

RESEARCH NOTE 80-15

LEVEL II



AD A092208

**PROGRAMMER'S MANUAL FOR THE
TACTICAL OPERATIONS SYSTEM
ANALYSIS PACKAGE**

Gary Witus, Mark Meerschaert, Donald Kleist

VECTOR RESEARCH, INCORPORATED
P.O. Box 1506
Ann Arbor, Michigan 48106

Human Factors Technical Area



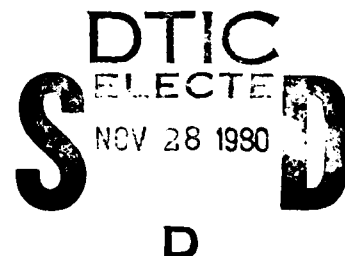
U. S. Army

Research Institute for the Behavioral and Social Sciences

May 1980

Contract DAHC19-78-C-0027

Approved for public release; distribution unlimited.



DDC FILE COPY

80 11 25 043

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM															
1. REPORT NUMBER (18) ARI Research Notes 80-15/	2. GOVT ACCESSION NO. AD-A092 208	3. RECIPIENT'S CATALOG NUMBER (19) RN-1															
4. TITLE (and Subtitle) (6) Programmer's Manual for the Tactical Operations System Analysis Package.		5. TYPE OF REPORT & PERIOD COVERED (7) Interim May 79-May 80															
7. AUTHOR(s) (10) Gary/Witus, Mark/Meerschaert, Donald/Kleist		6. PERFORMING ORG. REPORT NUMBER (14) VRI-ARI-3-FR80-4															
9. PERFORMING ORGANIZATION NAME AND ADDRESS Vector Research, Incorporated, PO Box 1506 Ann Arbor, Michigan 48106		8. CONTRACT OR GRANT NUMBER(s) (15) DAHC19-78-C-0027															
11. CONTROLLING OFFICE NAME AND ADDRESS CORADCOM Field Office ATTN: DRDCO-FL Ft. Leavenworth, Kansas 66027		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) 2Q363739A793 (12) 1A Task 1A Workunit 001															
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) US Army Research Institute for the Behavioral and Social Sciences ATTN: PERI-OS 5001 Eisenhower Ave Alexandria VA 22333		12. REPORT DATE 25 February 1980															
		13. NUMBER OF PAGES 175															
		15. SECURITY CLASS. (of this report) Unclassified															
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE															
16. DISTRIBUTION STATEMENT (of this Report) Approved for open release; distribution unlimited.																	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																	
18. SUPPLEMENTARY NOTES This research was technically monitored by Mr. Robert S. Andrews and Dr. Stanley M. Halpin of ARI and LTC Larry Walker of CORADCOM.																	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0"> <tr> <td>Command and Control</td> <td>Information Management</td> <td>Standing Operating</td> </tr> <tr> <td>Communications</td> <td>Information Summarization</td> <td>Procedures (SOP)</td> </tr> <tr> <td>Computers</td> <td>Mathematical Modeling</td> <td>Tactical Operations</td> </tr> <tr> <td>Data base Management</td> <td>Networks</td> <td>System (TOS)</td> </tr> <tr> <td>Design aid</td> <td>Queueing</td> <td></td> </tr> </table>			Command and Control	Information Management	Standing Operating	Communications	Information Summarization	Procedures (SOP)	Computers	Mathematical Modeling	Tactical Operations	Data base Management	Networks	System (TOS)	Design aid	Queueing	
Command and Control	Information Management	Standing Operating															
Communications	Information Summarization	Procedures (SOP)															
Computers	Mathematical Modeling	Tactical Operations															
Data base Management	Networks	System (TOS)															
Design aid	Queueing																
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This document discusses the programming details of a computer program package for the analysis of the Tactical Operations System (TOS). The program computer code is included. These programs were produced during the second phase of a project to develop information management concepts and procedures for automated battlefield command and control systems.																	

PREFACE

This document is one of eight reports which describe the work performed by Vector Research, Incorporated (VRI) and its subcontractor, Perceptronics, Incorporated, for the US Army Research Institute for the Behavioral and Social Sciences (ARI) under the second phase of contract number DAHC19-78-C-0027. The work described was performed over 12 months of an anticipated 36-month three-phased project. The overall objective of the project has been to produce procedural guidelines to be used by divisions in the field in developing standard operating procedures for information management in the Tactical Operations System (TOS). As a consequence of the redirection of the TOS development effort in November 1979, the objective of this work was reinterpreted to include automated battlefield command control systems (ABCCS) in general, using TOS for an explicit example of the design, human factors, and management control considerations which must be addressed.

The VRI study team for phase II was comprised of Dr. Robert W. Blum (Project Leader), Ms. Cathleen A. Callahan, Dr. W. Peter Cherry, Mr. Mark G. Graulich, Mr. Donald Kleist, Mr. Mark Meerschaert, Mr. Gregory Touma, and Mr. Gary Witus. The Perceptronics team for phase II consisted of Dr. Michael G. Samet and Dr. Ralph E. Geiselman.

The authors wish to acknowledge the helpful contributions of Dr. Stanley M. Halpin and Mr. Robert Andrews, who were charged with monitoring the study for ARI; and LTC L. Walker, MAJ. A. Edmonds, and Mr. M. Carrio, who performed a similar function for that portion of the study effort which was jointly sponsored with ARI by the US Army Communications Research and Development Command (CORADCOM).

The eight reports are as follows:

Blum et al. Information Management for an Automated Battlefield Command and Control System: Executive Summary. ARI Research Report 1249. -- presents an overview of the project and the other seven reports.

Callahan et al. Guidelines for Managing the Flow of Information in an Automated Battlefield Command and Control System. ARI Research Report 1348. -- describes considerations in and procedures for the management of contemporary ABCC systems.

Geiselman and Samet. Guideline Development for Summarization of Tactical Data. ARI Technical Report 458. -- an analysis of procedures for the extraction, summarization, and presentation of critical information.

Witus et al. Analysis of Information Flow in the Tactical Operations System (TOS). ARI Research Notes 80-12. -- describes the purpose, approach, and results of a TOS analysis which focused on TOS when integrated with a planned communications support system.

Witus et al. Description of the Tactical Operations System Information Flow Model. ARI Research Notes 80-13. -- describes the representation of TOS used to develop the analysis package and the mathematics of the model.

Witus et al. User's Manual for the Tactical Operations System Analysis Package. ARI Research Notes 80-14. -- explains the use and operation of the analysis package.

Witus et al. Programmer's Manual for the Tactical Operations System Analysis Package. ARI Research Notes 80-15. -- describes the programming details of the package to facilitate modifications or transfer between host systems.

Cherry, W. All Source Analysis System: Design Issues. ARI Working Paper HF80-XX. -- a discussion of design issues associated with the emerging ASAS concept.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Special
A	

CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	1-1
2.0 PROGRAM CREATE	2-1
2.1 Variables in Program Create	2-1
2.2 Basic Flow of Program Create	2-13
2.3 Subroutines in Create	2-17
2.4 Data File	2-19
2.5 Code of Program Create	2-22
3.0 PROGRAM MODIFY	3-1
3.1 Structure of Program MODIFY	3-1
3.2 Code of Program Modify	3-3
3.3 Flow Charts of the Program Modify and Selected Subroutines	3-5
4.0 PROGRAM DISPLAY	4-1
5.0 PROGRAM COMPUTE	5-1
APPENDIX A	A-1
APPENDIX B	B-1
APPENDIX C	C-1

EXHIBITS

<u>Number</u>	<u>Title</u>	<u>Page</u>
1-1	Analysis Package Organization	1-2
2-1	Configuration Variables in CREATE	2-2
2-2	Communications System Variables in CREATE	2-5
2-3	Message Variables in CREATE	2-6
2-4	Route Cross Message Array Variables in CREATE	2-7
2-5	Engineering Data in CREATE	2-8
2-6	Environmental Variables in CREATE	2-11
2-7	Logic Variables in CREATE	2-12
2-8	Basic Flow of CREATE	2-14
2-9	Subroutines in CREATE	2-18
2-10	Output File Format	2-20
3-1	Structure of Program MODIFY	3-2
3-2	Data Modification Subroutines Used By Program MODIFY	3-4
3-3	Flowchart for Program MODIFY	3-6
3-4	Flowchart of Subroutine COMSYS	3-11
3-5	Flowchart of Subroutine LDFACT	3-14
3-6	Flowchart of Subroutine NOISE	3-15
5-1	Organization of COMPUTE	5-2
5-2	Flowchart for the Input and Initialization Module	5-3
5-3	Flowchart for the Traffic Flow Module	5-4
5-4	Flowchart for the Operating Statistics Module	5-5

EXHIBITS

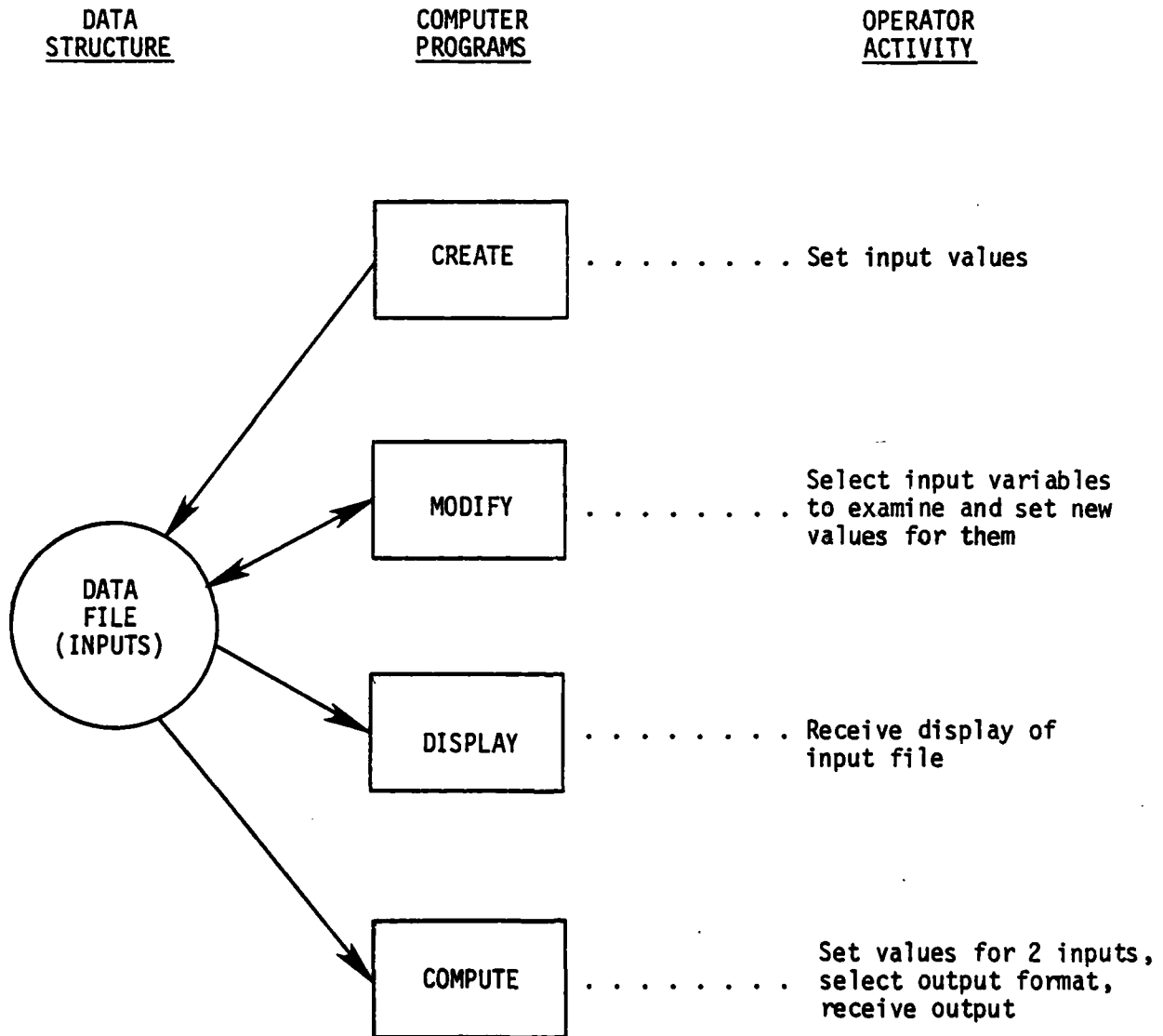
<u>Number</u>	<u>Title</u>	<u>Page</u>
5-5	Flowchart for Segment A of the Operating Statistics Module	5-6
5-6	Flowchart for Segment B of Operating Statistics Module	5-7
5-7	Flowchart for Subroutine TRANS	5-8
5-8	Flowchart for the Performance Measure Module	5-10
5-9	Flowchart for the Output Module	5-11
5-10	Variables in COMPUTE	5-12
5-11	Logical I/O Unit Assignments	5-18

1.0 INTRODUCTION

↓
The purpose of this volume is to document the computer programs and data structures in the TOS analysis package in sufficient detail to enable a programmer to modify the package or transfer it between computer systems. The volume is organized into five chapters and three appendices. The remainder of this chapter discusses the overall organization of the analysis package. Chapters 2.0 through 5.0 discuss, in turn, each of the four computer programs in the package and present flowcharts, descriptions of the variables and subroutines, and discussions of the code. Appendix A contains a listing of the four programs. Appendix B contains a variable name cross-reference dictionary. Appendix C contains a statement label dictionary. All programs are written in standard FORTRAN IV and contain internal commentary. They are currently implemented on the Amdahl 470/V7 computer at The University of Michigan.

The analysis package performs three basic functions: (1) interacting with the operator to set, alter, or examine the values of the inputs; (2) performing the computations specified by the mathematics of the model; and (3) displaying the outputs. The package consists of four computer programs and one external data structure. The organization of the package, the interactions between the programs and the data structure, and the operator activities are shown in exhibit 1-1. Program CREATE prompts the operator to specify values for all of the inputs in the data file and uses them to create a new data file. Program MODIFY reads an existing data file, and allows the user to select data elements

EXHIBIT 1-1: ANALYSIS PACKAGE ORGANIZATION



for display or modification. The program then creates a new data file. Program DISPLAY reads, formats, and displays an existing data file. Program COMPUTE performs all of the computations required by the model, allows the operator to select the output format, and displays the outputs.

2.0 PROGRAM CREATE

Program CREATE was written to facilitate construction of the data file which contains input data for the TOS simulation. Program CREATE is written in FORTRAN IV and is set up to be run interactively from a terminal. The programming code and logic are straightforward and should facilitate implementation on any computer the potential user desires. The remainder of this section discusses the variables and characteristics of program CREATE.

2.1 VARIABLES IN PROGRAM CREATE

Exhibits 2-1 through 2-7 contain all of the variables used in program CREATE, indicate the type of variable as implemented in the program, and give a definition of the meaning of each variables. The contents of these exhibits are:

- Exhibit 2-1: Network Configuration Data
- Exhibit 2-2: Communications System Data
- Exhibit 2-3: Message Types Data
- Exhibit 2-4: Route Cross Message Data
- Exhibit 2-5: Engineering Data
- Exhibit 2-6: Error Rate Data
- Exhibit 2-7: Program Execution Controls

The variables which contain information about the TOS system have names mnemonically derived so that understanding of the meaning of the variables is facilitated. The exceptions to this general rule are the logic variables which are used to control the program flow.

EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NBDE	I*4	The number of brigades
BDNAME(I)	R*8	The name of Brigade I
NBN(I)	I*4	Number of battalions in Brigade I
BD(I)	I*4	Type of processor at Brigade I: = 1 if TCS, = 2 if TCT
NLINC(I)	R*8	The name of the channel connecting Brigade I with the FEP
BNNAME(I,J)	R*8	The name of Battalion J of Brigade I
BN(I,J)	I*4	The type of processor at Battalion J of Brigade I: = 1 if TCS, = 2 if TCT
NLINK(I,J)	R*8	The name of the channel connecting Battalion J to Brigade I
NOU	I*4	The number of Other TOS Users (All users which are not either brigades or their subordinate battalions are considered Other TOS users.)
OUNAME(I)	R*8	The name of Other TOS User I
OU(I)	I*4	Type of processor at Other TOS User I: = 1 if TCS, = 2 if TCT
NLINCK(I)	R*8	Name of the channel connecting Other TOS User I to the FEP
NC	I*4	The number of communication channels in the TOS network
CNAM(I)	R*8	The name of TOS communication channel I

-- Continued --

EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE
(Continued)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NR	I*4	The number of routes in the TOS communication network: equals the number of Other TOS users plus number of brigades plus number of battalions
RP(I,1)	I*4	The number of the route. Routes are numbered in the same order as given for the route name. (The following order of routes is followed throughout the TOS analysis package: routes numbered 1 through NOU are the Other TOS Users; routes NOU+1 to NOU+NBDE are Brigades 1 through NBDE, next come the battalions, first the battalions of Brigade 1 then the battalions of 2, etc.)
RNAM(I)	R*8	The name of route I.
RP(I,2)	I*4	The node with which node I communicates. By convention this equals 0 (the FEP) for Other TOS Users and Brigades. For a Battalion, it equals the route number of its parent brigade
NL	I*4	Number of links, equals the number of routes
LP(I,1)	I*4	Equal to RP(I,1), the number of the link
LP(I,2)	I*4	Number of channel connecting node I with the FEP, if I is a brigade or Other TOS User, or with the parent brigade if I is a battalion
LP(I,3)	I*4	The number of the node with which node I communicates. If node I communicates with the FEP, LP(I,3) equals NR plus 1, if node I is a battalion, LP(I,3) is the number of its parent brigade

-- Continued --

EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE
(Concluded)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
LINC(I)	I*4	The number of the channel connecting Brigade I with the FEP
LINK(I,J)	I*4	The number of the channel connecting Battalion J with Brigade I
		LINCK(I) I*4 The number of the channel connecting Other TOS User I with the FEP
IPD(I)	I*4	The type of processor at node I: = 1 if TCS, = 2 if TCT
NP	I*4	Number of processors: equal NR+1
NN	I*4	Number of nodes: equals NP + NC + 3

EXHIBIT 2-2: COMMUNICATIONS SYSTEM VARIABLES
IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
CD(I,1)	R*4	Rise time, in milliseconds, of Channel I
CD(I,2)	R*4	Transmission rate, in characters per second, of Channel I
CD(I,3)	R*4	Proportion of time Channel I is being used for voice communications
CD(I,4)	R*4	Number of lines on Channel I

EXHIBIT 2-3: MESSAGE VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NM	I*4	Total number of message types in the system
NUM	I*4	Number of user-initiated messages in the TOS system (By implication, the number of TOS-generated messages is NM - NUM)
MNAM(I)	R*8	The name of Message Type I. (By convention user-initiated messages are numbered 1 through NUM and TOS-initiated messages are numbered NUM + 1 through NM)
MD(I,1)	R*4	The average length in characters of Message Type I
IMP(I,1)	R*8	The name of message type caused as output on the same route by Message Type I: = 0, if none
IMP(I,2)	R*8	The name of message type caused as output on another route by Message Type I: = 0, if none or if Message I is a TOS-generated message
MP(I,1)	I*4	The number of Message Type IMP(I,1)
MP(I,2)	I*4	The number of Message Type IMP(I,2)

EXHIBIT 2-4: ROUTE CROSS MESSAGE ARRAY VARIABLES IN CREATE
(I = Route Number, J = Message Number)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
(If $J \leq \text{NUM}$, Message J is a user-initiated message)		
RM(I,J,1)	R*4	Initiation rate, in messages per hour, of Message J at Node I
RM(I,J,2)	R*4	= 0, not used for user-initiated messages
RM(I,J,3)	R*4	= 0 if I is not a battalion. Equals the probability that Message J is deleted during hierarchical review at the parent brigade if I is a battalion
RM(I,J,4)	R*4	= 0 if I is not a battalion. Equals the probability that Message J is altered, but not deleted, during hierarchical review at the parent brigade if I is a battalion
(If $J > \text{NUM}$, Message J is a TOS-initiated message)		
RM(I,J,1)	R*4	= $\alpha(I,J)$, proportionality constant used to determine the initiation rate of TOS-initiated Message J on route I
RM(I,J,2)	R*4	= $\beta(I,J)$, proportionality constant used to which determine the initiation rate of TOS-initiated Message J on route I
RM(I,J,3)	R*4	= 0, not used for TOS-initiated messages
RM(I,J,4)	R*4	= 0, not used for TOS-initiated messages

EXHIBIT 2-5: ENGINEERING DATA IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
DD(1)	R*4	Message Disk Mean access time in milliseconds
DD(2)	R*4	Data Disk Mean access time in milliseconds
GPD(1,1)	R*4	CPU, in milliseconds per message, required to originate a message, at a TCS
GPD(1,2)	R*4	CPU, in milliseconds per message, required to originate a message at a TCS
GPD(2,1)	R*4	CPU, time in milliseconds per message, required to send the message, at a TCS
GPD(2,2)	R*4	CPU, time in milliseconds per message, required to send the message at a TCS
GPD(3,1)	R*4	CPU, time in milliseconds per message, required to receive a message at a TCS
GPD(3,2)	R*4	CPU, time in milliseconds per message, required to receive a message, at a TCS
GPD(4,1)	R*4	CPU, time in milliseconds per message, to template check a message, at a TCS
GPD(4,2)	R*4	CPU, time in milliseconds per message, required to template a check a message, at a TCS
GPD(5,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(5,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS

-- Continued --

EXHIBIT 2-5: ENGINEERING DATA IN CREATE
(Continued)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
GPD(6,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(6,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(7,1)	R*4	CPU, time in milliseconds per message, required to originate a message, at a TCS
GPD(7,2)	R*4	CPU, time in milliseconds per message, required to originate a message, at a TCS
GPD(8,1)	R*4	CPU, time in milliseconds per message, required to send a message, at a TCS
GPD(8,2)	R*4	CPU, time in milliseconds per message, required to send a message, at a TCS
GPD(9,1)	R*4	CPU, time in milliseconds per message, required to send to receive a message, per message
GPD(9,2)	R*4	CPU, time in milliseconds per message, required to receive a message, at a TCS
GPD(10,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(10,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(11,1)	R*4	CPU, time in milliseconds per message, required to initiate a message, at a TCS
GPD(11,2)	R*4	CPU, time in milliseconds per message, required to initiate a message at a TCS

-- Continued --

EXHIBIT 2-5: ENGINEERING DATA IN CREATE
(Concluded)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
MD(I,2)	R*4	Data base processing time, in milliseconds, of Message Type I
MD(I,3)	R*4	Number of message disk reads or writes for Message Type I
MD(I,4)	R*4	Number of data disks reads or writes for Message Type I
PD(I)	I*4	Number of templates at node I
MD(I,4)	R*4	Number of data disks reads or writes for Message Type I
PD(I)	I*4	Number if templates at node I: = 0 if node I is not a Brigade
PD(NR+1)	I*4	Number of templates at the FEP

EXHIBIT 2-6: ENVIRONMENTAL VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
LD(I,1)	R*4	The bit error rate for a message sent from Node I to the next node, in bits in error per 1000 bits
LD(I,2)	R*4	The bit error rate for a message sent to Node I, in bits in error per 1000 bits

EXHIBIT 2-7: LOGIC VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
IV	I*4	= 1 if user desires short prompts
ITCT	I*4	= 1 if a TCT is present in the TOS system
ITCT	I*4	= 2 if a TCT is present in the TOS system
IST	I*4	A temporary variable to determine if TCSs and TCTs are present
TCS	I*4	Literally, the character string "TCS"
TCT	I*4	Literally, the character string "TCT"
IFG	I*4	= 1 if the Route cross Message array is to be read in later
IW	I*4	= 1 if the user desires to call the display routine to print the data on the terminal
IWF	I*4	= 1 if the user desires to create an output file

(I, II, III, IJ, I1, I2, J, JI, JJ, JJJ, JK, JN, JR, K, RNUM are all indices used in the CREATE logic)

A convention for numbering the users of the TOS system is followed throughout all four of the computer programs. In this system, the first users are always the non-brigade users of the TOS system. Immediately following these are all of the brigade users. Following the brigades are their battalion users: all the battalions of the first brigade, then all the battalions of the second brigade, etc. In this way, the type of TOS user can be inferred by its number without having to append additional identifiers to the number or names. The use of this convention will become obvious when the data file is discussed.

