - OCT 83

UNIT - LRU 1 (1)
CONUS (2)

(3) (4) (5) (6)
SENSITIVITY PASS 2 LRU 11 VAR(3)= .1750000E+01 VAR(

PRESENT VALUE COST TOTALS IN THOUSANDS OF DOLLARS AVAILABILITY= .824157 INHERENT= .999689

PRIME	T.E.	TE SPACE	MANPOWER	SUPPLY	ORDERING	STORAGE	S.ADMIN	SHIPPING	TOTAL
0.	2017.	1.	12263.	893.	45.	0.	467.	3.	15589.

	INSTALLED EQUIPMENT	*INITIAL** PROVISIONS	*REORDER** ACTIONS	*REORDER** LOTS	*REORDER** QUANTITY	*CONSUMED* STOCK	*RESIDUAL* STOCK
UNITS.....	24.	0.	6.	1.	6.	5.	1.
MODULES...		8.	12.	2.	24.	31.	1.
PARTS.....		5.	18.	11.	198.	230.	3.

** TEST EQP AND REPAIR CHANNEL MMH/S ** ** PER HOUR PER MAINTENANCE LOCATION **	***ORGANIZATION** TEST	REPAIR	******DIRECT***** TEST	REPAIR	*****GENERAL***** TEST	REPAIR	*******DEPOT***** TEST	REPAIR
EACH LRU CASE	.0001	.0001	.0007	.0008	.0002	.0002	.0057	.0148
CUM FOR LRU CASES 11-11	.0001	.0001	.0007	.0008	.0002	.0002	.0057	.0148
CUM FOR ALL LRU CASES	.0001	.0001	.0007	.0008	.0002	.0002	.0057	.0148

TYPE I TEST EQP POSTED FOR LRUS 11-11	***TEST HRS***	**REPAIR HRS**	***TEST EQP***	***TEST MEN***	**REPAIR MEN**
ORGANIZATION	.0001	.0001	.0030	.0030	.0030
DIRECT SUPPORT	.0007	.0008	.0051	.0051	.0059
GENERAL SUPPORT	.0002	.0002	.0015	.0015	.0018
DEPOT	0.0000	0.0000	0.0000	0.0000	0.0000
TYPE II TEST EQP POSTED FOR LRUS 11-11					
DEPOT	.0058	.0148	.0245	.0245	.0622

COST OF INITIAL PROVISION

	EQPT.	DIRECT	GENERAL	DEPOT	TOTAL	RESIDUAL
UNIT	0.	0.	0.	0.	0.	94.
MODULE	23.	0.	0.	8.	30.	5.
PART		0.	0.	0.	0.	0.

	TOTALS	CUM	MANPOWER	DELTA
PRESENT VALUE COSTS	15689.	15689.	12263.	0.
EXPECTED VALUE COSTS	15689.	15689.	12263.	0.

SECTION 8 - SAMPLE OUTPUTS

8.8 LOGAM NAMELIST/L/ Outputs

(1) 1LRU 1	(3) .40	.20	1.00	985.	25926.	00000	2.00	30119.	25997.	17057.	439%6.	(4)
(5) 2LRU 2	.40	.20	1.00	985.	25926.	2859.	2.00	1886.	4413.	2943.	9590.	
3LRU 3	.40	.20	1.00	985.	25926.	00000	2.00	89569.	97178.	40784.	167941.	
4LRU 4	.40	.20	1.00	985.	25926.	3444.	2.00	9839.	7388.	4921.	18459.	
5LRU 5	.40	.20	1.00	985.	25926.	447.	2.00	1295.	958.	638.	2389.	
6LRU 6	.40	.20	1.00	985.	25926.	3518.	2.00	13093.	9538.	5029.	18869.	
7LRU 7	.40	.20	1.00	985.	25926.	231.	2.00	859.	696.	330.	1236.	
8LRU 8	.40	.20	1.00	985.	25926.	898.	2.00	2582.	1987.	1271.	4768.	
9LRU 9	.40	.20	1.00	985.	25926.	9857.	2.00	40447.	29838.	7229.	27804.	
10LRU 10	.40	.20	1.00	985.	25926.	00000	2.00	31512.	23709.	7229.	99274.	
11LRU 1	.40	.20	1.00	991.	25926.	00000	2.00	30119.	25987.	17057.	439%6.	
12LRU 2	.40	.20	1.00	991.	25926.	2859.	2.00	1886.	4413.	2943.	9590.	
13LRU 3	.40	.20	1.00	991.	25926.	00000	2.00	89569.	97178.	40784.	167941.	
14LRU 4	.40	.20	1.00	991.	25926.	3444.	2.00	9839.	7388.	4921.	18459.	
15LRU 5	.40	.20	1.00	991.	25926.	447.	2.00	1295.	958.	638.	2389.	
16LRU 6	.40	.20	1.00	991.	25926.	3518.	2.00	13393.	9538.	5029.	18869.	
17LRU 7	.40	.20	1.00	991.	25926.	231.	2.00	859.	696.	330.	1236.	
18LRU 8	.40	.20	1.00	991.	25926.	898.	2.00	2582.	1987.	1271.	4768.	
19LRU 9	.40	.20	1.00	991.	25926.	9857.	2.00	40447.	29838.	7229.	27804.	
20LRU 10	.40	.20	1.00	991.	25926.	00000	2.00	31512.	23709.	7229.	99274.	

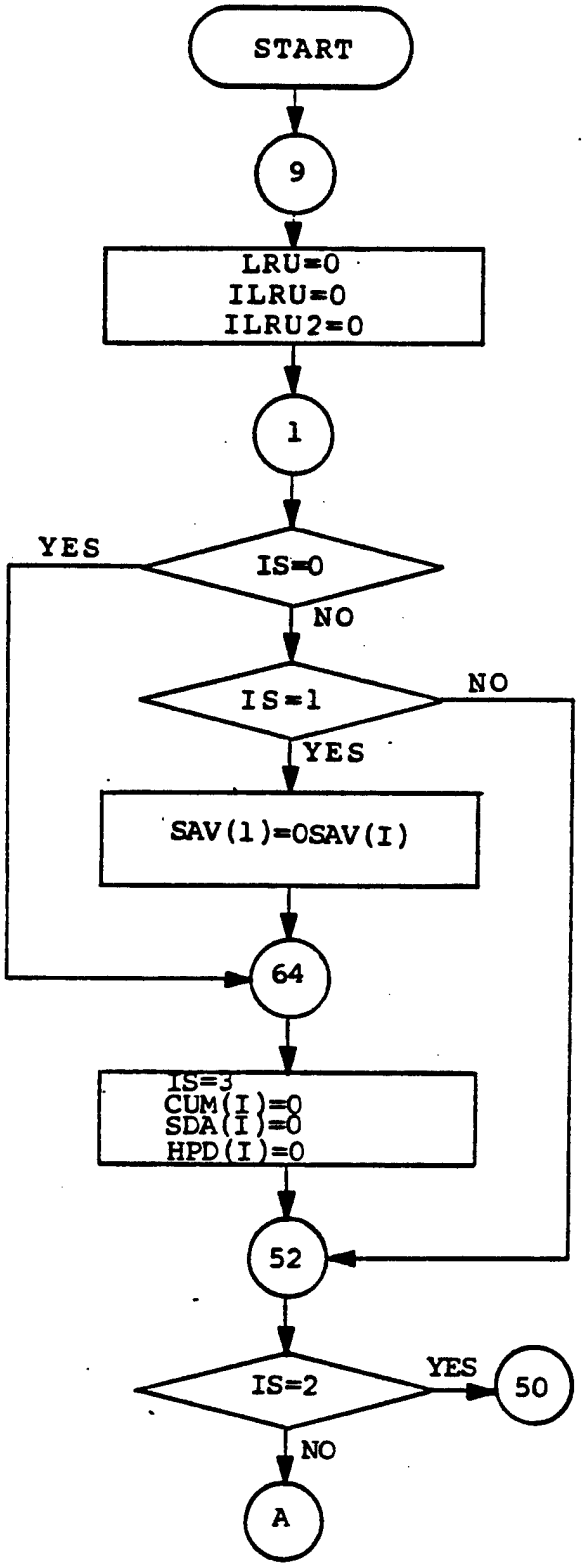
	CCSPP	CCSPR	CD01	CD02	CD03	CD10	CD10	CDIST	CDMAN	CD0E	CD31
1LRU 1	(6) 32409.	16205.	.30	.09	.21	.30	.13	.35	25926.	.09	.13
2LRU 2	5390.	2795.	.30	.09	.21	.30	.10	.35	25926.	.09	.10
3LRU 3	89398.	42545.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
4LRU 4	9367.	4674.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
5LRU 5	1213.	607.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
6LRU 6	9548.	4774.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
7LRU 7	626.	313.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
8LRU 8	2415.	1208.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
9LRU 9	13727.	6863.	.30	.09	.21	.30	.13	.35	25926.	.09	.13
10LRU 10	13727.	6863.	.30	.09	.21	.30	.10	.35	25926.	.09	.13
11LRU 1	32409.	16205.	.21	.09	.21	.21	.13	.35	25926.	.09	.13
12LRU 2	5390.	2795.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
13LRU 3	89398.	42545.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
14LRU 4	9367.	4674.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
15LRU 5	1213.	607.	.21	.09	.21	.21	.13	.35	25926.	.09	.13
16LRU 6	9548.	4774.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
17LRU 7	626.	313.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
18LRU 8	2415.	1208.	.21	.09	.21	.21	.10	.35	25926.	.09	.13
19LRU 9	13727.	6863.	.21	.09	.21	.21	.13	.35	25926.	.09	.13
20LRU 10	13727.	6863.	.21	.09	.21	.21	.13	.35	25926.	.09	.13

	CDPMAN	CDPERM	CDRMAN	CERMAN	CERM	CERMO	CERTAN	CFF3	CDMIN	CDIMAN
1LRU 1	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
2LRU 2	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
3LRU 3	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
4LRU 4	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
5LRU 5	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
6LRU 6	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
7LRU 7	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
8LRU 8	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
9LRU 9	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
10LRU 10	61409.	61409.	25926.	25926.	1816.	0.	25926.	1.33	25926.	25926.
11LRU 1	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
12LRU 2	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
13LRU 3	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
14LRU 4	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
15LRU 5	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
16LRU 6	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
17LRU 7	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
18LRU 8	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
19LRU 9	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.
20LRU 10	61409.	61409.	25926.	25926.	969.	0.	25926.	1.33	25926.	25926.

SECTION 9

LOGAM FLOWCHART

SECTION 9 - LOGAM FLOWCHART



Initialize Grand Total Accumulators and Read One-Time Inputs (8 Cards).

Return Here After Completing a Sensitivity Run (KAD=9).

Initialize LRU Counters.