The arrays BDNAM, BNNAME, OUNAM, CNAM, RNAM, and MNAM are declared as R*8 variables in program CREATE. They are, however, used as names in CREATE and may contain strings of up to eight alpha-numeric characters. They are included as a convenience to the user and are not used in the logic of any of the programs in the TOS analysis package.

It should be noted that some of the variables in CREATE are redundant. For example, the LP and RP arrays contain some of the same information. This organization facilitated the coding of program COMPUTE.

2.2 BASIC FLOW OF PROGRAM CREATE

Exhibit 2-8 contains a basic flowchart of program CREATE. The steps in CREATE are implemented in a straightforward serial fashion. They are set up so that succeeding steps use information which are input in earlier steps. This information is used to control the internal DO Loops so they provide the correct number of prompts.

EXHIBIT 2-8: BASIC FLOW OF CREATE

- Step 0: Set IV for long or short prompts IV = 1 short
- Step 1: Brigade Information
 Enter number of Brigades: NBDE
 For each Brigade, enter name, type of processor, and
 number of Battalions: BDNAM(I), BD(I), NBN(I)
 Call CHECK for BD(I) to check name
- Step 2: Battalion Information
 For each Battalion, enter name and type of processor
 BNNAME(I,J), BN(I,J) ((I,J) = Battalion J of Brigade I)
 Call CHECK for BN(I,J) to check name
- Step 3: Other TOS User information
 Enter number of Other TOS Users: NOU
 For each Other TOS User, enter name and type of processor
 OUNAM(I), OU(I)
 Call CHECK for OU(I) to check name
- Step 4: Channel Data
 Enter number of communications channels: NC
 For each channel, enter the name CNAM(I)
- Step 5: Connect User with Channels
 For each Battalion, enter name of channel connecting it to its
 parent Brigade, NLINK(I,J)
 Call CHECK 1 to check name and assign channel number to
 LINK(I,J)
 For each Brigade enter name of channel connecting it to the
 FEP: NLINC(I)
 Call CHECK1 to check name and assign channel number to
 LINC(I)
 For each Other TOS User, enter name of channel connecting it to
 the FEP: NLINCK(I)
 Call CHECK1 to check name and assign channel number to
 LINCK(I)
- Step 6: Make Configuration Assignments
 Compute NL, NR, NP, NN
 For each TOS user, assign values to RNAM(I), RP(I,1), RP(I,2),
 LP(I,1), LP(I,2), LP(I,3), IPD(I)
 Check to determine if configuration includes both TCSs and TCTs

-- Continued

EXHIBIT 2-8: BASIC FLOW OF CREATE (Continued)

Step 7: Message Data
 Enter number of messages and number of user initiated messages, NM, NUM
 For each user-initiated message enter name: MNAM(I)
 For each TOS-initiated message enter name: MNAM(I)
 For each user-initiated message enter mean length, name of message caused as output on same route, and name of message caused as output on other routes: MD(I,1), IMP(I,1), IMP(I,2)
 Call CHECK2 to check message names and assign message numbers to MP(I,1) and MP(I,2)
 For each TOS-initiated message, enter mean length: MD(I,1)
 Assign MP(I,1) = MP(I,2) = 0

Step 8: Route Cross Message Array
 (If Array to be inserted into file later, set IFG=1 and bypass step 8)
 For each Other TOS User, then for each Brigade, enter for each user-initiated message the message initiation rate: RM(I,J,1). Then assign RM(I,J,2) = RM(I,J,3) = RM(I,J,4) = 0
 For each TOS-initiated message enter proportionality constants $\alpha(I,J)$ and $\beta(I,J)$, RM(I,J,1) and RM(I,J,2). Then assign RM(I,J,3) = RM(I,J,4) = 0
 For each Battalion, for each user-initiated message, enter the message initiation rate, the probability that the message is deleted at Brigade review and the probability the message is altered at review: RM(I,J,1), RM(I,J,3), and RM(I,J,4).
 Call CHECK 3 to insure probabilities are in the interval 0-1; then assign RM(I,J,2) = 0
 For each TOS initiated message, enter proportionality constants $\alpha(I,J)$ and $\beta(I,J)$, and RM(I,J,2); then assign RM(I,J,3) = RM(I,J,4) = 0

Step 9: Engineering Data
 Enter data disk and message disk access times: DD(1), and DD(2)
 Enter TCS data: GPD(1...5, 1 and 2)
 Enter TCT data: GPD(7...10, 1 and 2)
 Enter FEP data: GPD(11, 1 and 2), and GPD(6, 1 and 2)
 For each channel, enter rise time, transmission rate, and number of lines: CD(I,1), CD(I,2), and CD(I,4)
 For each message, enter data base processing time, number of message disk reads or writes, and number of data disk reads or writes: MD(I,2), MD(I,3) and MD(I,4)
 For each Brigade enter number of templates: PD(I)
 Enter number of templates at FEP: PD(NR+1)
 Assign PD(I) = 0 for all nodes not a Brigade or the FEP

Step 10: Voice Competition Data
 For each channel, enter proportion of time channel is used for voice transmission: CD(I,3)
 Call CHECK4 to insure CD(I,3) is in the range 0-1

-- Continued --

EXHIBIT 2-8: BASIC FLOW OF CREATE (Concluded)

- STEP 11: Error Rate Data
 For each Other TOS User, then for each Brigade:
 enter the bit error rate for a message from the user to the
 the FEP and the bit error rate for a message from the FEP to
 the user: LD(I,1), and LD(I,2)
 Call CHECK 3 to ensure LD(I,1) and LD(I,2) are in the
 interval 0-1
 For each Battalion, enter the bit error rate for a message sent
 to the parent Brigade and the bit error rate for a message to
 the Battalion.
 Call CHECK 3 to ensure LD(I,1) and LD(I,2) are in the
 interval 0-1.
- Step 12: Display Data
 If user wishes to display data, enter IW=1
 If IW=1, call DISP, if IW $\neq$ 1 bypass step 12
- STEP 13: Write Output File
 If user wishes to create an output file, enter IFW=1
 If IFW $\neq$ 1, bypass step 13.
 If IFW=1, Write counting indices NR, NM, NN, NUM, NC, NP, NL,
 NOU, NBDE, and IFG
 For each route write RNAM(I), RP(I,1), RP(I,2), and IPD(I)
 For each message write MNAM(I), MP(I,1) MP(I,2), MD(I,1)
 MD(I,2), MD(I,3), and MD(I,4)
 For each channel, write CNAM(I), CD(I,1), CD(I,2), CD(I,3), and
 CD(I,4)
 For each processor, write PD(I)
 For each link, write LP(I,1), LP(I,2), LP(I,3), LD(I,1), and
 LD(I,2)
 For each route, for each message, write RM(I,J,1), RM(I,J,2),
 RM(I,J,3), and RM(I,J,4)
 (or write a line of "*" if array to be input later)
 Write DD(1), and DD(2)
 Write GPD(I,1) and GPD(I,2), for I=1,...,11
- Step 14: Terminate

Also included in the internal logic of each of the steps is a means of prompting using either short or long forms of the prompts. If the short form is elected in step 0, then in each succeeding step the first of a series of similar prompts will contain a long, sentence-like prompt. All subsequent prompts for similar data will be given in short form only. This was instituted to decrease the running time of program CREATE.

2.3 SUBROUTINES IN CREATE

Exhibit 2-9 lists the six subroutines which are part of program CREATE and indicates their basic functions. Five of the six subroutines are used for error checking during the course of using the program CREATE. The sixth subroutine permits display of the input data in an easy to read format. Each of these subroutines is discussed in greater detail in the following paragraphs.

Subroutine CHECK compares the name input by the user for the processor at a node against the symbols "TCT" and "TCS". If the processor type which is input by the user does not correspond to one of these two names, CHECK will so indicate to the user and prompt for the correct name. The checks and prompts will continue until the user inputs either "TCT" or "TCT".

Subroutine CHECK1 compares the channel name which the user has input as the channel connecting two nodes with the list of channel names which was input earlier. If the name the user has input is not contained in the previously defined list, CHECK1 will so indicate to the user and request another name to be input. This prompting will continue until a proper channel name is input. When CHECK1 sees a proper channel name, it will assign and return the number of that channel to the main program where channel numbers, not names, are used for identification.

EXHIBIT 2-9: SUBROUTINES IN CREATE

<u>Name</u>	<u>Function</u>
CHECK	Checks processor type
CHECK1	Checks channel name, assigns channel number
CHECK2	Checks message name, assigns message number
CHECK3	Checks probability range
CHECK4	Checks proportion range
DISP	Displays data in easy to read format

Subroutine CHECK2 performs much the same function as subroutine CHECK1. Here the name of the message caused as an output by a particular message is compared with the previously input list of message names. If the name is not contained in that list, the user is prompted to enter another name until a legal message name is seen. When this occurs, subroutine CHECK2 will determine the number of the message input and return that number to the main program.

During the course of CREATE, the user is prompted for the probabilities of various actions. When the user has answered such a prompt, the number is checked in the main program to see if it is in the range 0-1. If the input number is outside this range, CHECK3 is called and the user is prompted to enter a number in the range 0-1.

Subroutine CHECK4 is identical to subroutine CHECK3 except it is used to check the input values for proportions rather than probabilities. If the input proportion is outside the range of 0-1, CHECK4 is called.

When all the data have been input, the user is presented with the instruction, "If you want to display the output file, enter a 1". If the user enters a 1, subroutine DISP is called and the data are printed at the terminal in an easy to read format. Subroutine DISP is essentially program DISPLAY; however, the data are passed as subroutine arguments rather than being read in from an input file, as is the case with program DISPLAY.

2.4 DATA FILE

Exhibit 2-10 contains a description of the data file which is the output of program CREATE. The variable names are as defined in exhibits 2-1 through 2-7 in which the formats of the lines in the output files are

EXHIBIT 2-10: OUTPUT FILE FORMAT

<u>Line</u>	<u>Elements</u>	<u>Format</u>
COUNTERS		
1	NR, NM, NN, NUM, NC, NP, NL, NOU, NBDE, IFG	10(I3, 1X)
ROUTE INFORMATION		
2 thru NR+1	RNAM(I), RP(I,1), RP(I,2), IPD(I)	A8, 3(1X,I3)
MESSAGE INFORMATION		
NR+2 thru NR+NM+1	MNAM(I), MP(I,1), MP(I,2), MD(I,1), MD(I,2), MD(I,3), MD(I,4)	A8, 2(1X,I3), 4(1X, E9.3)
CHANNEL INFORMATION		
NR+NM+2 thru NR+NM+NC+1	CNAM(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)	A8, 4(1X,E9.3)
NUMBER OF TEMPLATES		
NR+NM+NC+2 thru NR+NM+NC+NP+1	PD(I)	2X, I3
LINK DATA		
NR+NM+NC+NP+2 thru NR+NM+NC+NP+NL+1	LP(I,1), LP(I,2), LP(I,3), LD(I,1), LD(I,2)	3(I3, 1X), 2(E12.6, 1X)

EXHIBIT 2-10: OUTPUT FILE FORMAT (Concluded)

ROUTE CROSS MESSAGE ARRAY (If IFG=1 the array is a line of 50 "s")

NR+NM+NC+NP+NL+2	}			
thru				
NR+NM+NC+NP+NL+NM+1			RM(1,J,1), RM(1,J,2), RM(1,J,3), RM(1,J,4)	4(E9.3,1X)
thru				
NR+NM+NC+NP+NL+(NR-1)*NM+2	}			
thru				
NR+NM+NC+NP+NL+NR*NM+1			RM(NR,J,1), RM(NR,J,2), RM(NR,J,3), RM(NR,J,4)	4(E9.3,1X)

DISK ACCESS TIMES

NR+NM+NC+NP+NL+NR*NM+2	DD(1), DD(2)	2(E9.3, 1X)
------------------------	--------------	-------------

GENERAL PROCESSOR DATA

NR+NM+NC+NP+NL+NR*NM+3	}			
thru				
NR+NM+NC+NP+NL+NR*NM+13			GPD(1,1), GPD(1,2)	2(E9.3,1X)

also shown. This output file is set up so that the first line contains the counters which are used to determine the number of lines in the different sections of the data file. For example, NR indicates the number of lines in the Route Information section; NM, the number of lines in the Message Information section; etc. The file is implemented in this manner so that the controls for reading the file would be contained within the file itself and no other information need be given. This data file is in a format compatible with the other three programs in the TOS analysis package.

2.5 CODE OF PROGRAM CREATE

Program CREATE is written in FORTRAN IV using library subroutines existing on the Amdahl 460/V7 computer at The University of Michigan. The code is liberally interspersed with comment statements which indicate the function of the various parts of the program.

The initial version of program CREATE was a short, simple program which would suffice for most applications; however, during the checkout of the other programs in the analysis package, it was obvious that additional logic needed to be included in CREATE to permit its use with various test cases. In its simplest form, CREATE was implemented assuming a standard division configuration which had more than one brigade, more than one battalion for each brigade, and more than one other TOS user. In this case, simple DO LOOPS will suffice; however, test cases were implemented which contained only one or two nodes and in those situations additional logic was required to bypass unnecessary DO LOOPS. Thus, there are a large number of "IF" comparisons in CREATE. These are checks to bypass portions of code which are not needed.

The evolution of program CREATE included a decision to implement a series of short prompts as part of the program. There are essentially two sets of queries contained in CREATE. One set contains long, sentence-like queries. In the other set the queries are short and are used to prompt for subsequent entries of a similar type. Thus, there are also numerous logic checks to determine when these short prompts are to be called. This also necessitated additional logic to account for the various test cases which were to be implemented, although most of that logic would not be exercised if the program is used for the input of a standard division configuration.

All of the input statements in program CREATE are free format and make use of the library program, FREAD, which is available on the Amdahl computer on which program CREATE was written. These free format read statements were implemented to make the user's task of inputting the data easier. If program CREATE is implemented on a different computer system, the user must determine if FREAD is available. If FREAD is not available, the input statements need to be replaced with either conventional formatted statements or with whatever free format subroutine is available on the host machine.

3.0 PROGRAM MODIFY

Program MODIFY was written to facilitate alteration of an existing data file. The program is written in FORTRAN IV and is designed to be run interactively from a terminal. This section provides information which may be of use to a programmer who wishes to modify the logic of this program or to transfer this program to a new computer system. Section 3.1 contains a description of the structure of the program MODIFY and provides a list of the subroutines which are used. Section 3.2 contains a discussion of the code and how to modify the existing code within the framework of the structure described in section 3.1. Section 3.3 contains flowcharts of the main program and subroutine.

3.1 Structure of Program MODIFY

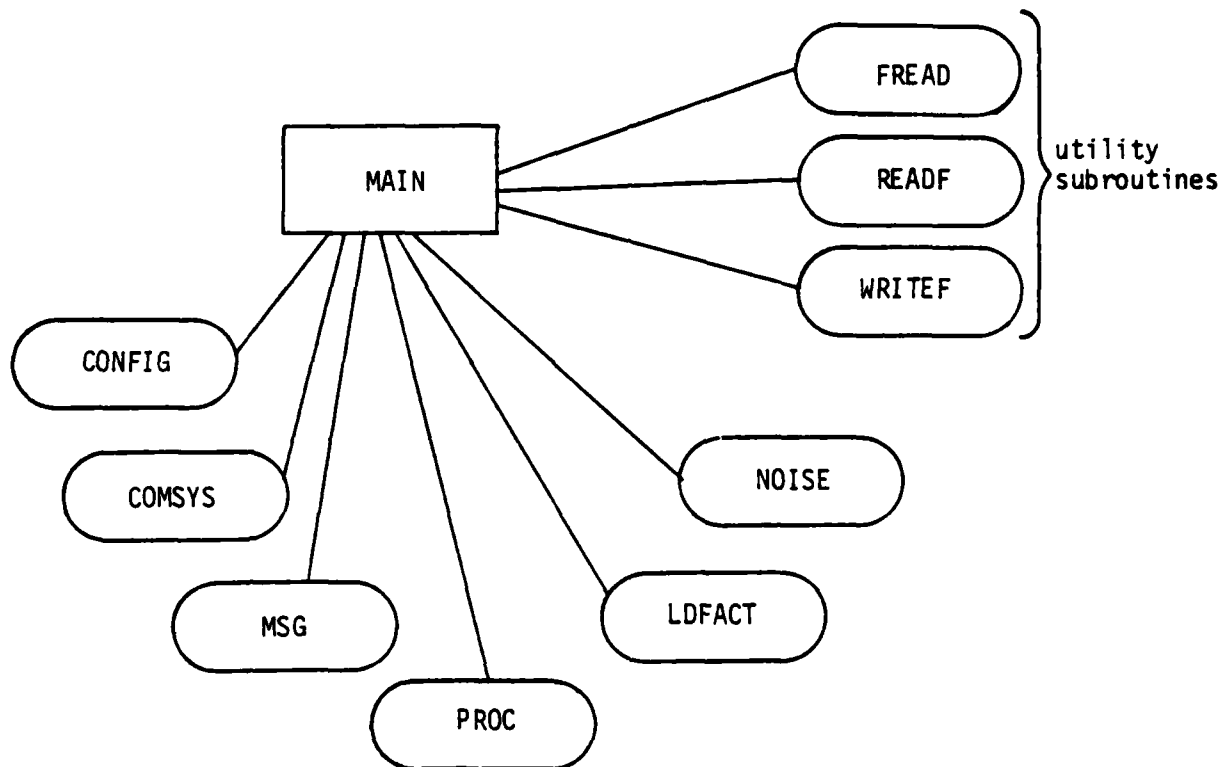
Program MODIFY is implemented as a main executive program which calls several subroutines to perform the actual modification of input data. Exhibit 3-1 illustrates the general structure of the program.

The data file for the TOS analysis package consists of data of six logical types. Exhibits 2-1 through 2-6 list the data by type. This classification of data by type is reflected in the structure of program MODIFY. The main executive program calls six subroutines to perform modification of data, one for each type of data. Exhibit 3-2 relates these six subroutines to the type of data which they modify.

Program MODIFY also uses three utility subroutines to perform I/O functions. Subroutine READF reads an existing data file from I/O

EXHIBIT 3-1: STRUCTURE OF PROGRAM MODIFY

Main Executive Program and Subroutines



unit 1. Subroutine WRITEF writes a new data file onto I/O unit 2. Subroutine FREAD is a general free-format read routine which is discussed in section 2.3.

3.2 CODE OF PROGRAM MODIFY

The code of program MODIFY was produced using a top-down approach. The result is a modular program which consists of a short executive main program and a number of specialized subroutines. Section 3.1 describes the structure of program MODIFY and lists its subroutines. A detailed flowchart of program MODIFY is found in section 3.3.

Three utility subroutines are used to perform I/O functions. READF reads the data file from I/O unit 1. This routine was created by copying the code used to perform the same function in program DISPLAY. WRITEF writes the new data file to I/O unit 2 and was created by copying code used by program CREATE. The free-format read subroutine, FREAD, is a library program on the Amdahl computer at The University of Michigan.

The actual modification of data is performed by the six subroutines listed in exhibit 3-2. The six subroutines correspond to the six types of data in the data file. The coding of these subroutines was performed to allow modification of those data which were changed in the course of the analysis of the TOS communications network; however, the program does not currently allow modification of all of the data items.