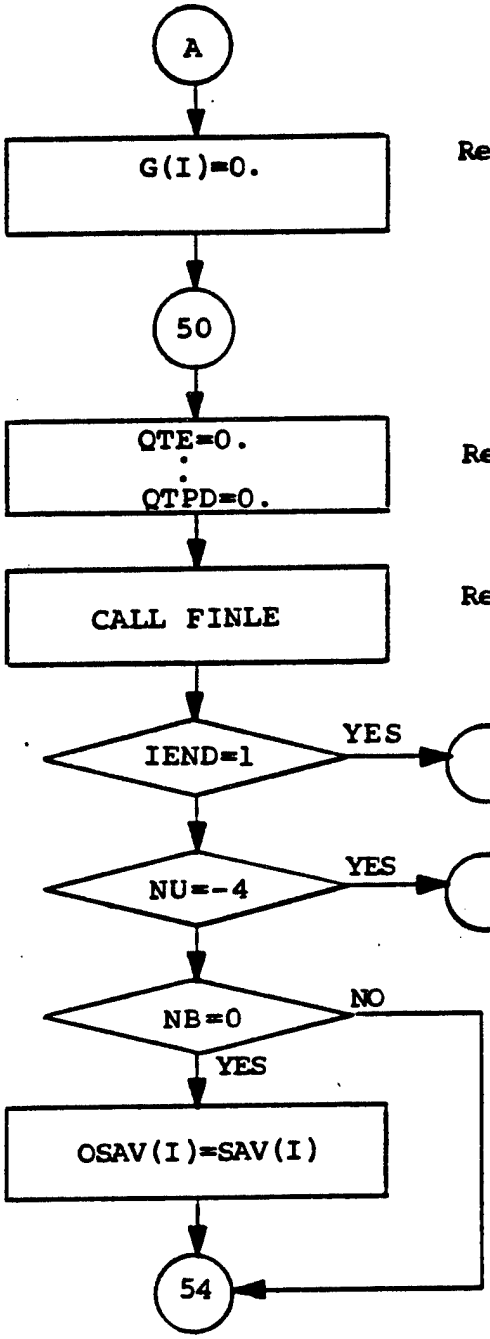
Return Here After an Individual LRU Case (KAD=1).

Recall a Previous LRU case.

Initialize Case Total Accumulators.

By-pass Resetting Maintenance Policies.

SECTION 9 - LOGAM FLOWCHART



Reset Maintenance Policies.

Reset Initial Provisions.

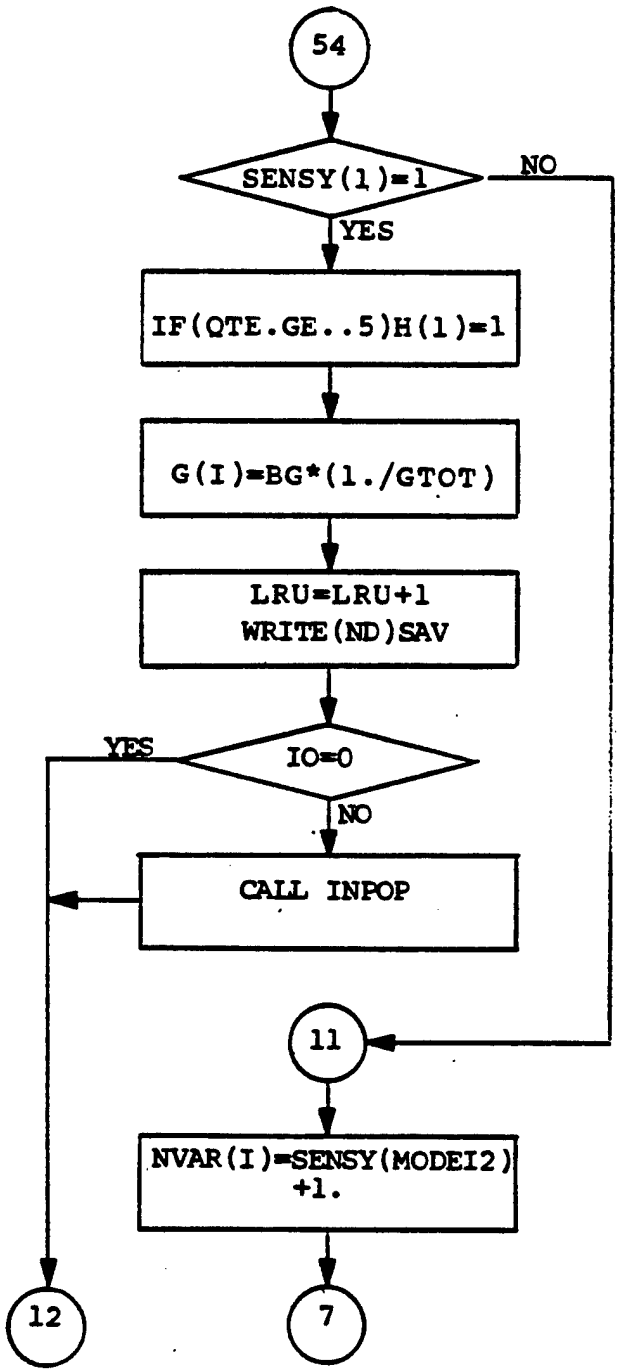
Read NAMELIST Inputs.

End-of-File in NAMELIST.

Positive Program Stop.

Store Input Case for Later Recall.

SECTION 9 - LOGAM FLOWCHART



Set 4 Stock Authorization Flags if Provisions are Input.

Set Maintenance Policies if Total of Input Values not 100%.

Increment LRU Counters and Write Input Case to Working File.