Subroutine COMSYS allows the user to modify and examine any of the communication system data. Subroutine LDFACT allows modification of some of the route cross message data. The user is allowed to multiply message traffic rates for the entire communications network by a factor. Subroutine NOISE allows the user to set bit error rates by channel, and to

EXHIBIT 3-2: DATA MODIFICATION SUBROUTINES USED BY PROGRAM MODIFY

<u>Subroutine</u>	<u>Type of Data Modified</u>
CONFIG	System Configuration
COMSYS	Communications System
MSG	Message Data
PROC	Processor Data
LDFACT	Route Cross Message Data
NOISE	Bit Error Rates

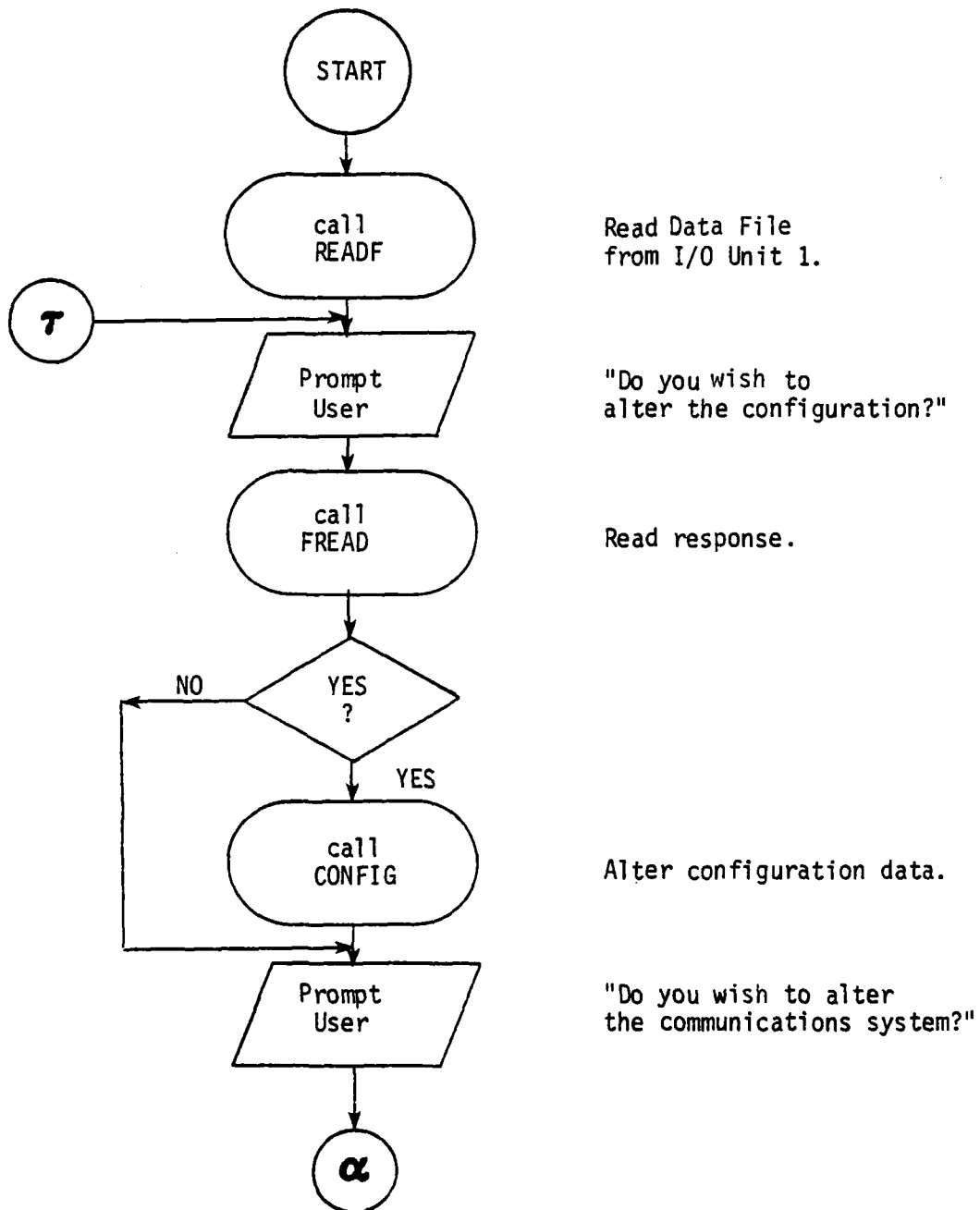
display bit error rates. All of the code which displays input data was copied from program DISPLAY. Subroutines CONFIG, MSG, and PROC are dummy routines, since it was not necessary, in the course of the analysis of TOS, to use program MODIFY to change the values of any of the configuration, message, or processor data.

In order to extend the range of the current version of MODIFY to allow modification of other data items, the programmer may write new versions of some or all of the six special subroutines which perform data modification. Some changes would be straightforward. For example, it would be easy to write into subroutine PROC the ability to replace most of the processor data; however, some other modifications are more complex, and could involve the creation of interdependence among the six subroutines. For example, the insertion or deletion of nodes in CONFIG, which handles configuration data, may necessitate insertion, deletion, or adjustment of communication system data, message data, route cross message data, and error rate data. Such changes should preferably be handled by requiring the user to run program CREATE to create a new data file. The current version of MODIFY is designed to replace existing values in the input data file, not to insert or delete data.

3.3 FLOW CHARTS OF THE PROGRAM MODIFY AND SELECTED SUBROUTINES

Exhibits 3-3 through 3-6 contains detailed flowcharts of the main program and subroutines COMSYS, LDFACT, and NOISE.

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY



-- Continued --

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)

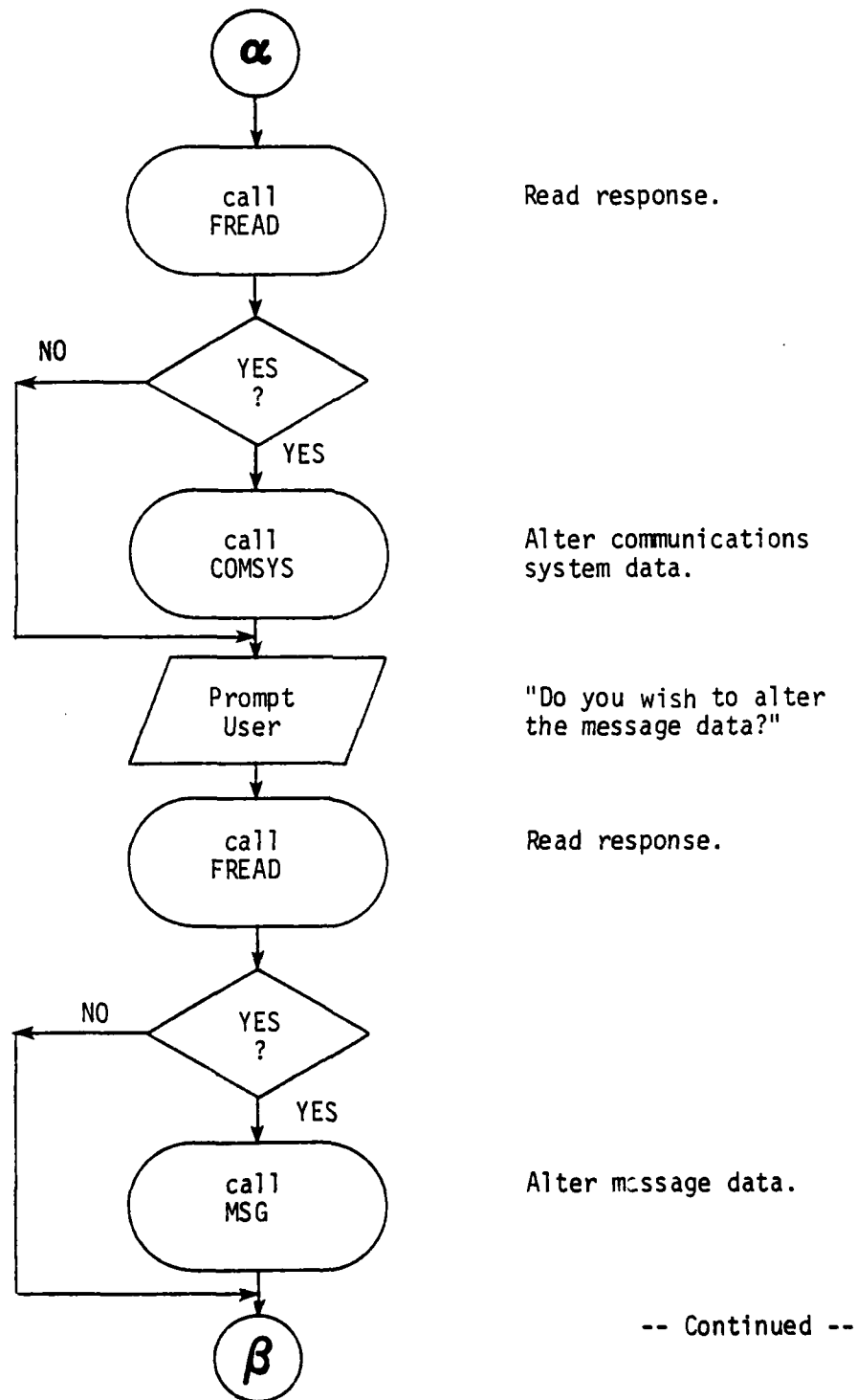
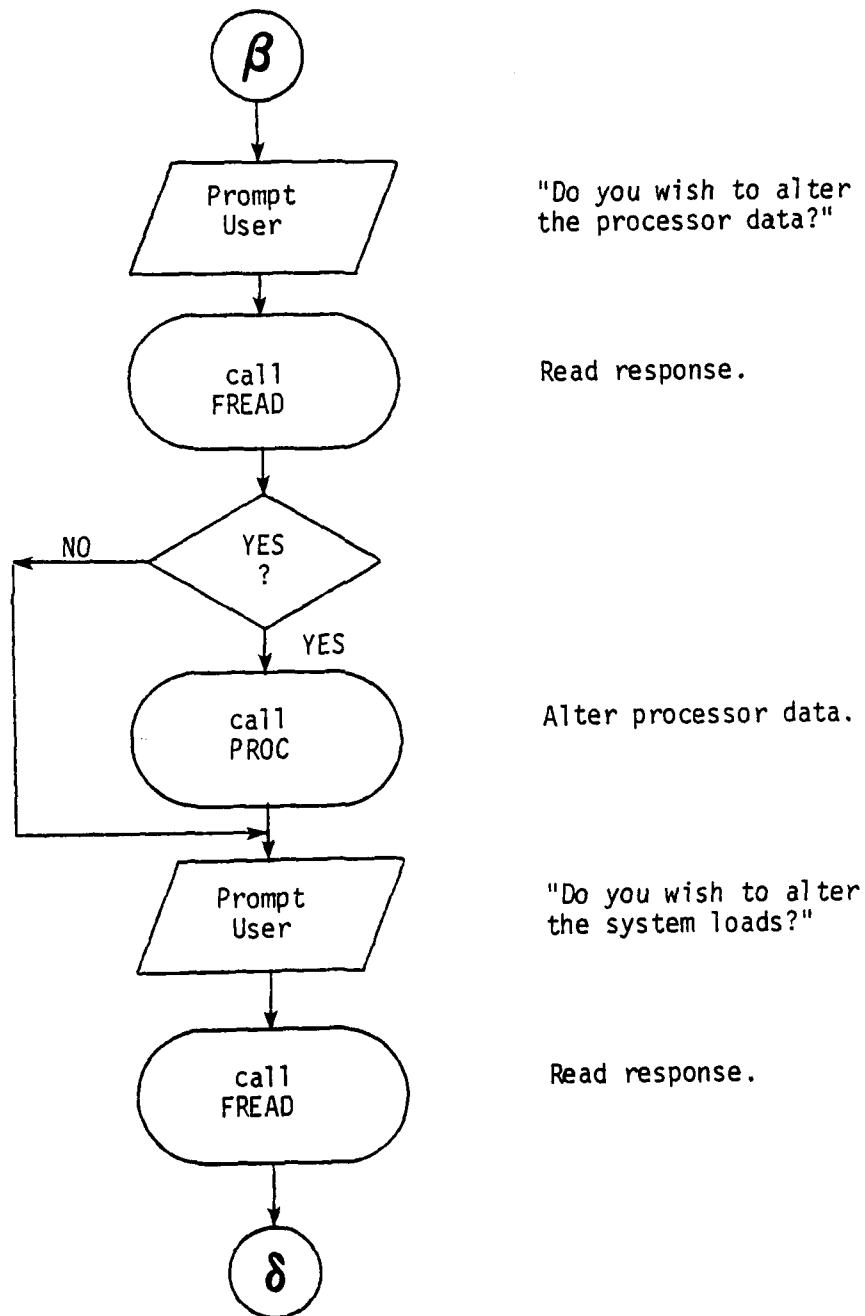


EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)



-- Continued --

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)

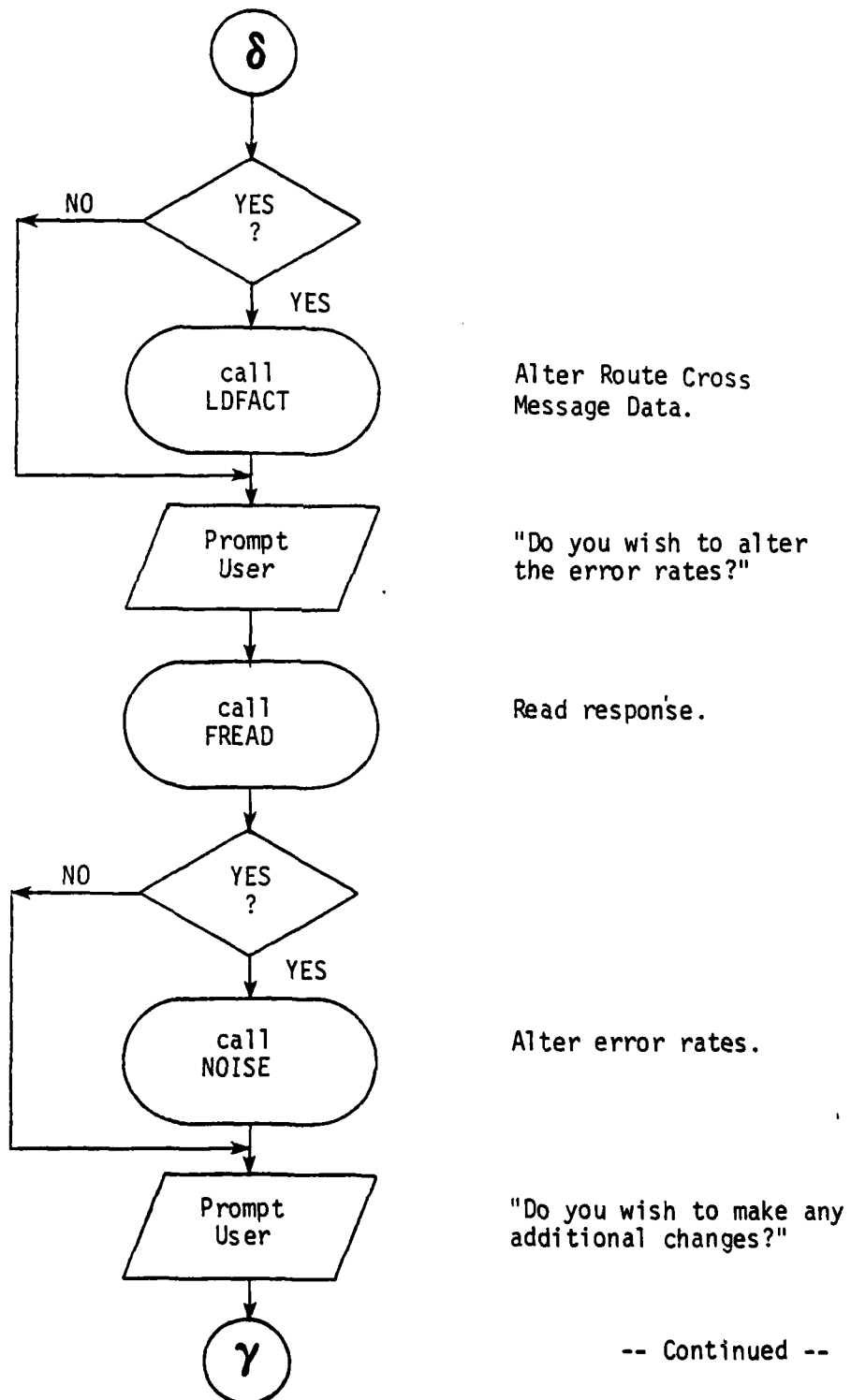


EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Concluded)

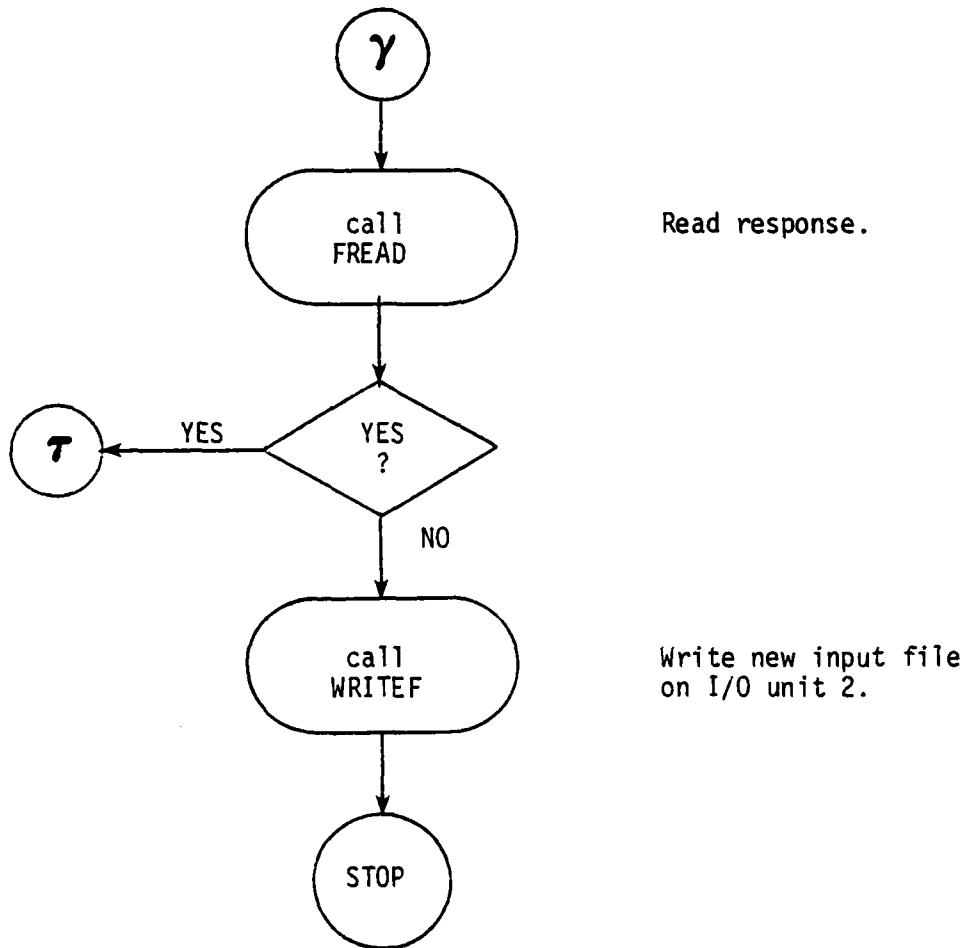


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS

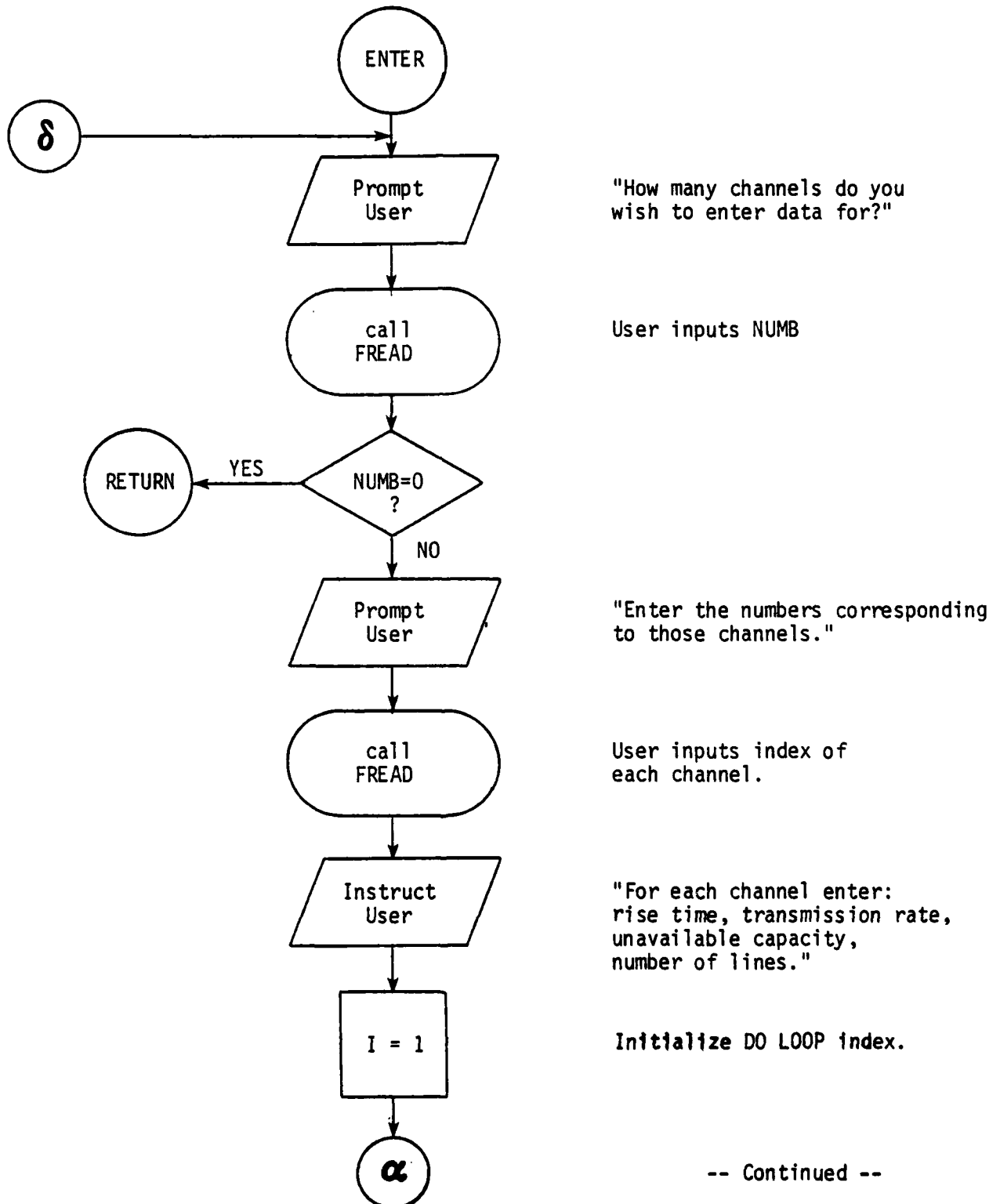


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS (Continued)

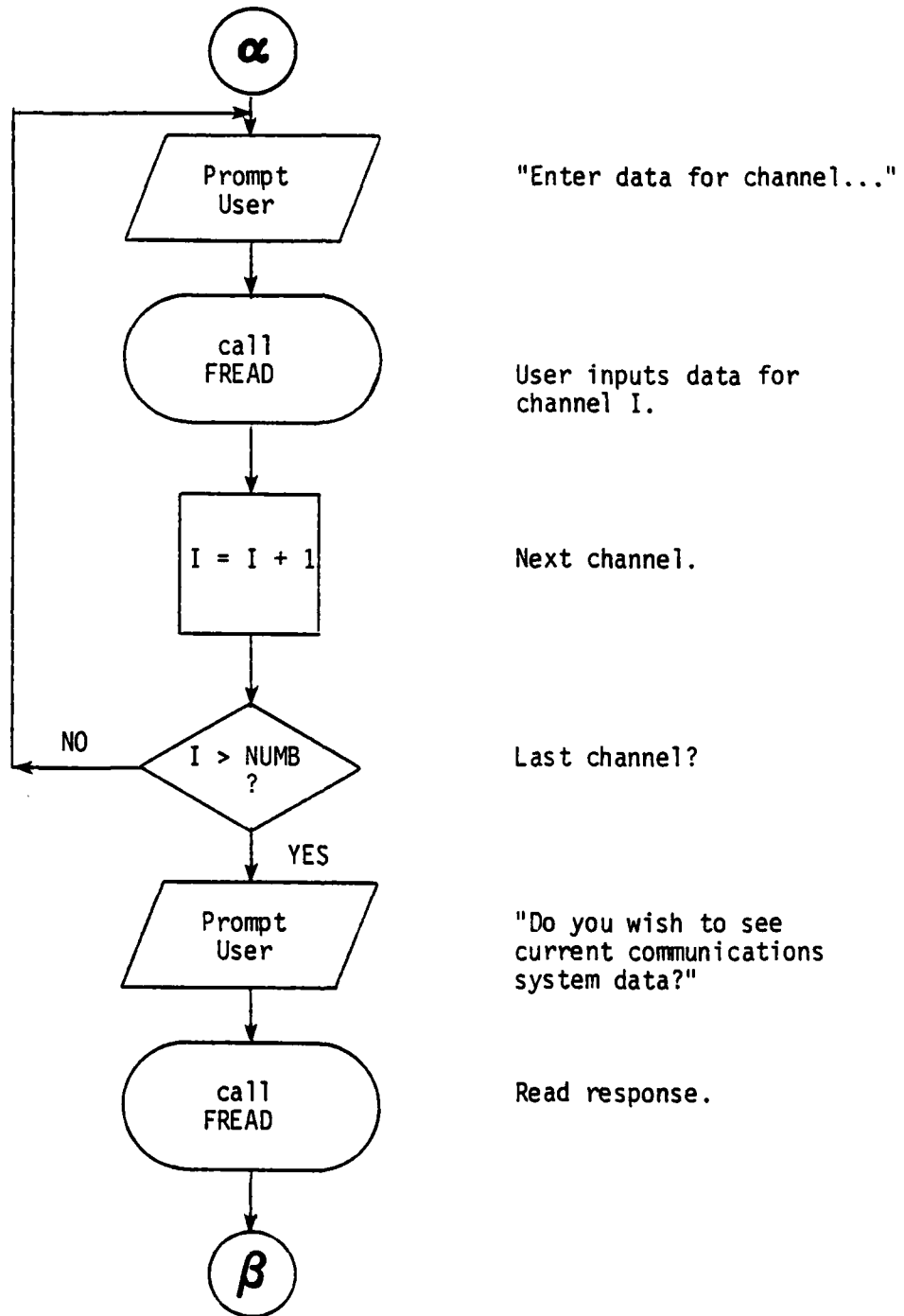


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS (Concluded)

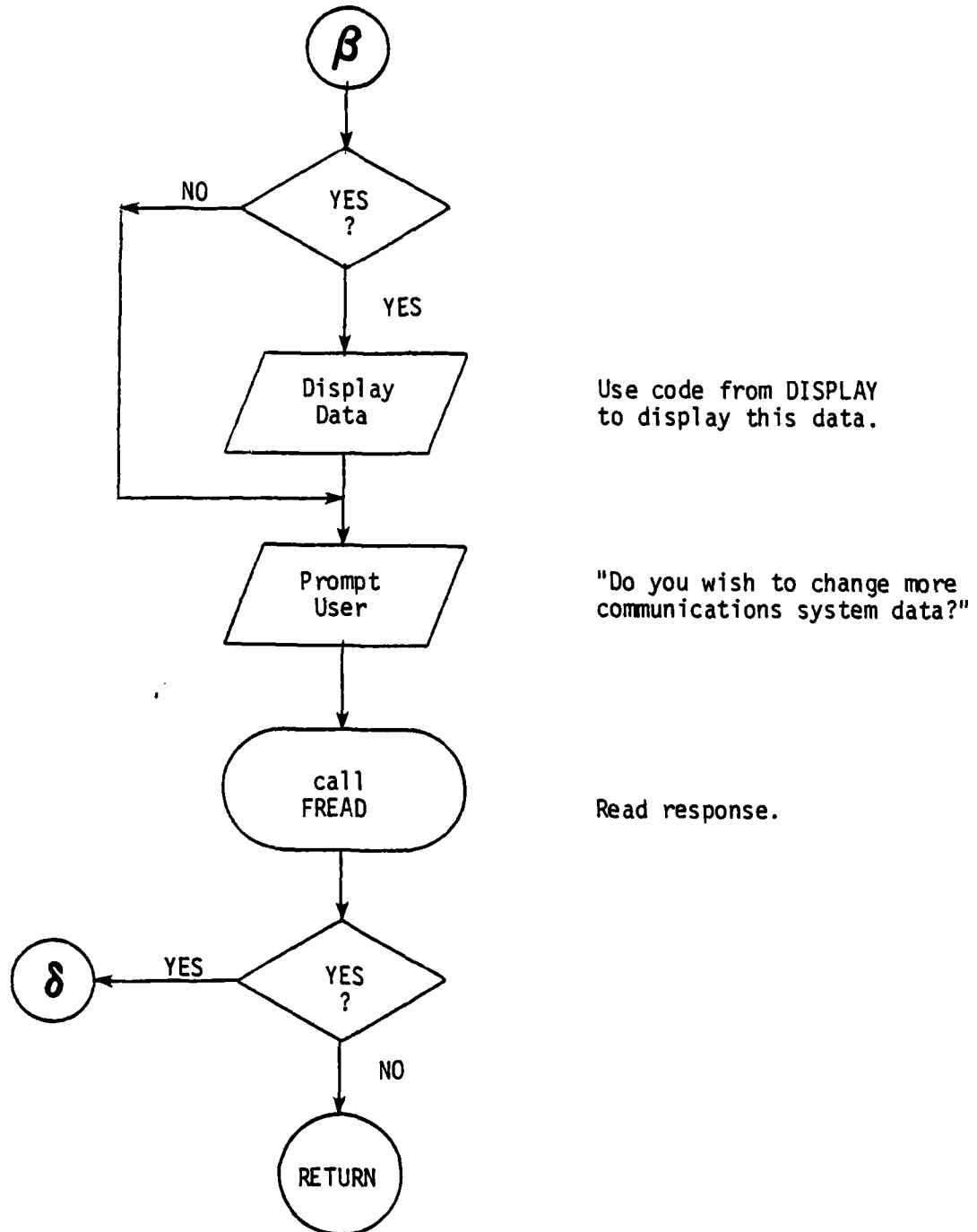


EXHIBIT 3-5: FLOWCHART OF SUBROUTINE LDFACT

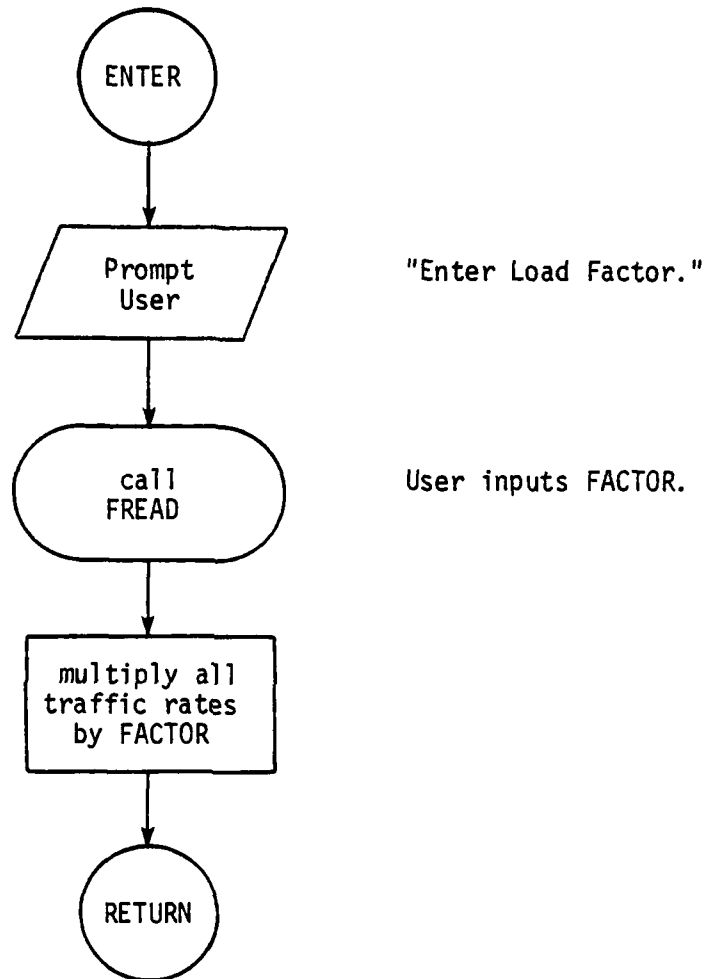


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE

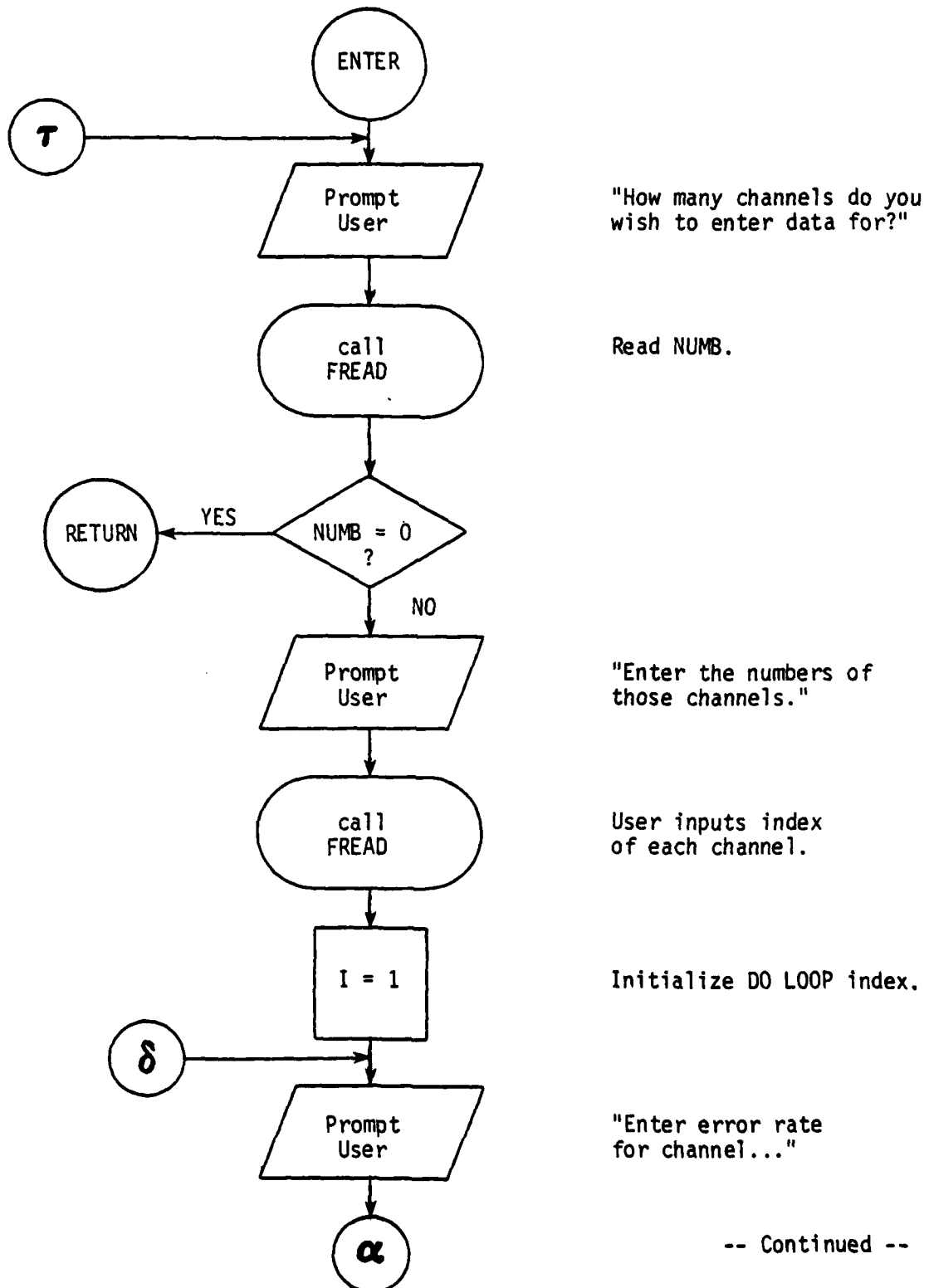


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Continued)

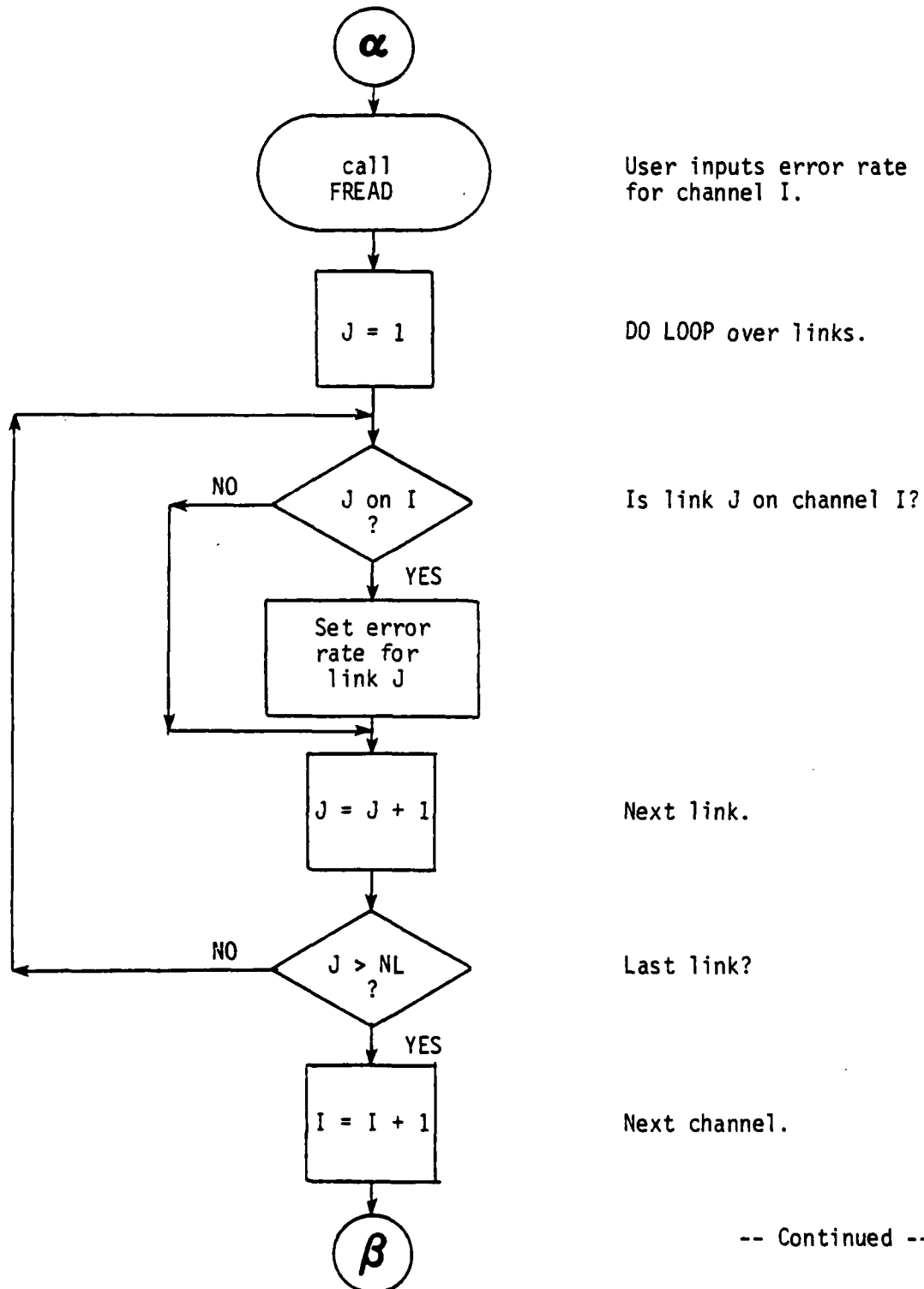


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Continued)

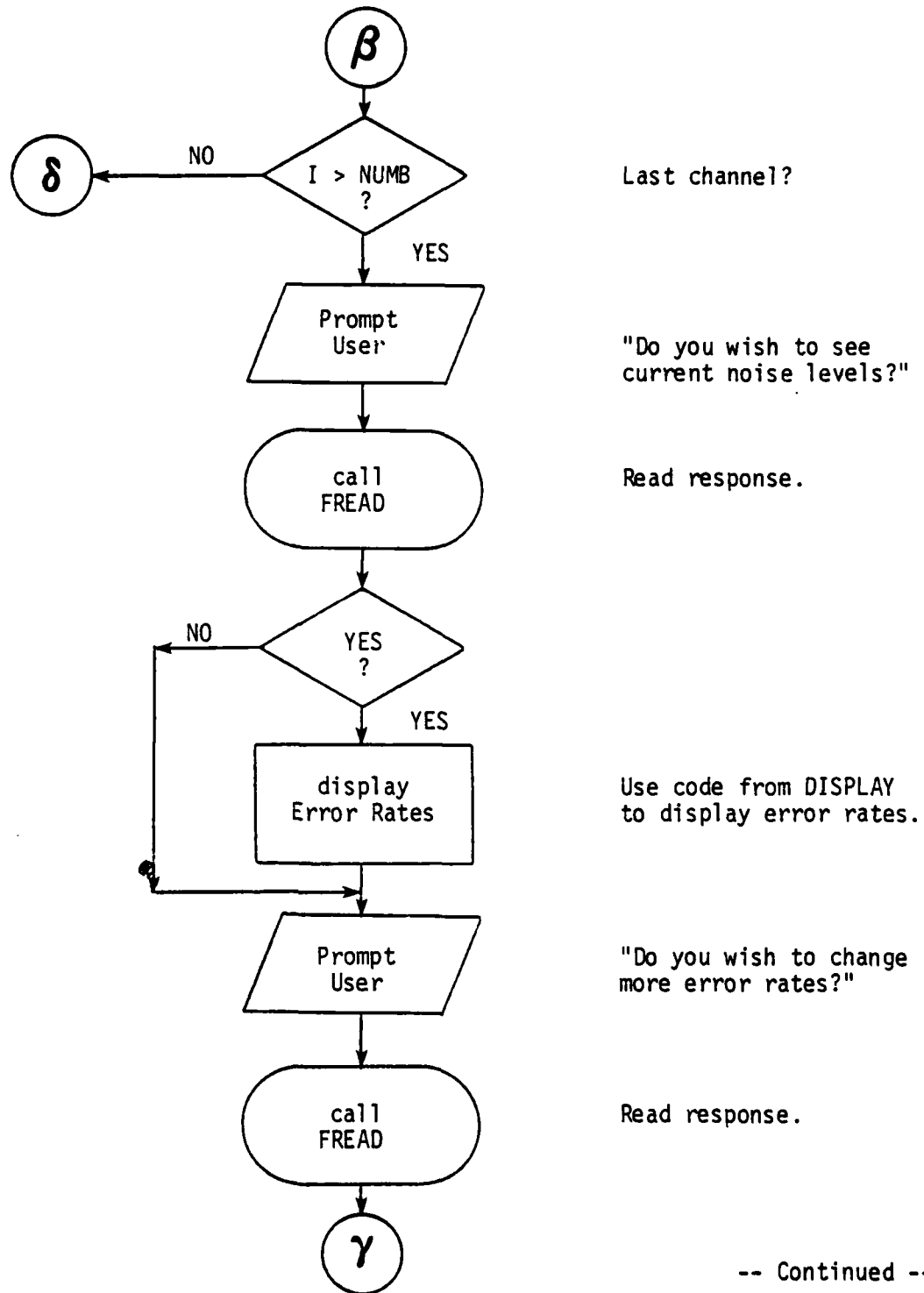
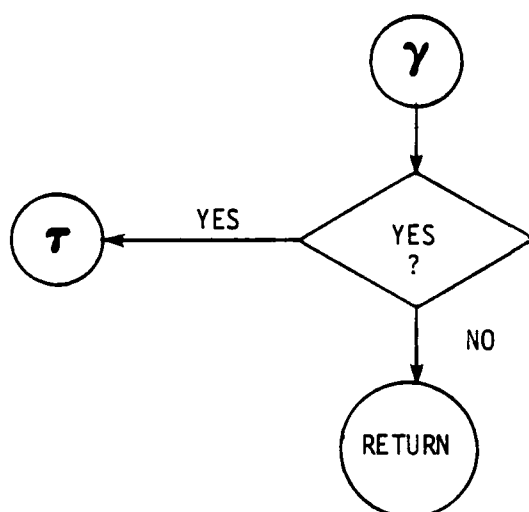


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Concluded)



4.0 PROGRAM DISPLAY

The third of the four programs in the TOS analysis package was written to assist the user in examining the contents of the data file. Program DISPLAY will read the data file and print the information with appropriate headings so that the user can easily examine the contents of the data file.