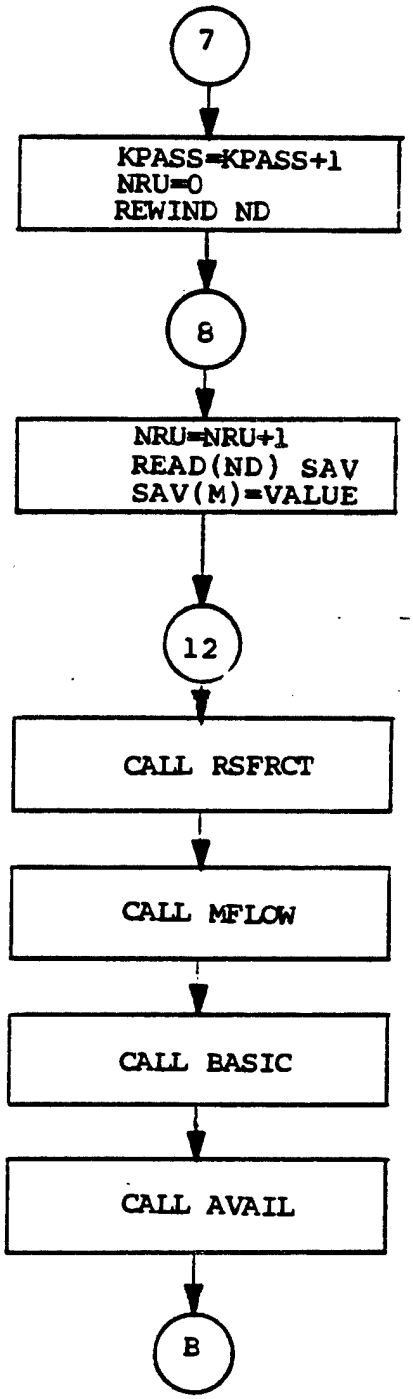
Print Input Values in Alphabetical Order.

Enter Here for a Sensitivity Run.

Store Positions for Variables to be Modified in a SENSY Run.

Return Here after Completion of a Sensitivity Pass (KAD=7).

SECTION 9 - LOGAM FLOWCHART



Start a New Pass of Sensitivity Runs.

Return Here After Completion of an LRU Case in a Sensitivity Run (KAD=8).

Read Inputs for a Sensitivity Case and Modify Values of Sensy Input.

Begin Evaluation of an Individual LRU Case.

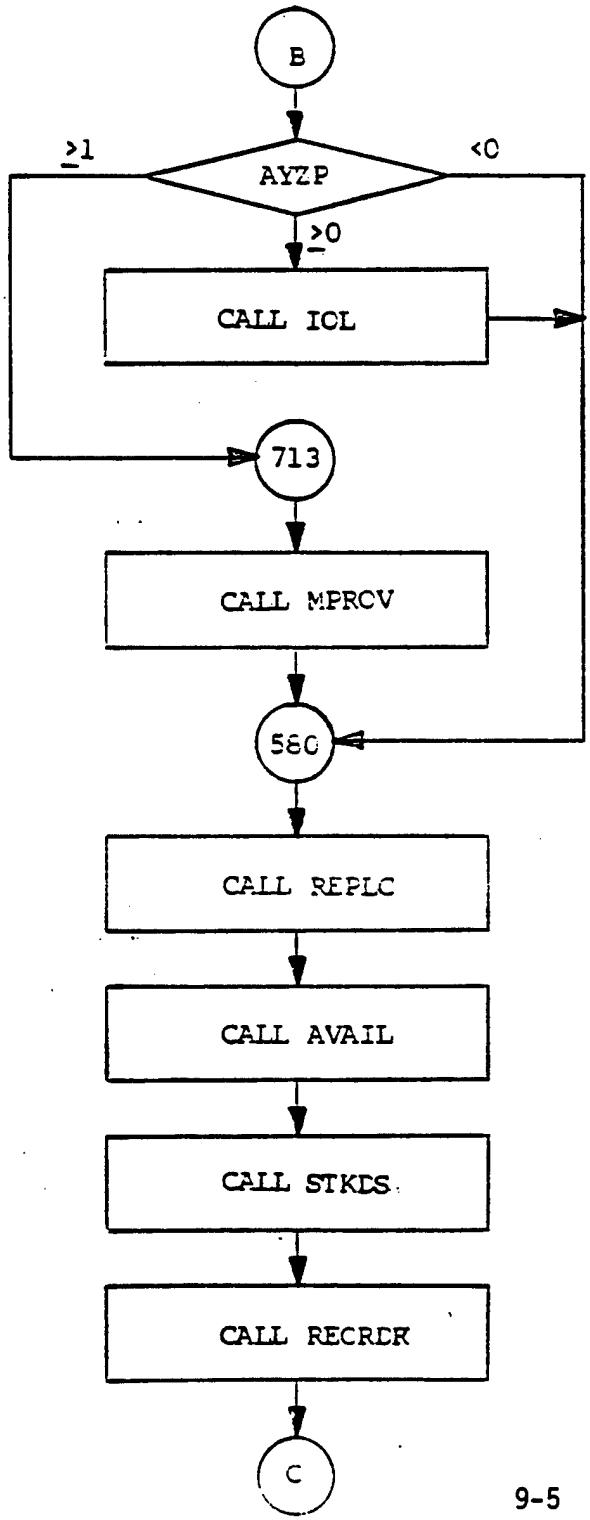
Compute Repair/Scrap Fractions.

Compute Maintenance Flows.

Compute Pipeline Supply Quantities.

Compute Inherent Availability.

SECTION 9 - LOGAM FLOWCHART



AYZP<0, All Initial Provisions are Predetermined.

Compute Initial Provisions Using LOGAM Supply Rules.

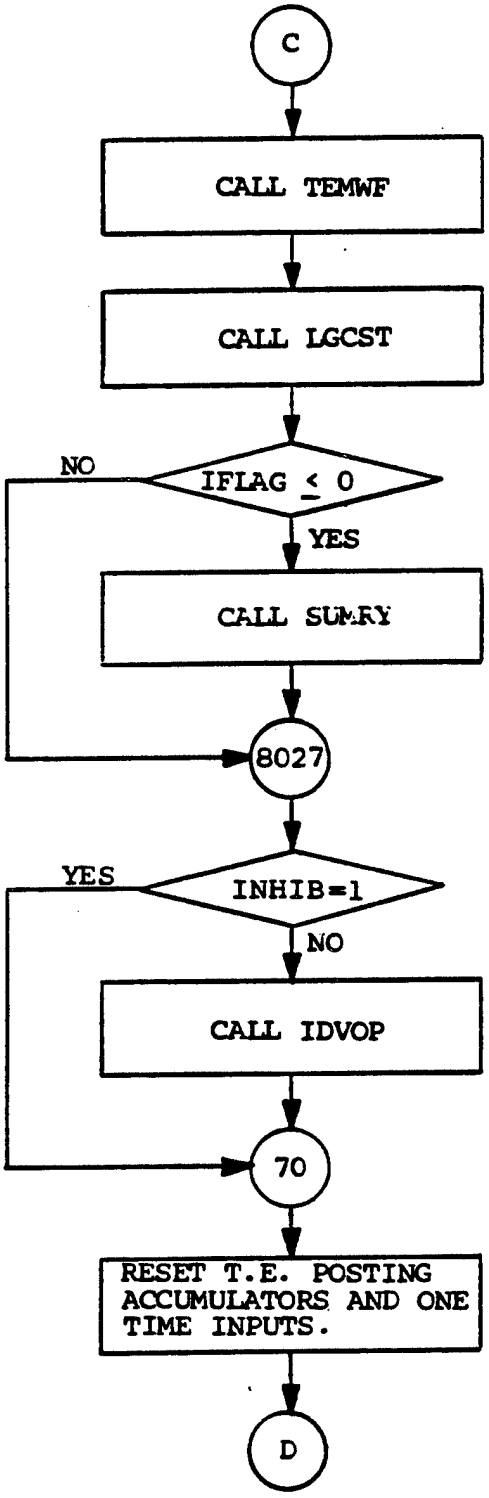
Compute Initial Provisions Using LOGAM Maintenance Rules.

Replace Computed Provisions with Predetermined Value.

Compute Operational Availability and Modify Stock if Over-Ride Factor Used in AYZP.

Distribute Initial Provisions Equally Over the Stock Points at a Supply Echelon.

Compute Replenishement Spares.