Program DISPLAY will mesh with the outputs of both program CREATE and program MODIFY. The read statements in program DISPLAY are copies of the write statements from the other two programs, with the word READ inserted in the place of WRITE. This was done to assure compatibility between the programs. Program DISPLAY is set up to operate on the Amdahl computer at The University of Michigan. This computer allows the user to specify input devices on the job control cards. Thus, the read statements in program DISPLAY are not the normal "READ(5,)" statements which are found in most FORTRAN codes, but rather are READ(2,). Thus, the user is required to inform the operating system that device two contains the file which is to be displayed. If DISPLAY is implemented on another computer system, the compatibility with the read statements and the job control cards must be ensured so that program DISPLAY is linked to the proper data file.

Program DISPLAY was written in standard FORTRAN IV. The listing contained in Appendix A contains extensive comment lines which are used to identify the functions being performed. The first card read from the data file contains the counters which are used to control the DO LOOPS

which make up the remainder of program DISPLAY. The ordering of the route numbers, messages, and channels is the same as that contained in the other three programs and is used internally in DISPLAY. Program DISPLAY contains the option of reading data files which do not contain the Route Cross Message Array. If such a data file is used as the input for DISPLAY, the header for the Route Cross Message Array is printed in the output along with a message saying that the array is to be read in later.

5.0 PROGRAM COMPUTE

Program COMPUTE performs the computations specified by the model and displays the outputs. COMPUTE is organized into five modules, as shown in exhibit 5-1. Three of the program modules, the Traffic Flow Module, the Operating Statistics Module, and the Performance Measure Module, correspond to the three modules of the same names in the mathematical model.¹ The correspondence is not quite exact due to programming considerations such as producing efficient code and minimizing the program's memory requirements. Nonetheless, the primary content of the modules is unchanged.

The program documentation consists of a series of exhibits presenting: (1) flowcharts; (2) descriptions of the program variables; and (3) use of I/O units. Flowcharts for the Input and Initialization and Traffic Flow Modules are presented in exhibits 5-2 and 5-3. A flowchart for the Operating Statistics Module is presented in exhibit 5-4. In this flowchart, two segments are labeled A and B. More detailed flowcharts for these segments are presented in exhibits 5-5 and 5-6. Exhibit 5-7 contains a flowchart for a subroutine called by segment A of the Operating Statistics Module. Flowcharts for the Performance Measure Module and the Output Module are presented in exhibits 5-8 and 5-9. A list and description of the variables used in the program are presented in exhibit 5-10. Exhibit 5-11 shows the use of logical I/O units.

¹See ARI Research Notes 80-13, Chapter 3.0.

EXHIBIT 5-1: ORGANIZATION OF COMPUTE

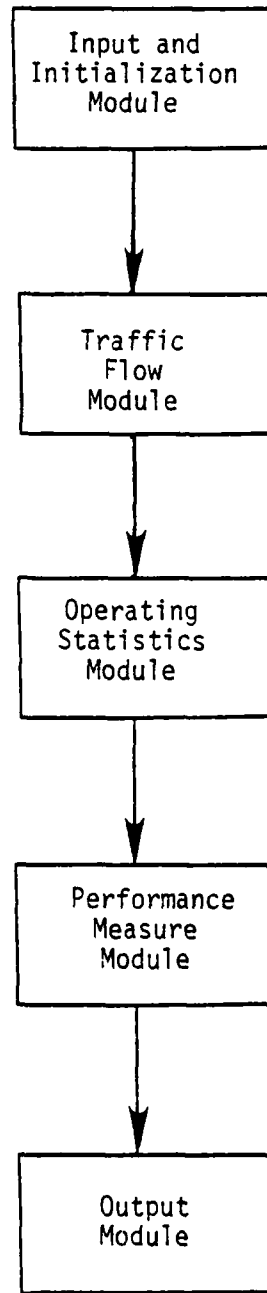


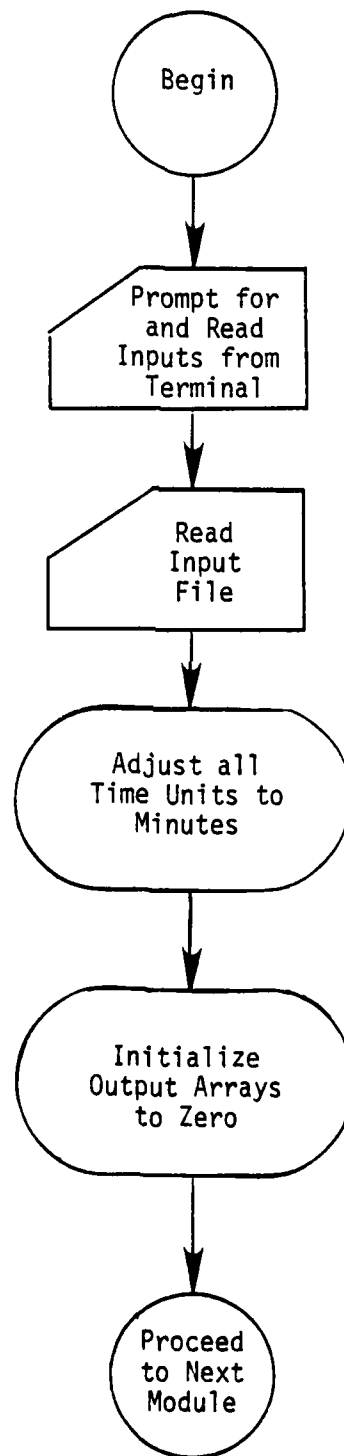
EXHIBIT 5-2: FLOWCHART FOR THE INPUT AND
INITIALIZATION MODULE

EXHIBIT 5-3: FLOWCHART FOR THE TRAFFIC FLOW MODULE

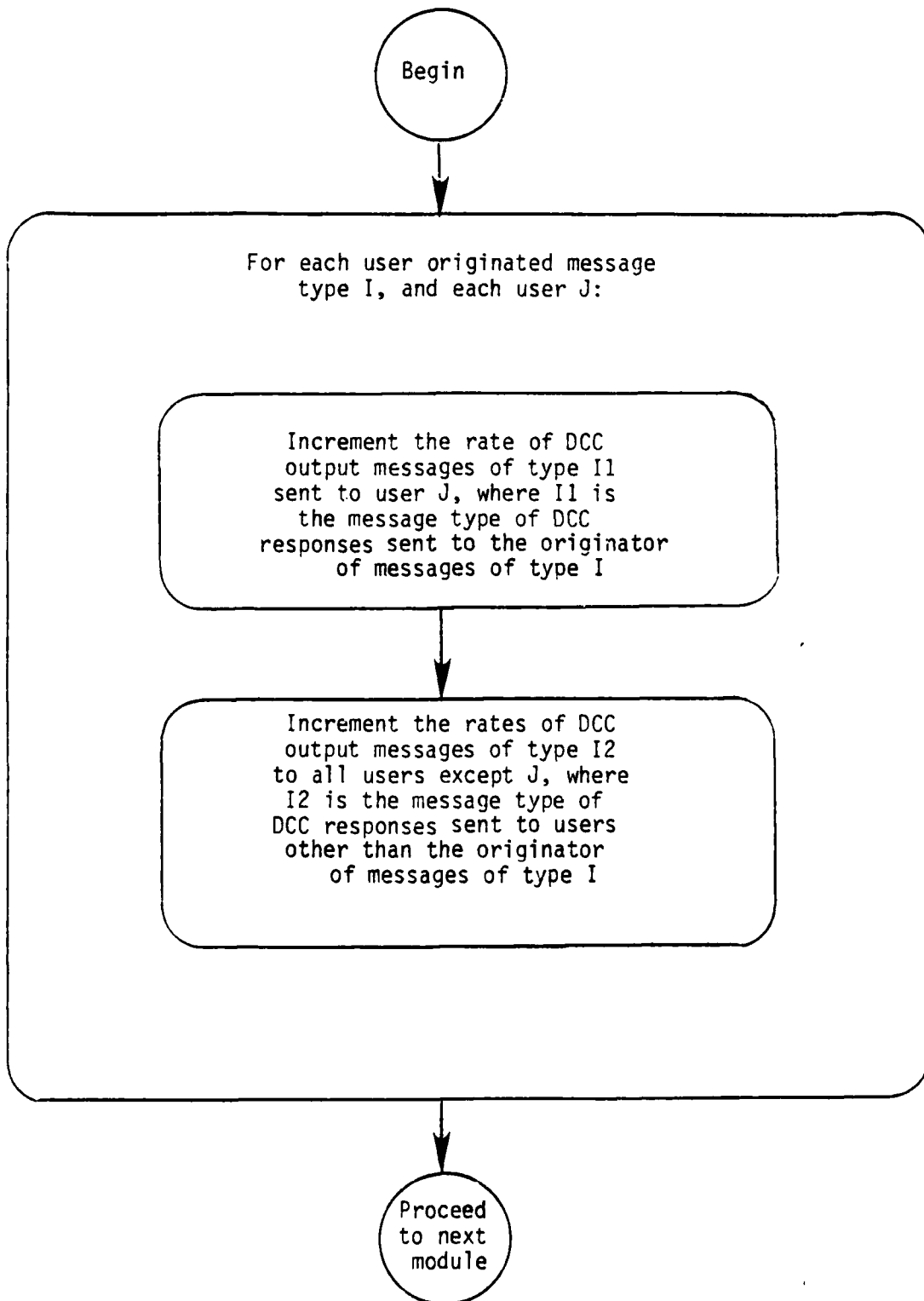


EXHIBIT 5-4: FLOWCHART FOR THE OPERATING STATISTICS MODULE

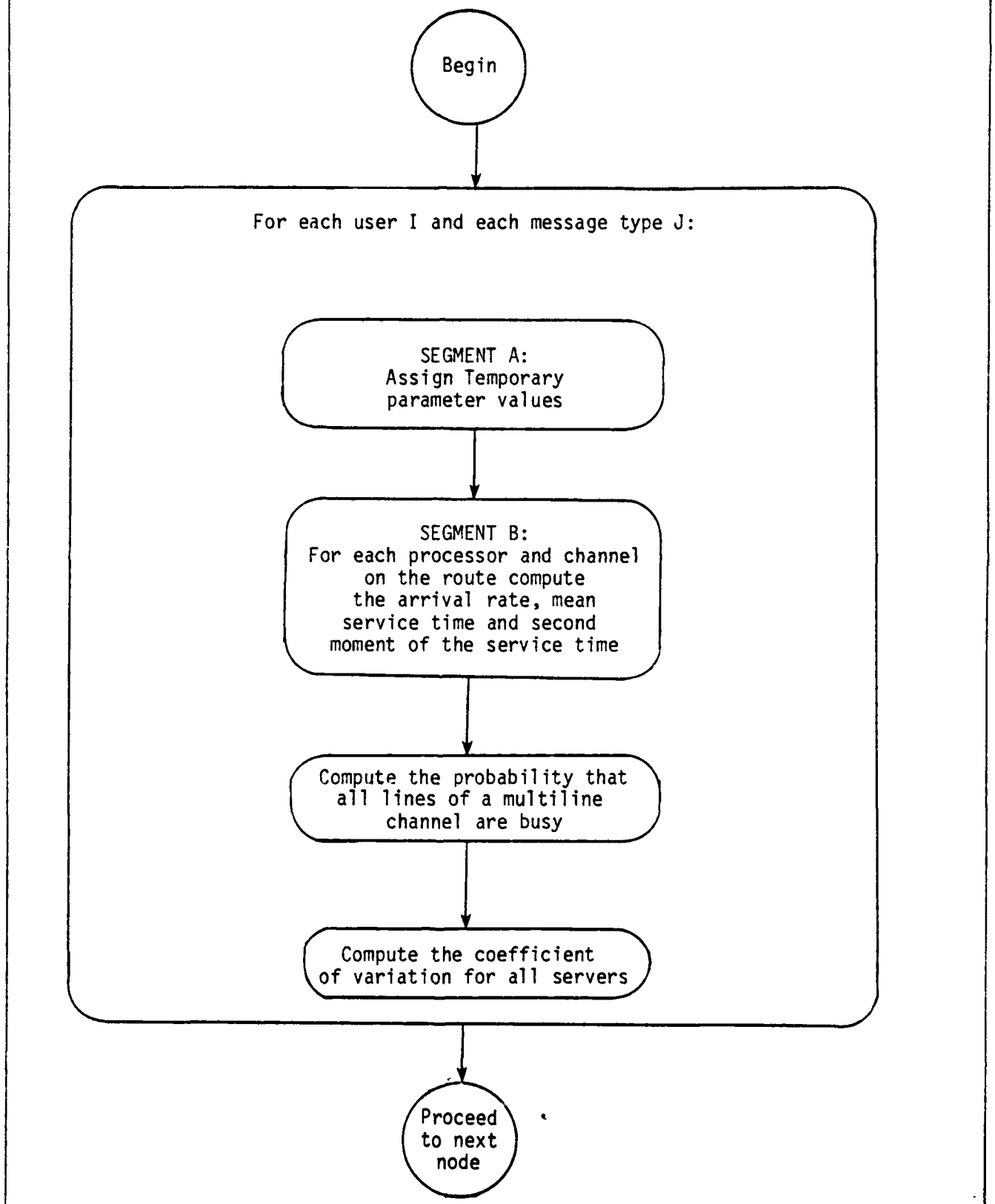


EXHIBIT 5-5: FLOWCHART FOR SEGMENT A OF THE OPERATING STATISTICS MODULE

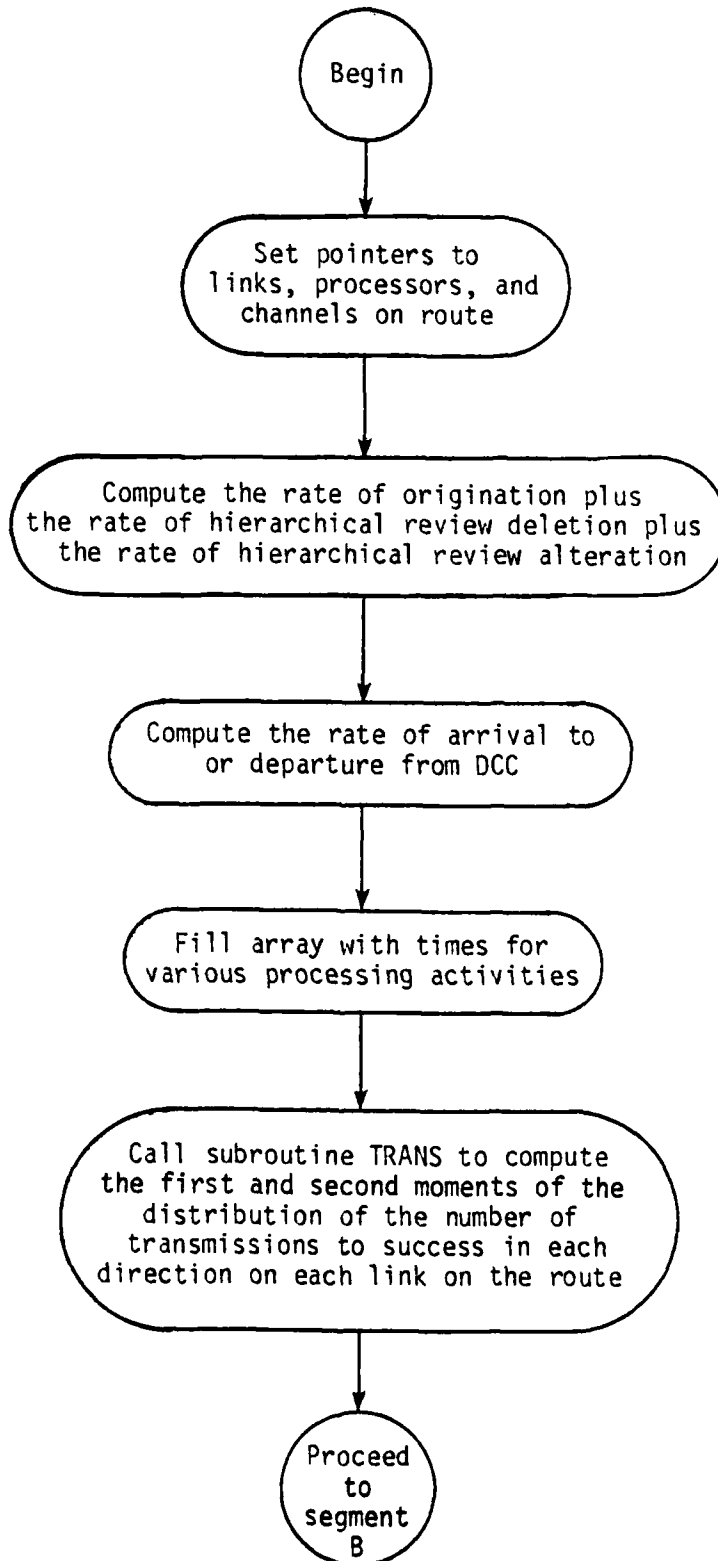


EXHIBIT 5-6: FLOWCHART FOR SEGMENT B OF OPERATING STATISTICS MODULE

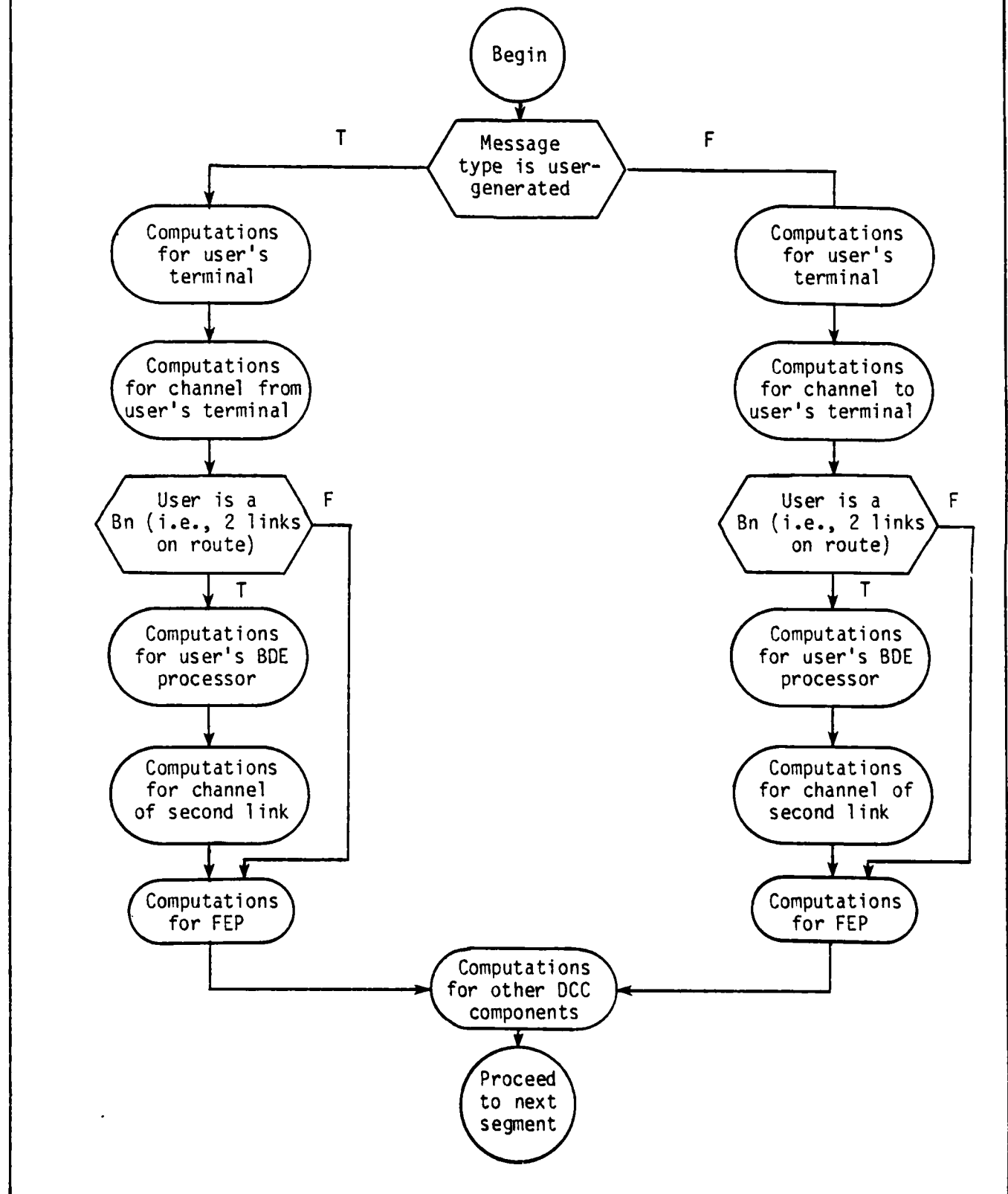


EXHIBIT 5-7: FLOWCHART FOR SUBROUTINE TRANS

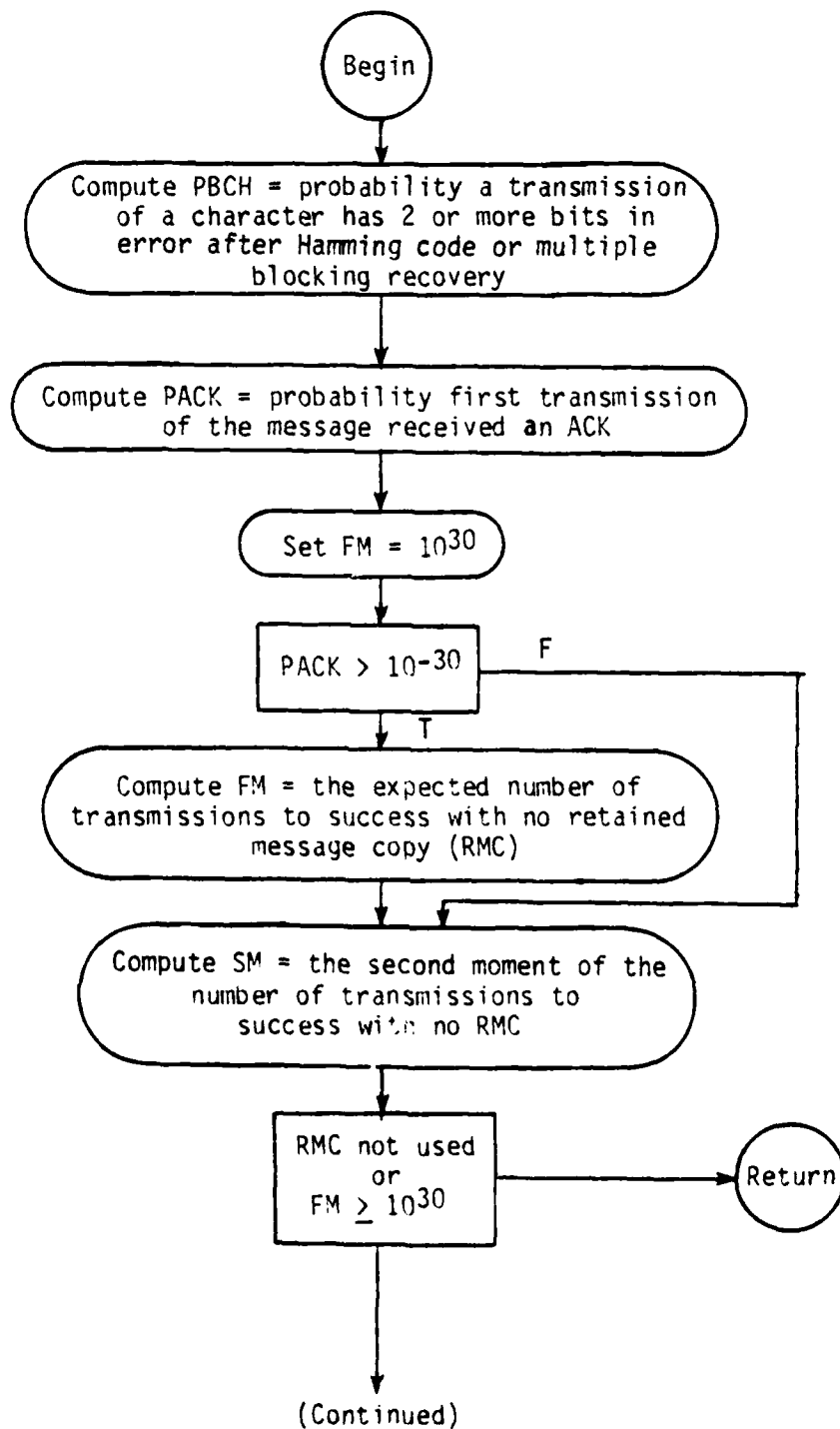


EXHIBIT 5-7: FLOWCHART FOR SUBROUTINE TRANS (Concluded)

(Continuation)

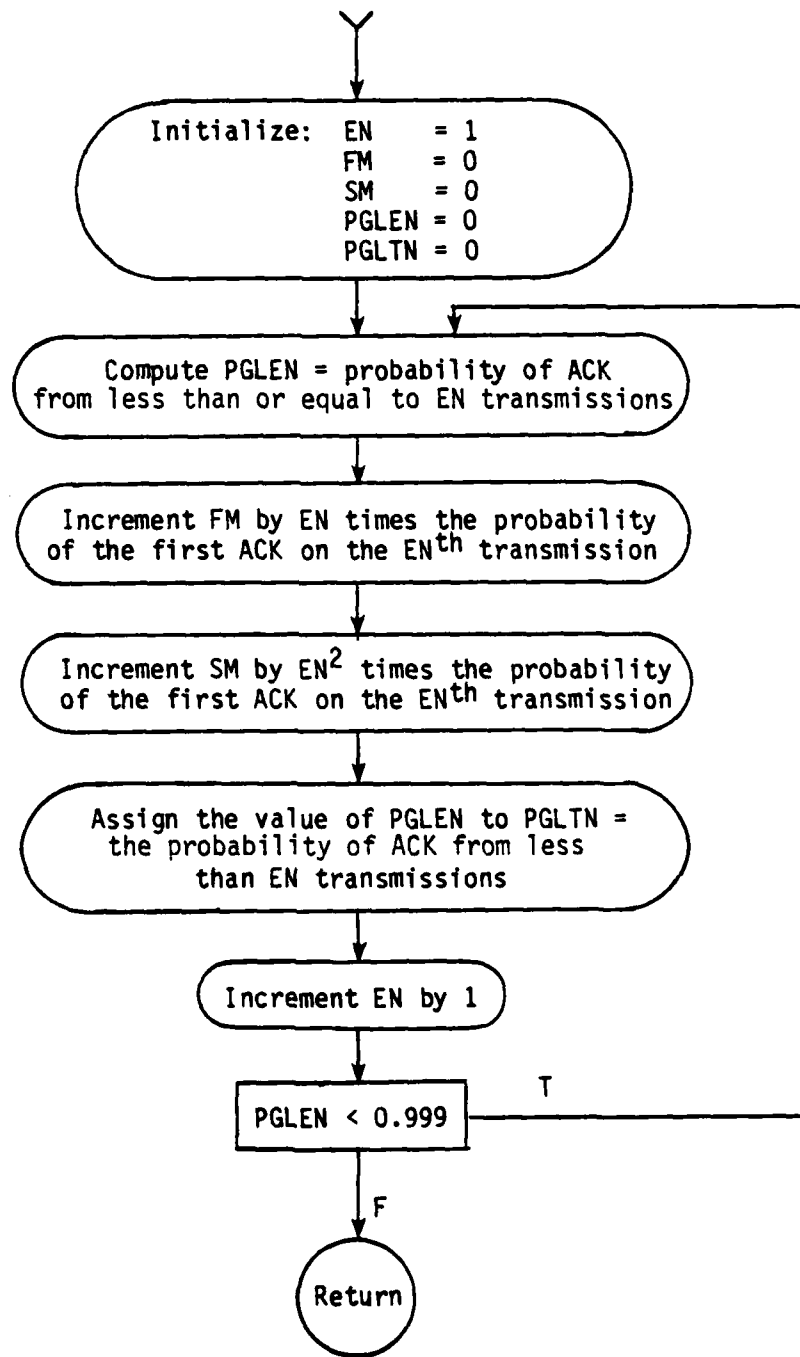


EXHIBIT 5-8: FLOWCHART FOR THE PERFORMANCE MEASURE MODULE

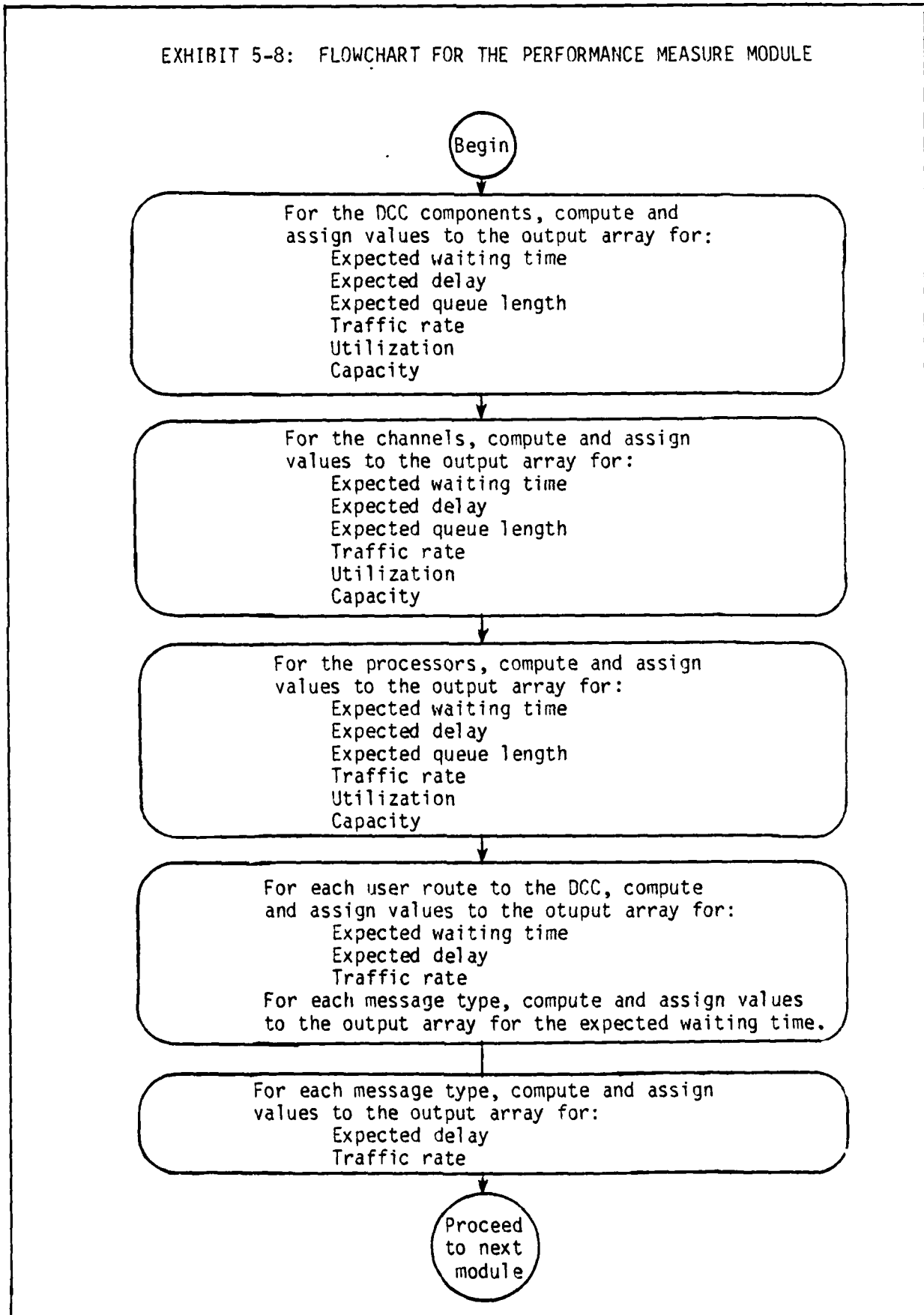


EXHIBIT 5-9: FLOWCHART FOR THE OUTPUT MODULE

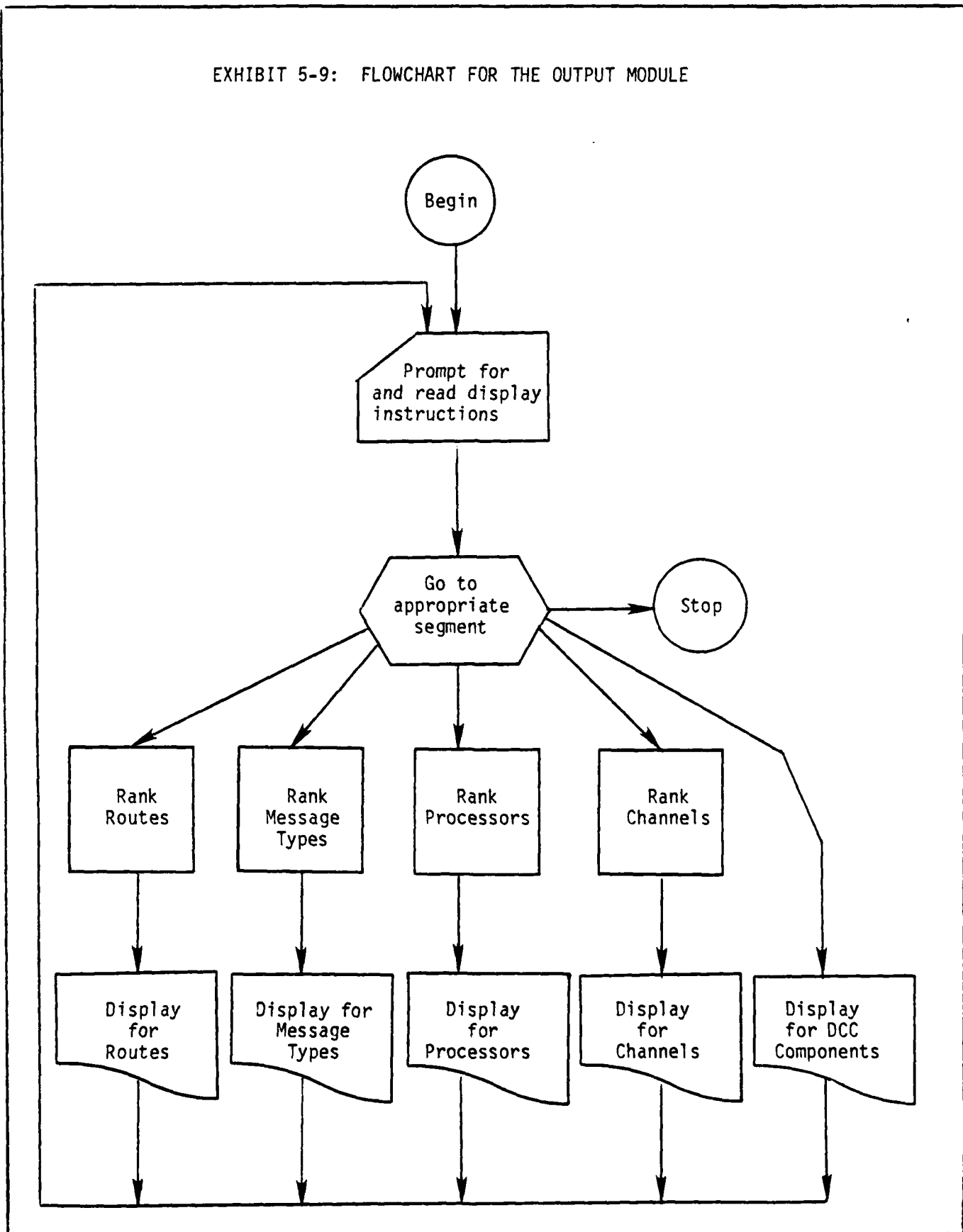


EXHIBIT 5-10: VARIABLES IN COMPUTE

<u>NAME</u>	<u>USE</u>
A	Temporary - Rate at which messages of type J travel to or from user I (including returns due to hierarchical review).
AM1	Temporary - Mean time for a processing or communications operation.
AM2	Temporary - Mean time for a processing or communications operation
AM3	Temporary - Mean time for a processing or communications operation.
B	Temporary - Rate at which messages of type J arrive or leave the DCC to/from user I.
CD (I, J)	Input - channel data: CD (I, 1) = overhead time per transmission on channel I CD (I, 2) = transmission rate on channel I CD (I, 3) = fraction of time used for voice on channel I CD (I, 4) = number of links on channel I
CNAM (I)	Input - name of channel I
COUT (I, J)	Output - channel outputs: COUT (I, 1) = expected delay on channel I COUT (I, 2) = expected queue length on channel I COUT (I, 3) = traffic rate on channel I COUT (I, 4) = utilization on channel I COUT (I, 5) = capacity on channel I COUT (I, 6) = rank on channel I
CS (I)	Temporary - probability all lines on channel I are busy.
C1	Temporary - pointer to channel of first link on a route.
C2	Temporary - pointer to channel of second link on a route.
DD (I)	Input - disk controller data: DD (1) = mean access time of message disk. DD (2) = mean access time of data disk.

-- Continued --

EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
DOUT (I, J)	Output - DCC component (except FEP) output: D (I, 1) = expected delay D (I, 2) = expected queue length D (I, 3) = traffic rate D (I, 4) = utilization D (I, 5) = capacity I = 1 => DBP I = 2 => Message Disk Controller I = 3 => Data Disk Controller
FACT	Temporary - factorial
FM	Temporary - Mean number of transmissions to success in one direction on a link
FM1	Temporary - Mean number of transmissions to success in one direction on a link
FM2	Temporary - Mean number of transmissions to success in one direction on a link
FM3	Temporary - Mean number of transmissions to success in one direction on a link
FM4	Temporary - Mean number of transmissions to success in one direction on a link
GPD (I, J)	Input - Processing time information: GPD (1, J) = TCS origination time GPD (2, J) = TCS send time GPD (3, J) = TCS receive time GPD (5, J) = TCS terminate time GPD (6, J) = FEP terminate time GPD (7, J) = TCT terminate time GPD (8, J) = TCT send time GPD (9, J) = TCT receive time GPD (10, J) = TCT terminate time GPD (11, J) = FEP originate time J = 1 => per message J = 2 => per character
I	Temporary - Counter or pointer
IA	Input - Pointer to current output table

-- Continued --

EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
IB	Input - Pointer to column for ranking.
IC	Input - Reordering switch.
IP	Temporary - Counter.
IPD (I)	Input - Processor data - type of processor at node I, 1 => TCS, 2 => TCT.
IRMC	Input - Retained Message Copy (RMC) Switch.
I1	Temporary - Pointer to message type of outputs in response to inputs of type I sent to the originator of the message of type I.
I2	Temporary - Pointer to message type of outputs in response to inputs of type I sent to users other than the originator of the message of type I.
J	Temporary - Counter and pointer.
JPD	Temporary - Type of current processor.
K	Temporary - Counter and pointer.
LARGE	Temporary - 10^{40} .
LD (I,J)	Input - Link data on error rates LD (I, 1) = Error rate on Link I for messages sent towards the DCC. LD (I, 2) = Error rate on Link I for messages sent towards a user.
LP (I,J)	Input - Pointers to devices on links. LP (I, 1) = Processor closest to user on Link I LP (I, 2) = Channel used by Link I LP (I, 3) = Processor closest to DCC on Link I
L1	Temporary - Pointer to the first link on the route from a user to the DCC.
L2	Temporary - Pointer to the second link on the route from a user to the DCC. Value zero if there is no second link.

-- Continued --

EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
MD (I,J)	Input - Message data: MD (I, 1) = Number of characters in Message Type I MD (I, 2) = DBP processing time of Message Type I MD (I, 3) = Number of message disk accesses for messages of type I MD (I, 4) = Number of Data Disk accesses for messages of type I
MDJ1	Temporary - Length of current message type
MNAM (I)	Input - Message Names
MOUT (I, J)	Output - Message outputs: MOUT (I, 2) = Expected delay in Message Type I MOUT (I, 2) = Traffic rate in Message Type I MOUT (I, 3) = Rank in Message Type I
MP (I,J)	Input - MP (I, 1) = Message type of same-route outputs in response to messages of type I MP (I, 2) = Message type of other-route outputs in response to messages of type I
MS	Temporary.
NBLOCK	Input - Blocking number
NC	Input - Number of channels
NL	Input - Number of links
NM	Input - Number of message types.
NN	Input - Number of nodes
NO (I, J)	Temporary - NO (I, 1) = Accumulator for Node I mean service time NO (I, 2) = Node I arrival rate. NO (I, 3) = Accumulator for Node I standard deviation of service time.
NON1	Temporary - Utilization of a component
NP	Input - Number of processors

-- Continued --

EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
NP11	Temporary - NP-1.
NR	Input - Number of users.
NUM	Input - Number of user input message types.
N1	Temporary - General purpose pointer.
N2	Temporary - General purpose pointer.
N3	Temporary - General purpose pointer.
N4	Temporary - General purpose pointer.
N5	Temporary - General purpose pointer.
PD (I)	Input - Number of templates for screening messages at a processor I. (0 for TCTs).
POUT (I, J)	Output - Processor outputs: POUT (I, 1) = expected delay at processor I POUT (I, 2) = expected queue length at processor I POUT (I, 3) = traffic rate at processor I POUT (I, 4) = utilization at processor I POUT (I, 5) = capacity at processor I POUT (I, 6) = rank at processor I
RM (I, J, K)	Input - Route cross message type data: RM (I, J, 1) = Initiation rate for user generated messages, proportionality constant for TOS-generated messages. RM (I, J, 2) = Second proportionality constant for TOS-generated messages. RM (I, J, 3) = Probability of hierarchical review deletion for battalion user-generated messages, else zero. RM (I, J, 4) = Probability of hierarchical review alteration for battalion-generated messages, else zero.
RMO (I, J)	Temporary - Rate of each type of message at each node
RNAM (I)	Input - Names of users.

-- Continued --

EXHIBIT 5-10: VARIABLES IN COMPUTE (Concluded)

<u>NAME</u>	<u>USE</u>
ROUT (I, J)	Output - Route outputs: ROUT (I, 1) = Expected delay on route I ROUT (I, 2) = Traffic rate on route I ROUT (I, 3) = Rank of route I
RP (I, J)	Input - Pointers to the first and second links on route I. (Second pointer zero if no second link.)
SM	Temporary - Second moment of the number of transmissions to success on a link to the FEP.
SM1	Temporary - Second moment of the number of transmissions to success on a link.
SM2	Temporary - Second moment of the number of transmissions to success on a link.
SM3	Temporary - Second moment of the number of transmissions to success on a link.
SM4	Temporary - Second moment of the number of transmissions to success on a link.
SUM	Temporary - Scratch pad.
T (I, J, K)	Temporary - Times for various processing activities.
VT	Temporary - Scratch pad.
VM1	Temporary - Second moment of service time.
VM2	Temporary - Second moment of service time.
VM3	Temporary - Second moment of service time.
X	Temporary - Scratch pad.

EXHIBIT 5-11: LOGICAL I/O UNIT ASSIGNMENTS

<u>UNIT</u>	<u>USE</u>
5	Write to operator at terminal.
6	Read from operator at terminal.
7	Read data file.
8	Write output tables.

APPENDIX A: LISTINGS OF TOS SIMULATION PACKAGE PROGRAMS

This appendix contains source listings of the programs which make up the TOS Simulation Package. These programs are named CREATE, DISPLAY, MODIFY, and COMPUTE. The statements within a single routine can be referenced either by a line number or by an internal statement number. Line numbers, shown on the left-hand side of each listing, begin with unity for each program and continue sequentially to the end of the program, each line being assigned a number one integer greater than its predecessor. Internal statement numbers, appearing on the right-hand side of each listing, begin with unit for each routine within a program. Unlike line numbers, internal statement numbers index FORTRAN statements. Thus, there are no internal statement numbers assigned to comment lines, and just one integer assigned to multiline FORTRAN statements.