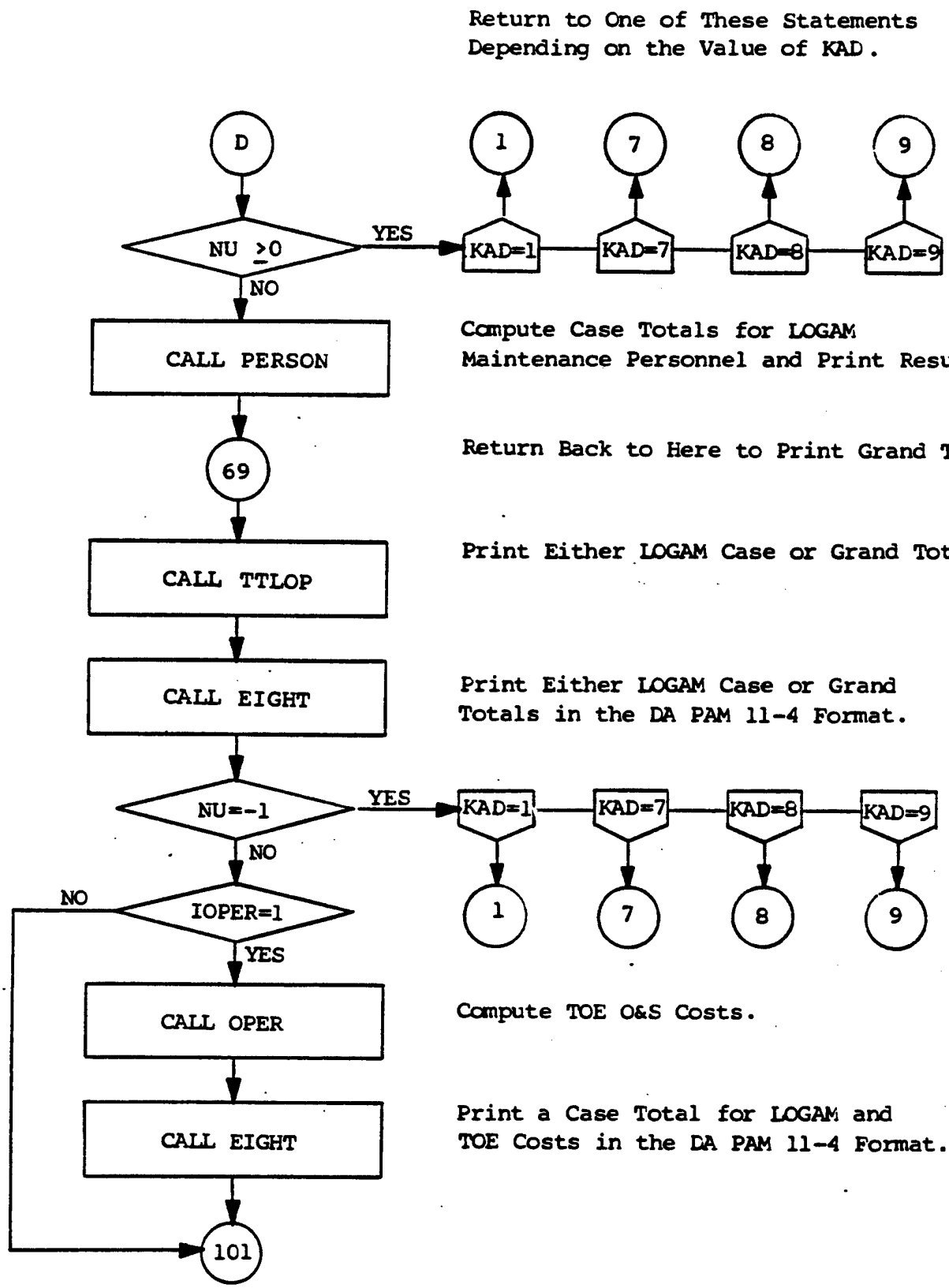


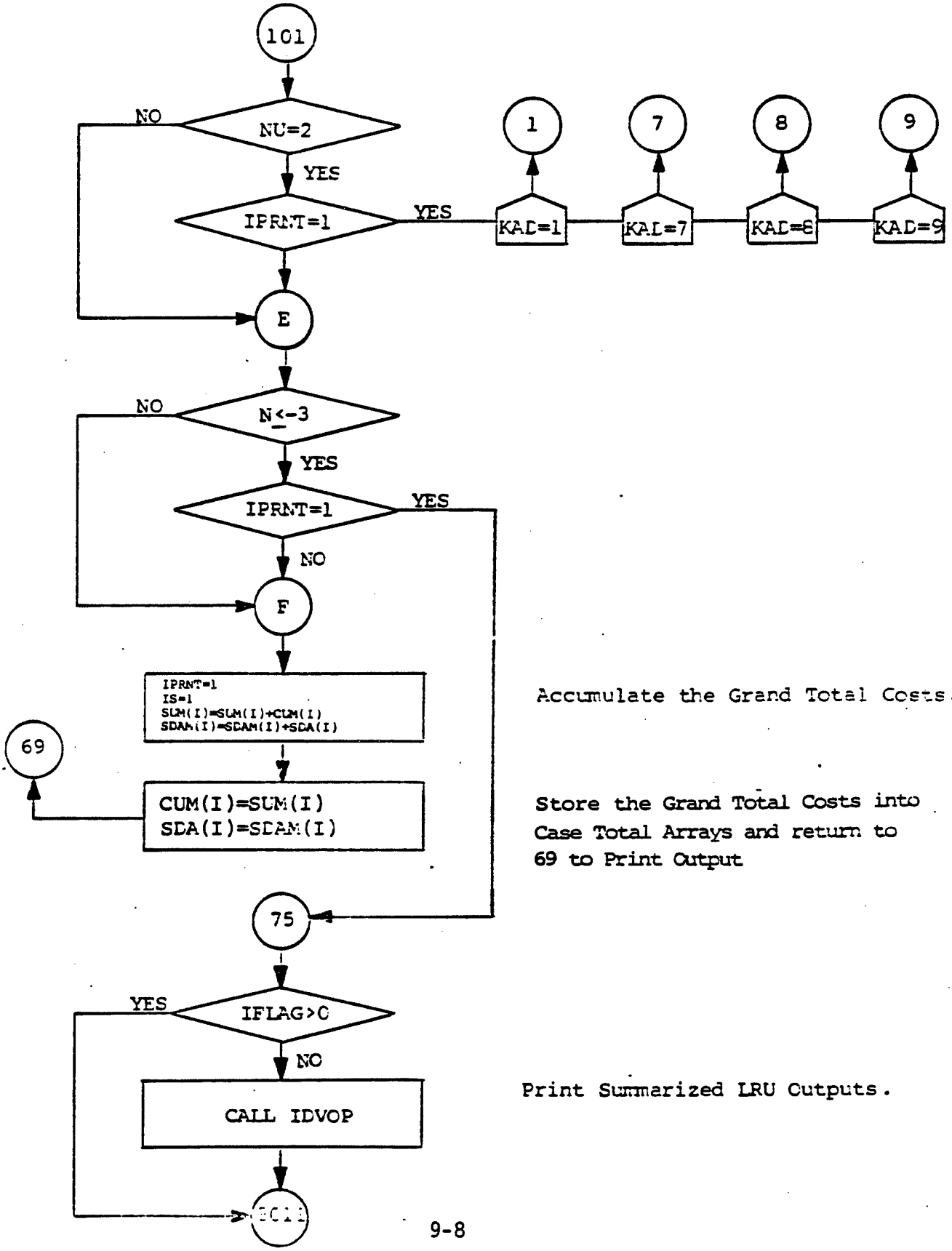
Compute Test Equipment and Manpower Demands.

Compute LOGAM System Costs.

Summarize Distinct LRU Cases.

Print Outputs from an Individual LRU Case.

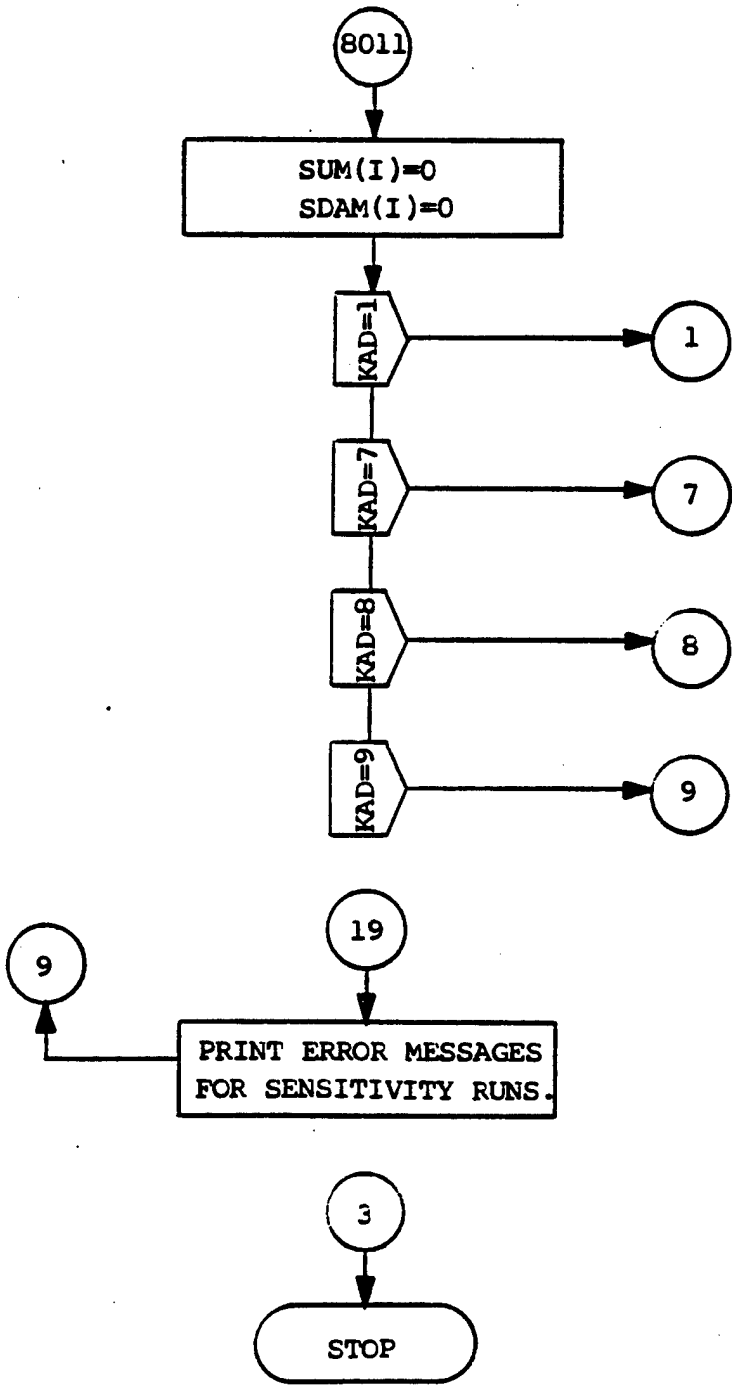

Return to One of These Statements Depending on the Value of KAD.
D
1
7
8
9
NU ≥0
YES
KAD=1
KAD=7
KAD=8
KAD=9
NO
CALL PERSON
Compute Case Totals for LOGAM Maintenance Personnel and Print Results.
69
Return Back to Here to Print Grand Total
CALL TTLOP
Print Either LOGAM Case or Grand Totals
CALL EIGHT
Print Either LOGAM Case or Grand Totals in the DA PAM 11-4 Format.
NU=-1
YES
KAD=1
KAD=7
KAD=8
KAD=9
NO
1
7
8
9
NO
IOPER=1
YES
CALL OPER
Compute TOE O&S Costs.
CALL EIGHT
Print a Case Total for LOGAM and TOE Costs in the DA PAM 11-4 Format.
101



Accumulate the Grand Total Costs.

Store the Grand Total Costs into Case Total Arrays and return to 69 to Print Output

Print Summarized LRU Outputs.



Reinitialize the Grand Total Accumulators.

Return After for a Baseline LRU Case.

Return to Sensitivity Section for Next Pass Through LRU Cases.

Return to Sensitivity Section - for Next LRU Case of a Pass.

Return After Completion of all Sensitivity Passes.

This Statement is Entered if an End-of-File is Read on One of the Formatted Input Cards.

DISTRIBUTION

COPIES

1 Deputy Under Sec'y of the Army, ATTN: Office of Op Research
1 Asst Sec'y of the Army (I,L&FM), Pentagon, Wash., DC 20310
Headquarters, US Army Materiel Development & Readiness Command
1 DRCDM-S
1 DRCDMR
1 DRCDRM
1 DRCMM
1 DRCMM-R
1 DRCMM-S
1 DRCMM-M
1 DRSAC
1 DRCSM
1 DRCSM-E
1 DRCSM-W
1 DRCSM-WR
1 DRCSM-WA
1 DRCSM-WG
1 DRCSM-WC
1 DRCSM-I
1 DRCSM-P
1 DRCSM-PI
1 DRCSM-PS
1 DRCSM-PM
1 DRCSM-PP
1 DRCSM-PAS
1 DRCRE
1 DRCRE-C
1 DRCPP-E
1 DRCRE-I
1 DRCDM-S
1 DRCDP
1 DRCMS
1 DRCQA
1 DRCDE
Dep Chf of Staff for Logistics, Pentagon, Wash., DC 20310
1 DALO-SMP-U
1 DALO-SMZ-B
1 DALO-PLZ-X
1 DALO-SML
1 Ofc, Asst Sec'y of Defense, ATTN: MRA&L-SR, Pentagon, Wash., DC 20310
1 Commandant, US Army Logistics Management Center, Ft. Lee, VA 23801
1 Defense Logistics Studies Info Exchange, DRXMD-D, ALMC, Ft. Lee, VA 2380
2 Defense Technical Info Center, Cameron Station, Alexandria, VA 22314
Commander, USA Armament Materiel Readiness Cmd, Rock Island, IL 61299
1 DRSAR-MM
1 DRSAR-SA