<PAGE 2>

```

1 DATE:02-05-80,10:40 CWRN:IDSH FILE:CREATE
1

```

LSN

```

59 C      PATTALIONS
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C

45 CONTINUE

DO 60 I = 1, NPEE
  IF (NUN(I) .EQ. (J)) GO TO 60
  III = NUN(I)

DO 55 J = 1, III
  WRITE (6,50) J, RENAM(I)
  FORMAT (' ENTER NAME OF BATTALION ', I1, ' ,PRIGADE ',
  1      A8)
  CALL FREAD(5, 'S:', UNAME(I,J), 8)
  WRITE (6,30) UNAME(I,J)
  CALL FREAD(5, 'S:', UN(I,J), 4)
  CALL CHECK(UN(I,J))
  IF (IV .EQ. 1) GO TO 160
55 CONTINUE

60 CONTINUE

C      OTHER TOS USERS

65 CONTINUE
70 FORMAT (' ENTER NUMBER OF OTHER TOS USERS')
CALL FREAD(5, 'I:', NCH)
IF (NCH .EQ. (0)) GO TO 85

DO 80 I = 1, NCH
  WRITE (6,75) I
  FORMAT (' ENTER NAME OF OTHER TOS USER ', I2)
  CALL FREAD(5, 'S:', CUNAM(I), 9)
  WRITE (6,10) CUNAM(I)
  CALL FREAD(5, 'S:', OU(I), 4)
  CALL CHECK(CU(I))
  IF (IV .EQ. 1 .AND. NCH .GT. 1) GO TO 180
80 CONTINUE

C      COMMUNICATIONS NETWORK

85 CONTINUE
90 FORMAT (' HOW MANY COMMUNICATIONS CHANNELS ARE THERE?')
CALL FREAD(5, 'I:', NC)

C      NAMES OF CHANNELS

DO 100 I = 1, NC
  WRITE (6,95) I
  FORMAT (' ENTER NAME OF CHANNEL ', I2)
  CALL FREAD(5, 'S:', CHAN(I), 8)
  IF (IV .EQ. 1 .AND. NC .GT. 1) GO TO 195
  IF (IV .EQ. 1 .AND. NC .EQ. 1) GO TO 210
100 CONTINUE

C      BATTALION - BRIGADE

```

<PAGE 2>

>>>> MAIN PROGRAM <<<<

///// FILE:CREATE /////

<PAGE 2>

```

117      IF (NDEF .EQ. 0.) GO TO 130
118
119      DO 115 I = 1, NDEF
120        IF (NDN(I) .EQ. 0) GO TO 115
121        ITT = NDN(I)
122
123        PO 110 J = 1, III
124          WRITE (6,105) BNAME(I,J), UDRAME(I)
125          FORMAT (' ENTER NAME OF CHANNEL CONNECTING BATTALION ',
126                AB, ' WITH BRIGADE ', AB)
127          CALL FREAD(5, 'S:', NINCK(I,J), 8)
128          CALL CHECK1(NINCK(I,J), CNAME, NC, LINK(I,J))
129          CONTINUE
130
131      C 115 CONTINUE
132
133      C BRIGADE - FEP
134
135      DO 125 I = 1, NDEF
136        WRITE (6,120) EDRAME(I)
137        FORMAT (' ENTER NAME OF CHANNEL CONNECTING BRIGADE ', AB,
138              ' AND THE FEP')
139        CALL FREAD(5, 'S:', NLINC(I), 8)
140        CALL CHECK1(NLINC(I), CNAME, NC, LINCK(I))
141        CONTINUE
142
143      C 125 CONTINUE
144
145      C OTHER TOS USERS - FEP
146
147      C 130 CONTINUE
148      IF (NOU .EQ. 0) GO TO 250
149
150      DO 140 I = 1, NOU
151        WRITE (6,135) CNAME(I)
152        FORMAT (' ENTER NAME OF CHANNEL CONNECTING ', AB,
153              ' AND THE FEP')
154        CALL FREAD(5, 'S:', NLINCK(I), 8)
155        CALL CHECK1(NLINCK(I), CNAME, NC, LINCK(I))
156        CONTINUE
157
158      C GO TO 255
159
160      C SHORT PROMPTS FOR BRIGADE INFORMATION
161
162      C 145 CONTINUE
163
164      DO 155 I = 2, NDEF
165        WRITE (6,150) I
166        FORMAT (' NAME, TYPE PROCESSOR, # OF BN IN BDE ', I2)
167        CALL FREAD(5, 'S:', DENAMP(I), 8, 'S:', DD(I), 4, 'I:',
168              DEN(I))
169        CALL CHECK(BD(I))
170        CONTINUE
171
172      C GO TO 45
173
174      C SHORT PROMPTS FOR BATTALION INFORMATION
175
176      C 160 CONTINUE

```

[illegible]

LINE	CODE	TEXT	DATE	TIME	USER	IP	CREATE
233		CALL CHECK1(INLNK(I), CNAME, NC, LINK(I))					
234		220 CONTINUE					
235	C						
236		ATTENTION - BRIGADE					
237	C						
238		225 CONTINUE					
239	C						
240		DO 240 I = 1, NDC					
241		IF (NDC(I) .EQ. 0) GO TO 240					
242		JI = NDC(I)					
243	C						
244		DO 245 J = 1, JI					
245		WRITE (6,230) BNAME(I,J), DNAME(I)					
246		FORMAT (' ', A8, 'TO ', A9)					
247		CALL FREAL(5, 'S:', MINK(I,J), R)					
248		CALL CHECK1(MINK(I,J), CNAME, NC, LINK(I,J))					
249		245 CONTINUE					
250	C						
251		240 CONTINUE					
252	C						
253		OTHER TOS USERS - FEP					
254	C						
255		245 CONTINUE					
256		IF (NDC .EQ. 0) GO TO 255					
257	C						
258		DO 255 I = 1, NDC					
259		IF (NDC .EQ. 0 .AND. I .EQ. 1) WRITE (6,115) DNAME(I)					
260		IF (NDC .EQ. 0 .AND. I .EQ. 1) CALL FREAL(5, 'S:',					
261		1 MINK(I), R)					
262		IF (NDC .EQ. 0 .AND. I .EQ. 1) GO TO 250					
263		WRITE (6,215) CNAME(I)					
264		CALL FREAL(5, 'S:', MINK(I), R)					
265		CALL CHECK1(MINK(I), CNAME, NC, LINK(I))					
266		255 CONTINUE					
267	C						
268		***** MAKE CONFIGURATION ASSIGNMENTS *****					
269	C						
270		COUNT NUMBER OF ROUTES					
271	C						
272		260 CONTINUE					
273		NR = NDC + NDC					
274		IF (NDC .EQ. 0) GO TO 270					
275	C						
276		DO 265 I = 1, NDC					
277		NR = NR + NDC(I)					
278		RL = NR					
279		265 CONTINUE					
280	C						
281		270 CONTINUE					
282		NR = NR + 1					
283		NR = NR + NC + 1					
284	C						
285		FORTE AND LINK PCINTERS, OTHER TOS USERS					
286	C						
287		IF (NDC .EQ. 0) GO TO 280					
288	C						
289		DO 275 I = 1, NDC					
290		DNAME(I) = DNAME(I)					
291	C						
292		275 CONTINUE					
293	C						
294		270 CONTINUE					
295	C						
296		275 CONTINUE					
297	C						
298		270 CONTINUE					
299	C						
300		275 CONTINUE					


```

340
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406

ITCT = 0
DO 310 I = 1, NUM
  IF (IPD(I) .EQ. 1) ITCS = 1
  IF (IPD(I) .EQ. 2) ITCT = 2
  IF (ITCT + ITCS .EQ. 3) GO TO 315
310 CONTINUE
C
315 CONTINUE
  NL = NUM
C
C ***** MESSAGE DATA *****
C
  WRITE (6, 120)
320 FORMAT (' ENTER NUMBER OF MSG TYPES AND NUMBER OF USER ',
1 ' INPUT MSG TYPES')
  CALL FREAD(-1, 'DELI')
  CALL FPAR(5, '21', NM, NUM)
  CALL FREAD(-2, 'DELI', '//', 1)
C
C USER INITIATED MESSAGE NAMES
C
  IF (NM .EQ. 0) GO TO 335
  DO 330 I = 1, NUM
    WRITE (6, 125) I
    FORMAT (' ENTER NAME OF USER MSG TYPE ', I2)
    CALL FPAR(5, '5:', NAME(I), 8)
    IF (IV .EQ. 1 .AND. NUM .GT. 1) GO TO 385
    IF (IV .EQ. 1 .AND. NUM .EQ. 1 .AND. NM .GT. NUM)
      GO TO 400
330 CONTINUE
C
C TOS GENERATED MESSAGE NAMES
C
335 CONTINUE
  IF (NM .EQ. NM) GO TO 350
  JJ = NUM + 1
  DO 340 I = JJ, NM
    JJ = I - NUM
    WRITE (6, 140) IJ
    FORMAT (' ENTER NAME OF TOS MSG TYPE ', I2)
    CALL FPAR(5, '5:', NAME(I), 8)
    IF (IV .EQ. 1 .AND. NUM .EQ. 0 .AND. NM .GT. 1)
      GO TO 410
345 CONTINUE
C
C USER MESSAGE DATA
C
  PEAN LENGTHS
C
350 CONTINUE
  CALL FPAR(-1, 'DELI')
  IF (NM .EQ. 0) GO TO 375
  DO 370 I = 1, NUM
    WRITE (6, 155) NAME(I)

```

<PAGE 7>

ISN

DATE:02-05-80, 14:40 CUNPR:IDSU FILE:CREATE

<PAGE 7>

>>>> MAIN PROGRAM <<<<

///// FILE:CREATE /////

<PAGE 7>

[illegible]

```

465 CALL EPFAD(5, 'S:', NMAN(1), 0)
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

CALL EPFAD(5, 'S:', NMAN(1), 0)
 CONTINUE
 GO TO 150
 SHORT PROMPTS FOR MESSAGE DATA
 USER MESSAGES
 CONTINUE
 DO 410 I = 2, NUM
 WRITE (6,425) NMAN(I)
 FORMAT (' FOR MSG ', AR, ' LENGTH, SAME AND OTHER ROUTE')
 CALL EPFAD(5, 'R:', MD(I,1), 'S:', MP(I,1), 0, 'S:',
 1 MP(I,2), 8)
 CALL CHECK2(MP(I,1), NM, NMAN, MP(I,1))
 CALL CHECK2(MD(I,2), NM, NMAN, MP(I,2))
 CONTINUE
 405 MESSAGES
 IF (NUM - EQ. NM) GO TO 450
 CONTINUE
 WRITE (6,355) NMAN(NUM + 1)
 CALL EPFAD(5, 'R:', MD(NUM + 1,1))
 MP(NUM + 1,1) = 0
 MP(NUM + 1,2) = 0
 IF (NUM + 1 - EQ. NM) GO TO 450
 33 = NUM + 2
 DO 445 I = 33, NM
 WRITE (6,440) NMAN(I)
 FORMAT (' MSG ', AR)
 CALL EPFAD(5, 'R:', MD(I,1))
 MP(I,1) = 0
 MP(I,2) = 0
 CONTINUE
 ***** ROUTE X MESSAGE DATA *****
 CHECK TO SEE IF ARRAY IS TO BE INPUT
 CONTINUE
 WRITE (6,455)
 455 FORMAT (' IF YOU WANT TO REAT THE ROUTE CROSS MESSAGE ARRAY',
 1 ' INTO THE OUTPUT FILE LATER, ENTER A "YES"')
 CALL EPFAD(5, 'L:', LEG)
 IF (LEG - EQ. 1) WRITE (6,460)
 460 FORMAT (' THE OUTPUT FILE WILL CONTAIN A LINE OF ASTERISKS',
 1 ' WHERE THE ROUTE CROSS MESSAGE ARRAY SHOULD BE INSERTED')
 IF (LEG - EQ. 1) GO TO 545
 IF (IV - EQ. 1) GO TO 560
 ENTER ARRAY
 3N = NREF + NCU

<PAGE 9>

I DATE:02-05-80,14:40 CMBEE:IDSH FILE:CREATE I

LSN

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

<PAGE 9>

>>>> MAIN PROGRAM <<<<

///// FILE:CREATE /////

<PAGE 9>

```

C 315
C 316
C 317
C 318
C 319
C 320
C 321
C 322
C 323
C 324
C 325
C 326
C 327
C 328
C 329
C 330
C 331
C 332
C 333
C 334
C 335
C 336
C 337
C 338
C 339
C 340
C 341
C 342
C 343
C 344
C 345
C 346
C 347
C 348
C 349
C 350
C 351
C 352
C 353
C 354
C 355
C 356
C 357
C 358
C 359
C 360
C 361
C 362
C 363
C 364
C 365

C 400
C 401
C 402
C 403
C 404
C 405
C 406
C 407
C 408
C 409
C 410
C 411
C 412
C 413
C 414
C 415
C 416
C 417
C 418
C 419
C 420
C 421
C 422
C 423
C 424
C 425
C 426
C 427
C 428
C 429
C 430
C 431
C 432
C 433
C 434
C 435
C 436
C 437
C 438
C 439
C 440
C 441
C 442
C 443
C 444
C 445
C 446
C 447
C 448
C 449
C 450
C 451
C 452
C 453
C 454
C 455
C 456
C 457
C 458
C 459
C 460
C 461
C 462
C 463
C 464
C 465
C 466
C 467
C 468
C 469
C 470
C 471
C 472
C 473
C 474
C 475
C 476
C 477
C 478
C 479
C 480
C 481
C 482
C 483
C 484
C 485
C 486
C 487
C 488
C 489
C 490
C 491
C 492
C 493
C 494
C 495
C 496
C 497
C 498
C 499
C 500
C 501
C 502
C 503
C 504
C 505
C 506
C 507
C 508
C 509
C 510
C 511
C 512
C 513
C 514
C 515
C 516
C 517
C 518
C 519
C 520
C 521
C 522
C 523
C 524
C 525
C 526
C 527
C 528
C 529
C 530
C 531
C 532
C 533
C 534
C 535
C 536
C 537
C 538
C 539
C 540
C 541
C 542
C 543
C 544
C 545
C 546
C 547
C 548
C 549
C 550
C 551
C 552
C 553
C 554
C 555
C 556
C 557
C 558
C 559
C 560
C 561
C 562
C 563
C 564
C 565
C 566
C 567
C 568
C 569
C 570
C 571
C 572
C 573
C 574
C 575
C 576
C 577
C 578
C 579
C 580
C 581
C 582
C 583
C 584
C 585
C 586
C 587
C 588
C 589
C 590
C 591
C 592
C 593
C 594
C 595
C 596
C 597
C 598
C 599
C 600
C 601
C 602
C 603
C 604
C 605
C 606
C 607
C 608
C 609
C 610
C 611
C 612
C 613
C 614
C 615
C 616
C 617
C 618
C 619
C 620
C 621
C 622
C 623
C 624
C 625
C 626
C 627
C 628
C 629
C 630
C 631
C 632
C 633
C 634
C 635
C 636
C 637
C 638
C 639
C 640
C 641
C 642
C 643
C 644
C 645
C 646
C 647
C 648
C 649
C 650
C 651
C 652
C 653
C 654
C 655
C 656
C 657
C 658
C 659
C 660
C 661
C 662
C 663
C 664
C 665
C 666
C 667
C 668
C 669
C 670
C 671
C 672
C 673
C 674
C 675
C 676
C 677
C 678
C 679
C 680
C 681
C 682
C 683
C 684
C 685
C 686
C 687
C 688
C 689
C 690
C 691
C 692
C 693
C 694
C 695
C 696
C 697
C 698
C 699
C 700
C 701
C 702
C 703
C 704
C 705
C 706
C 707
C 708
C 709
C 710
C 711
C 712
C 713
C 714
C 715
C 716
C 717
C 718
C 719
C 720
C 721
C 722
C 723
C 724
C 725
C 726
C 727
C 728
C 729
C 730
C 731
C 732
C 733
C 734
C 735
C 736
C 737
C 738
C 739
C 740
C 741
C 742
C 743
C 744
C 745
C 746
C 747
C 748
C 749
C 750
C 751
C 752
C 753
C 754
C 755
C 756
C 757
C 758
C 759
C 760
C 761
C 762
C 763
C 764
C 765
C 766
C 767
C 768
C 769
C 770
C 771
C 772
C 773
C 774
C 775
C 776
C 777
C 778
C 779
C 780
C 781
C 782
C 783
C 784
C 785
C 786
C 787
C 788
C 789
C 790
C 791
C 792
C 793
C 794
C 795
C 796
C 797
C 798
C 799
C 800
C 801
C 802
C 803
C 804
C 805
C 806
C 807
C 808
C 809
C 810
C 811
C 812
C 813
C 814
C 815
C 816
C 817
C 818
C 819
C 820
C 821
C 822
C 823
C 824
C 825
C 826
C 827
C 828
C 829
C 830
C 831
C 832
C 833
C 834
C 835
C 836
C 837
C 838
C 839
C 840
C 841
C 842
C 843
C 844
C 845
C 846
C 847
C 848
C 849
C 850
C 851
C 852
C 853
C 854
C 855
C 856
C 857
C 858
C 859
C 860
C 861
C 862
C 863
C 864
C 865
C 866
C 867
C 868
C 869
C 870
C 871
C 872
C 873
C 874
C 875
C 876
C 877
C 878
C 879
C 880
C 881
C 882
C 883
C 884
C 885
C 886
C 887
C 888
C 889
C 890
C 891
C 892
C 893
C 894
C 895
C 896
C 897
C 898
C 899
C 900
C 901
C 902
C 903
C 904
C 905
C 906
C 907
C 908
C 909
C 910
C 911
C 912
C 913
C 914
C 915
C 916
C 917
C 918
C 919
C 920
C 921
C 922
C 923
C 924
C 925
C 926
C 927
C 928
C 929
C 930
C 931
C 932
C 933
C 934
C 935
C 936
C 937
C 938
C 939
C 940
C 941
C 942
C 943
C 944
C 945
C 946
C 947
C 948
C 949
C 950
C 951
C 952
C 953
C 954
C 955
C 956
C 957
C 958
C 959
C 960
C 961
C 962
C 963
C 964
C 965
C 966
C 967
C 968
C 969
C 970
C 971
C 972
C 973
C 974
C 975
C 976
C 977
C 978
C 979
C 980
C 981
C 982
C 983
C 984
C 985
C 986
C 987
C 988
C 989
C 990
C 991
C 992
C 993
C 994
C 995
C 996
C 997
C 998
C 999
C 1000

C 400
C 401
C 402
C 403
C 404
C 405
C 406
C 407
C 408
C 409
C 410
C 411
C 412
C 413
C 414
C 415
C 416
C 417
C 418
C 419
C 420
C 421
C 422
C 423
C 424
C 425
C 426
C 427
C 428
C 429

```

```

C 11 PM(1,1,2) = 0
C 12 RM(1,1,3) = 0
C 13 RM(1,1,4) = 0
C 14
C 15 SHORT PROMPTS FOR USER INITIATION RATES AT FIRST USER
C 16
C 17 IF (NUM .EQ. 1) GO TO 515
C 18
C 19 DO 510 J = 2, NUM
C 20 WRITE (6,505) MMAM(J), RMAM(J)
C 21 FORMAT (' ', A8, ' AT ', A6)
C 22 CALL FPRAT(5, 'R:', RM(1,J,1))
C 23 RM(1,J,2) = 0
C 24 RM(1,J,3) = 0
C 25 RM(1,J,4) = 0
C 26 510 CONTINUE
C 27
C 28 INITIATION RATES AT OTHER NODES
C 29
C 30 515 CONTINUE
C 31 IF = NCD * NRCF
C 32 IF (II .EQ. 1) GO TO 510
C 33
C 34 DO 525 I = 2, II
C 35
C 36 DO 520 J = 1, NUM
C 37 WRITE (6,505) MMAM(J), RMAM(J)
C 38 CALL FPRAT(5, 'R:', RM(I,J,1))
C 39 RM(I,J,2) = 0
C 40 RM(I,J,3) = 0
C 41 RM(I,J,4) = 0
C 42 520 CONTINUE
C 43
C 44 525 CONTINUE
C 45
C 46 INITIATION RATES & REVIEW PROBABILITIES FOR MESSAGES
C 47 FROM BATTALIONS TO BRIGADES
C 48
C 49 530 CONTINUE
C 50 IF (II .EQ. MP) GO TO 560
C 51 II = II + 1
C 52 WRITE (6,505) MMAM(II), RMAM(II)
C 53 CALL FPRAT(5, 'R:', RM(II,1,1))
C 54 RM(II,1,2) = 0
C 55
C 56 FIRST LONG PROMPT FOR REVIEW PROBABILITIES
C 57
C 58 WRITE (6,470) MMAM(II), RMAM(II)
C 59 CALL FPRAT(5, 'R:', RM(II,1,1))
C 60 IF (RM(II,1,3) .GT. 1.-) .OR. RM(II,1,4) .LT. 0.0) CALL CHECK3(RM(
C 61 II,1,1))
C 62
C 63 WRITE (6,475) MMAM(II), RMAM(II)
C 64 CALL FPRAT(5, 'R:', RM(II,1,4))
C 65 IF (RM(II,1,4) .GT. 1.- .OR. RM(II,1,4) .LT. 0.0) CALL CHECK3(RM(
C 66 II,1,4))
C 67
C 68 SHORT PROMPTS FOR FIRST PATTALION
C 69
C 70

```

<PAGE 11>

>>>> MAIN PROGRAM <<<<

/// FILE:CREATE ///

<PAGE 11>