Commander, USA Communications-Electronics Cmd, Ft. Monmouth, NJ 07703
1 DRSEL-MM
1 DRSEL-PL-SA
Commander, USA Troop Support & Aviation Materiel Readiness Command, St. Louis, MO 63120
1 DRSTS-SPF
1 DRSTS-SPS
1 DRSTS-BA
Commander, US Army Tank-Automotive Command, Warren, MI 48090
1 ATTN: DRSTA-F
1 ATTN: DRSTA-S
1 Commander, USA Armament R&D Cmd, ATTN: DRDAR-SE, Dover, NJ 07801
1 Commander, USA Aviation R&D Cmd, 4300 Goodfellow Blvd, St. Louis, MO 63120
1 Commander, USA Electronics R&D Cmd, ATTN: DRDEL-ST-SA, Rm 2D315, Bldg 2700, Ft Monmouth, NJ 07703
1 Commander, USA Mobility Equipment R&D Cmd, ATTN: DRDME-O, Ft. Belvoir, VA 22060
1 Commander, USA Natick R&D Cmd, ATTN: DRXNM-O, Natick, MA 01760
Director, Army Materiel Systems Analysis Activity, APG, MD 21005
1 DRXSY-D
1 DRXSY-G
1 DRXSY-A
1 DRXSY-C
1 DRXSY-R
1 DRXSY-RM
1 DRXSY-F
1 DRXSY-FX
1 DRXSY-FA
1 DRXSM-FM
1 DRXSY-FR
1 DRXSY-FO
1 DRXSY-DA
1 Commander, US Army Logistics Center, Ft. Lee, VA 23801
1 Commander, US Army Logistics Evaluation Agency, New Cumberland Army Depot, New Cumberland, PA 17070
1 Commander, US Army Depot Systems Command, Chambersburg, PA 17201
1 Commander, US Air Force Logistics Cmd, WPAFB, ATTN: AFLC/XRS, Dayton, Ohio 45433
1 US Navy Fleet Materiel Support Office, Naval Support Depot, Mechanicsburg, PA 17055
1 Mr. James Prichard, Navy SEA Systems Cmd, ATTN: PMS 3061, Dept of US Navy, Wash., DC 20362
1 George Washington University, Inst. of Management Science & Engr., 707 22nd St., N.M., Wash., DC 20006
1 Naval Postgraduate School, ATTN: Dept of Opns Anal, Monterey, CA 93940
1 Air Force Institute of Technology, ATTN: SLGQ Head Quantitative Studies Dept., Dayton. OH 43433
1 US Army Military Academy, West Point, NY 10996

1 Commander, US Army Logistics Center, ATTN: Concepts & Doctrine Directorate, Ft. Lee, VA 23801
1 Scientific Advisor, ATCL-SCA, Army Logistics Center, Ft. Lee, VA 23801
1 HQ, Dept of the Army, (DASG-HCL=P), Wash., DC 20314
1 Librarian, Logistics Mgt Inst., 4701 Sanquamore Rd., Wash., DC 20016
1 Wallace M. Cohen, Asst Director Systems Analysis, FGMSD, General Accounting Ofc, Wash., DC 20548
1 HQ, Dept of the Army, (DASG-HCL-P), Wash, DC 20314
1 Commander, USDRC Automated Logistics Mgt Systems Activity. P.O. Box 1578, St. Louis, MO 63188
1 Director, DARCOM Logistics Systems Support Agency, Letterkenny Army Depot, Chanbersburg, PA 17201
1 Commander, Materiel Readiness Support Activity, Lexington, KY 40507
1 Director, Army Management Engineering Training Agency, Rock Island Arsenal, Rock Island, IL 61299
1 Defense Logistics Agcy, ATTN: DLA-LO, Cameron Sta, Alexandria, VA 22314
1 Dep Chf of Staff (I&L), HQ USMC-LMP-2, ATTN: LTC Sonneborn, Jr., Wash., DC 20380
1 Commander, US Army Depot Systems Command, Letterkenny Army Depot, ATTN: DRSDS-LL, Chambersburg, PA 17201
1 Logistics Control Activity, Presidio of San Francisco, CA 94120
1 Churck B. Foster, Communications Security Logistics Activity, DRSEL-CCM-NMP, Ft Huachuca, AX 85613
1 Logistics Studies Office, DRXSY-FLSO, Ft. Lee, VA 23801
1 Commander, US Army Communications Command, ATTN: Dr. Forry, CC-LOG-LEO, Ft. Huachuca, AX 85613
1 Commander, US Army Test & Evaluation Cmd, ATTN: DRSTE-SY, Aberdeen Proving Ground, MD 21005
1 DARCOM Intern Training Center, Red River Army Depot, Texarkana, TX 75501
1 US Army Training & Doctrine Command, Ft. Monroe, VA 23651
1 Operations & Inventory Analysis Office, NAVSUP (Code 04A) Dept of Navy, Wash., DC 20376
1 US Army Research Office, ATTN: Robert Launer, Mat. Div., P.O. Box 12211, Research Triangle Park, NC 27709
Air Force Logistics Mgt Center, Gunter Air Force Station, AL 36144 AFLMC/LGY
1 Engineer Studies Center, 6500 Brooks Lane, Wash., DC 20315
1 Dr. Carl Weisman, C.A.C.I., 1815 N. Fort Myer Drive, Arlington, VA 22209