```

C
IP (NMN ,FO, 1) GO TO 545
DO 540 J = 2, NMN
WRITE (6,535) NMN(J), RNAM(11)
535 FORMAT (' RATE, PROB OF DEFPTE, ALTRF UP MSG ', M0, ' AT ',
AP)
CALL FRAC(5, 'JR:', RN(11,J,1), RN(11,J,3), RN(11,J,4))
IF (RN(11,J,1) -GT. 1.0 -OR. RN(11,J,3) -LT. 0.0)
CALL CHECK3(FM(11,J,3))
1 CALL (RN(11,J,4) -GT. 1.0 -OR. RN(11,J,4) -LT. 0.0)
1 CALL CHECK3(RN(11,J,4))
RN(11,J,2) = 0
540 CONTINUE
C
SHORT PROMPTS FOR OTHER BATTALIONS
C
545 CONTINUE
IF (11 -FO. NR) GO TO 560
11 = 11 + 1
C
DO 555 I = 11, NM
C
TO 550 J = 1, NM
WRITE (6,535) NMN(J), RNAM(11)
CALL FRAC(5, 'JR:', RN(1,J,1), RN(1,J,3), RN(1,J,4))
1P (RN(1,J,1) -GT. 1.0 -OR. RN(1,J,3) -LT. 0.0)
1 CALL CHECK3(RN(1,J,3))
1P (RN(1,J,4) -GT. 1.0 -OR. RN(1,J,4) -LT. 0.0)
1 CALL CHECK3(RN(1,J,4))
RN(1,J,2) = 0
550 CONTINUE
C
555 CONTINUE
C
560 CONTINUE
C
TOS GENERATED MESSAGES
C
FIRST LONG PROMPT FOR 1ST TOS MESSAGE PROPORTIONALITY
CONSTANTS, 1ST USER
C
IF (NMN -FO. NM) GO TO 585
JJ = NMN + 1
WRITE (6,445) NMN(JJ), RNAM(11)
CALL FRAC(5, '2R:', RN(1,JJ,1), RN(1,JJ,2))
RN(1,JJ,3) = 0
RN(1,JJ,4) = 0
C
SHORT PROMPTS FOR PROPORTIONALITY CONSTANTS, 1ST USER
C
IF (JJ -FO. NM) GO TO 570
JJ = NMN + 2
C
DO 565 J = 1,J, NM
WRITE (6,505) RNAM(J), RNAM(11)
CALL FRAC(5, '2R:', RN(1,J,1), RN(1,J,2))
RN(1,J,3) = 0
RN(1,J,4) = 0
C
600
610
620
630
640
650
660
670
680
690
700
710
720
730
740
750
760
770
780
790
800
810
820
830
840
850
860
870
880
890
900
910
920
930
940
950
960
970
980
990

```

<PAGE 11>

```

697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754

565 CONTINUE
C
C      SHORT INCRPTS FOR OTHER NODES
C
570 CONTINUE
IF (NF.FO. 1) GO TO 595
C
DO 580 I = 2, NH
C
    IO 575 J = JJ, NH
    WRITE (6,505) NHAM(J), NHAM(I)
    CALL FREQ(5, '2R:', NH(I,J,1), NH(I,J,2))
    NH(I,J,3) = 0
    NH(I,J,4) = 0
575 CONTINUE
C
580 CONTINUE
C
C      ***** ENGINEERING DATA *****
C
C      TASK ACCESS TIMES
C
585 CONTINUE
WRITE (6,590)
590 FORMAT (' ENTER MEAN DATA DISK ACCESS TIME IN MILLISECONDS')
CALL FREQ(5, 'R:', DD(2))
WRITE (6,595)
595 FORMAT (' ENTER MEAN MESSAGE DISK ACCESS TIME IN MILLISECONDS')
CALL FREQ(5, 'R:', DD(1))
IF (TCS.NP. 1) GO TO 625
C
C      TCS DATA
C
WRITE (6,600)
600 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCS TO ORIGINATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(1,1), GPD(1,2))
IF (IV.FO. 1) GO TO 605
WRITE (6,605)
605 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCS TO SEND, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(2,1), GPD(2,2))
610 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCS TO RECEIVE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(3,1), GPD(3,2))
WRITE (6,615)
615 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCS TO TEMPLATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(4,1), GPD(4,2))
WRITE (6,620)
620 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCS TO TERMINATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(5,1), GPD(5,2))
GO TO 635
625 CONTINUE
C
DO 630 I = 1, 5

```

<PAGE 11>

///// FILE:CREATE /////

>>>> MAIN PROGRAM <<<<

<PAGE 11>

```

<PAGE 14>
1 DATE:02-05-80, 14:40 CREF:IDSH FILE:CREATE
1
15N
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
<PAGE 14>

610 CONTINUE
615 CONTINUE
C
C TCT DATA
C
IF (TCT.NF. 2) GO TO 660
WRITE (6,600)
600 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO ORIGINATE, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(7,1), GPD(7,2))
IF (IV.FO. 1) GO TO 715
WRITE (6,605)
605 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO SEND, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(8,1), GPD(8,2))
WRITE (6,650)
650 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO RECEIVE, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(9,1), GPD(9,2))
WRITE (6,655)
655 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO TERMINATE, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(10,1), GPD(10,2))
GO TO 670
660 CONTINUE
C
DO 665 I = 1, 4
GPD(I + 6,1) = 0.
GPD(I + 6,2) = 0.
665 CONTINUE
C
IF (IV.FO. 1) GO TO 720
670 CONTINUE
C
C FFP DATA
C
WRITE (6,675)
675 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT THE PER TO INITIATE, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(11,1), GPD(11,2))
WRITE (6,680)
680 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT THE PER TO TERMINATE, ',
1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
CALL FREAD(5, '2R:', GPD(6,1), GPD(6,2))
GO TO 730
C
C SHORT PROMPTS FOR TCS DATA
C
685 CONTINUE
WRITE (6,690)
690 FORMAT (' TO SEND')
CALL FREAD(5, '2R:', GPD(2,1), GPD(2,2))
WRITE (6,695)
695 FORMAT (' TO RECEIVE')
CALL FREAD(5, '2R:', GPD(3,1), GPD(3,2))

```

>>>> MAIN PROGRAM <<<<

///// FILE:CREATE /////

<PAGE 14>

<PAGE 16>

DATE:02-05-80,14:40 CMMER:IDSH FILE:CREATE

15N

```

871 C MESSAGE REQUIREMENTS
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C

      DO 790 I = 1, NM
      WRITE (6,775) NMAM(I)
      FORMAT (' ENTER DEP PROCESSING IN MILLISECONDS TIME FOR MSG ',
      1      'A')
      CALL FPEAT(5, 'R:', MD(I,2))
      WRITE (6,780) NMAM(I)
      FORMAT (' ENTER MSG DISK READS/Writes FOR MSG ', A9)
      CALL FPEAT(5, 'R:', MD(I,3))
      WRITE (6,785) NMAM(I)
      FORMAT (' ENTER DATA DISK READS/Writes FOR MSG ', A9)
      CALL FPEAT(5, 'R:', MD(I,4))
      IF (IIV.EC. 1 .AND. NM .GT. 1) GO TO 795
790 CONTINUE
C
      GO TO 910
C
C      SHORT PROMPTS FOR MESSAGE REQUIREMENTS
C
795 CONTINUE
C
      DO 805 I = 2, NM
      WRITE (6,800) NMAM(I)
      FORMAT (' DEP TIME, MSG DISK R/W, DATA DISK R/W, FOR MSG ',
      1      'A')
      CALL FPEAT(5, 'R:', MD(I,2), MD(I,3), MD(I,4))
805 CONTINUE
C
      DO 810 I = 1, NM
      IF (NM.I.EC. 0) GO TO 820
      DO 815 I = 1, NM
      ED(I) = 0
815 CONTINUE
C
820 CONTINUE
      IF (MPOF.EC. 0) GO TO 835
      JJ = NM + 1
      JK = EC0 + NMDF
C
      DO 830 I = JJ, JK
      WRITE (6,825) NMAM(I)
      FORMAT (' ENTER NUMBER OF TEMPLATES AT ', A9)
      CALL FPEAT(5, 'I:', ED(I))
      IF (IIV.EC. 1 .AND. NMDF .GT. 1) GO TO 845
830 CONTINUE
C
      P15 WRITE (6,840)
      840 FORMAT (' ENTER NUMBER OF TEMPLATES AT THE PEP')
      CALL FPEAT(5, 'I:', ED(NR + 1))
      GO TO 870
C

```

<PAGE 16>

>>>> MAIN PROGRAM <<<<

///// FILE:CREATE /////

<PAGE 16>

```

<PAGE 17>
1 DATE:02-05-80,14:40 CUNER:TDSH IIL:CNFATE
ISN
SHORT PROMPTS FOR # OF TEMPLATES
895 CONTINUE
IP (NPCE -FO. 1) GO TO 860
JJ = JJ + 1
DO 955 I = JJ, JK
  WRITE (6,950) JNAN(I)
  FORMAT (' AT ', A3)
  CALL FREQ(5, '1:', PL(NR + 1))
  CALL FREQ(5, '1:', PL(I))
955 CONTINUE
960 CONTINUE
  WRITE (6,865)
  FORMAT (' AT FFF ')
  CALL FREQ(5, '1:', PL(NR + 1))
  CALL FREQ(5, '1:', PL(I))
  IP (NCH + NDEF -EO. NR) GO TO 890
  JK = JK + 1
DO 975 I = JK, NR
  PR(I) = 0
975 CONTINUE
895 ***** ENVIRONMENTAL DATA *****
980 CONTINUE
  DO 895 I = 1, NC
    WRITE (6,895) CNA3(I)
    FORMAT (' ENTER PROPORTION OF TIME CHANNEL ', A3, ' IS ',
      ' PING USED FOR VOICE TRANSMISSIONS')
    CALL FREQ(5, 'R:', CL(I, 1))
    IF (C0(I, 1) -GT. 1.0 .OR. C0(I, 3) -LT. 0.0) CALL CHECK4(C0(I,
      1))
    IP (LV -FO. 1 .AND. NC -GT. 1) GO TO 895
990 CONTINUE
  GO TO 910
SHORT PROMPTS FOR VOICE COMPETITION INFO
895 CONTINUE
  DO 905 I = 2, NC
    WRITE (6,900) CNAH(I)
    FORMAT (' CHANNEL ', A3)
    CALL FREQ(5, 'R:', C0(I, 1))
    IF (C0(I, 1) -GT. 1.0 .OR. C0(I, 3) -LT. 0.0) CALL CHECK4(C0(I,
      1))
905 CONTINUE
  ERROR RATES: TO 6 FROM FOP
910 CONTINUE
  JN = NCH + NDEF

```

<PAGE 17>

///// PEEP:CREATE /////

>>>> MAIN PROGRAM <<<<

<PAGE 17>

MSI
PAGE 18

DATE: 02-05-90, 14:40 OWNER: USN FILE: CREATP

<PAGE 14>

```

907 DO 925 I = 1, JH
908   WRITE (6,915) RNAME(I)
909   FORMAT (' ENTER EIT ERROR RATE FROM ', A9, ' TO ',
910   ' THE FEE')
911   1
912   CALL FREAD(5, 'R:', LD(I,1))
913   WRITE (6,920) RNAME(I)
914   FORMAT (' ENTER BIT ERROR RATE FROM THE PER TO ', A9)
915   CALL FREAD(5, 'R:', LD(I,2))
916   IF (IV - FC - 1 .AND. NR .GT. 1) GO TO 950
917   925 CONTINUE
918 C
919   920
921   921
922   922
923   923
924   924
925   925
926   926
927   927
928   928
929   929
930   930
931   931
932   932
933   933
934   934
935   935
936   936
937   937
938   938
939   939
940   940
941   941
942   942
943   943
944   944
945   945
946   946
947   947
948   948
949   949
950   950
951   951
952   952
953   953
954   954
955   955
956   956
957   957
958   958
959   959
960   960
961   961
962   962
963   963
964   964
965   965
966   966
967   967
968   968
969   969
970   970
971   971
972   972
973   973
974   974
975   975
976   976
977   977
978   978
979   979
980   980
981   981
982   982
983   983
984   984
985   985
986   986
987   987
988   988
989   989
990   990
991   991
992   992
993   993
994   994
995   995
996   996
997   997
998   998
999   999
1000  1000

```

<PAGE 10>

>>>> MAIN PROGRAM <<<<

////// FILE:CRFATE /////

<PAGE 1A>

```

1045 2      LR, GPD, IICS, IICU)
1046 C
1047 C      ***** WRITE OUTPUT FILE HERE *****
1048 C
1049 C      WRITE (6,900)
1050 900 FORMAT (' IF YOU WANT TO CREATE THE OUTPUT FILE ENTER, * A *1000)
1051 CALL FREAD(5, '1:', IWF)
1052 IF (IWF .NE. 0) GO TO 1015
1053 C
1054 C      INDEXING INFORMATION AND COUNTS
1055 C
1056 C      WRITE (2,905) NR, NP, NW, NW, NC, NP, NI, NOW, NDOZ, IFG
1057 905 FORMAT (10(I1,1X))
1058 C
1059 C      ROUTE INFORMATION
1060 C
1061 C      DO 995 I = 1, NR
1062     WRITE (2,900) RMAN(I), RE(I,1), RP(I,2), IPD(I)
1063     FORMAT (A9, 3(I1,1))
1064 995 CONTINUE
1065 C
1066 C      MESSAGE INFORMATION
1067 C
1068 C      DO 1005 I = 1, NW
1069     WRITE (2,1000) MNAME(I), MP(I,1), MP(I,2), MD(I,1), MD(I,2),
1070     PD(I,3), PD(I,4)
1071     FORMAT (A4, 2(I1,1), 4(I1,F9.3))
1072 1005 CONTINUE
1073 C
1074 C      CHANNEL INFORMATION
1075 C
1076 C      DO 1015 I = 1, NC
1077     WRITE (2,1010) CNAME(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)
1078     FORMAT (A6, 4(I1,F9.3))
1079 1015 CONTINUE
1080 C
1081 C      NUMBER OF ITERATES
1082 C
1083 C      DO 1025 I = 1, NP
1084     WRITE (2,1020) ID(I)
1085     FORMAT (2X, I1)
1086 1025 CONTINUE
1087 C
1088 C      LINK DATA
1089 C
1090 C      DO 1035 I = 1, NI
1091     WRITE (2,1030) IP(I,1), IP(I,2), LP(I,1), LP(I,2), LD(I,2)
1092     FORMAT (3(I1,1X), 2(E12.6,1X))
1093 1035 CONTINUE
1094 C
1095 C      ROUTE X MESSAGE ARRAY
1096 C
1097 C      IF (IFG .EQ. 1) WRITE (2,1040)
1098     FORMAT (' ', 50(' '))
1099 1040 IF (IFG .EQ. 1) GO TO 1060
1100 C
1101 C      DO 1055 I = 1, NW
1102 1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112

```

<PAGE 17>

<PAGE 19>

DATE:02-05-80,10:40 CNAME:IDSH FILE:CREATE

ISM

<PAGE 19>

<PAGE 19>

>>>> DATA PROGRAM <<<<

///// FILE:CREATE /////

```

1103      DO 1050 J = 1, NM
1104          WRITE (2,1045) RM(I,J,1), P*(I,J,2), RM(I,J,3),
1105              RM(I,J,4)
1106          FORMAT (4(F9.3,1X))
1107      CONTINUE
1108  C
1109  C
1110  C      RISK ACCESS TIMES
1111  C
1112  C
1113  C
1114  C      DO 1060 CONTINUE
1115  C          WRITE (2,1065) DD(1), DD(2)
1116  C          FORMAT (2(F9.3,1X))
1117  C
1118  C      GENERAL IPROCESSOR DATA
1119  C
1120  C          DO 1070 I = 1, 11
1121  C              WRITE (2,1065) GPD(I,1), GPD(I,2)
1122  C          CONTINUE
1123  C
1124  C      1075 CONTINUE
1125  C          END

```

<PAGE 20>

```

1  DATE: 02-05-90, 14:40  CYNSE: 1DSH  FILE: CREATP

```

ISM

702

703

704

705

706

707

708

709

710

711

712

713

714

<PAGE 20>

>>>> MAIN PROGRAM <<<<

///// FILE: CREATP /////

<PAGE 20>

<PAGE 21>

1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144

<PAGE 21>

1 DATE:02-05-00, 14:40 CMMR:TOSH FILE:CRPATF
1

ISM

***** SUBROUTINES *****

THIS SUBROUTINE CHECKS THE NAME OF THE PROCESSOR TYPE,
THE CORRECT NAME IS RETURNED AS VARIABLE "TYPE"

SUBROUTINE CHECK(TYPE)

10 CONTINUE
INTEGER TYPE, TCT /'TCT ', TCS /'TCS '
IF (TYPE .EQ. TCT) RETURN
IF (TYPE .EQ. TCS) RETURN
WRITE (6,15) TYPE
IS FORNAT (' THERE IS NO PROCESSOR NAMED ', A0, 'TRY AGAIN')
CALL FPEAD(5, 'S:', TYPE, 4)
GO TO 10
END

1
2
3
4
5
6
7
8
9
10

<PAGE 21>

///// FILE:CREATE /////

>>>> SUBROUTINE CUPCK <<<<

<PAGE 21>

<PAGE 22>
15N

I DATE: 07-05-80, 14:40 GMRB: 1950 FILE: CREATE I

```

1185 C
1186 C
1187 C
1188 C
1189 C
1190 C
1191 C
1192 C
1193 C
1194 C
1195 C
1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C
1213 C
1214 C
1215 C
1216 C
1217 C
1218 C
1219 C
1220 C
1221 C
1222 C
1223 C
1224 C
1225 C
1226 C
1227 C
1228 C
1229 C
1230 C
1231 C
1232 C
1233 C
1234 C
1235 C
1236 C
1237 C
1238 C
1239 C
1240 C
1241 C
1242 C
1243 C
1244 C
1245 C
1246 C
1247 C
1248 C
1249 C
1250 C
1251 C
1252 C
1253 C
1254 C
1255 C
1256 C
1257 C
1258 C
1259 C
1260 C
1261 C
1262 C
1263 C
1264 C
1265 C
1266 C
1267 C

```

THIS SUBROUTINE CHECKS THE NAME OF THE CHANNEL "NIN" AND
 RETURNS THE NUMBER OF THAT CHANNEL IN VARIABLE "NIN1"

SUBROUTINE CHECK1(NIN, CNAM, NC, NIN1)
 REAL*8 NIN, CNAM(10)
 10 CONTINUE
 DO 15 I = 1, NC
 IF (NIN - CNAM(I)) GO TO 15
 NIN1 = I
 RETURN
 15 CONTINUE
 WRITE (6,20) NIN
 20 PRINT (' THERE IS NO CHANNEL NAMED ', AD, ' TRY AGAIN')
 CALL FREAD('S', 'S:', NIN, 8)
 GO TO 10
 END

<PAGE 22>

>>>> SUBROUTINE CHECK1 <<<<

///// FILE:CREATE /////

<PAGE 22>

<PAGE 23>

15N

DATE:07-05-00,10:40 CUMER:105H FILE:CREATF

<PAGE 24>

1164
1165
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195

THIS SUBROUTINE CHECKS THE NAME OF THE MESSAGE "HOUT" AND
RETURNS THE NUMBER OF THAT MESSAGE IN VARIABLE "MOUT1"

SUBROUTINE CHECK2(MCUT, NM, MNAM, MOUT1)

REAL*8 MOUT, MNAM(50), ZERO
DATA ZERO /0.0/
10 CONTINUE
IF (MCUT.EQ. ZERO) GO TO 25

DO 15 I = 1, NM
IF (MNAM(I).NE. HOUT) GO TO 15
MOUT1 = I
RETURN
15 CONTINUE

WRITE (6,20) MCUT
20 FORMAT (' THERE IS NO MSG NAMED ', A8, 'TRY AGAIN')
CALL FREAD(5, 'S:', HOUT, 0)
GO TO 10
25 CONTINUE
MOUT1 = 0
RETURN
END

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

<PAGE 23>

>>>> SUBROUTINE CHECK2 <<<<

///// FILE:CREATF /////

<PAGE 24>

```

1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C

      THIS SUBROUTINE REPLACES PROBABILITIES WHICH HAVE
      BEEN ENTERED OUT OF THE RANGE 0-1 AND RETURNS THE
      EXCEED VALUE IN VARIABLE "PROB".

      SUBROUTINE CHECK1(PRCB)
      10 CONTINUE
      15 WRITE (6,15)
      15 FORMAT (' THE PROBABILITY IS OUT OF THE RANGE 0-1, TRY AGAIN')
      CALL PFAD(5, 'R:', PRCH)
      IF (PROB .LT. 1.0 .AND. PRCB .GT. 0.0) RETURN
      GO TO 10
      END

```

1
2
3
4
5
6
7
8

<PAGE 24>

>>>> SUBROUTINE CHECK1 <<<<

///// FILE:CREATE /////

<PAGE 25>

1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229

1 DATE:02-05-80, 14:40 CTYPE:DSH FILE:CREATF
ISM

<PAGE 25>

THIS SUBROUTINE REPLACES PROPORTIONS WHICH HAVE
BEEN ENTERED OUT OF THE RANGE 0-1 AND RETURNS THE
PROPER VALUE IN VARIABLE "PROP".

SUBROUTINE CHECK4(PRCF)

10 CONTINUE

WRITE (6,15)

15 FORMAT (' THE PROPORTION IS OUT OF THE RANGE 0-1, TRY AGAIN!')

CALL READ(5, 'R:', PROP)

IF (PROP .LT. 0.0 .AND. PROP .GT. 0.0) RETURN

GO TO 10

END

1
2
3
4
5
6
7
8

<PAGE 25>

//// FILE:CREATF ////

>>>> SUBROUTINE CHECK4 <<<<

<PAGE 25>

MS1
PAGE 262

DATE: 02-05-00, 14:40 COUNTER: IDSH FILE: CHATE

THIS SUBROUTINE DISPLAYS THE OUTPUT FILE IN AN EASY TO READ FORMAT. THIS SUBROUTINE IS ALSO SOMEONE AS PART OF PROGRAM DISPLAY.

[illegible]

<PAGE 26>

>>>> SUBROUTINE DISP <<<<

<PAGE 26>

<PAGE 27>

 1 DATE:02-05-80,14:40 USER:IOSH FILE:CREATE

<PAGE 27>

158

```

1298 DO 70 I = 1, NR
1299   IF (LP(I,1) .NE. 1) WRITE (6,115)
1300   ISPACE = IP(I,3)
1301   IF (IP(I,1) .EQ. 1) GO TO 60
1302   WRITE (6,115) READ(I), CHAM(LP(I,2)), RNAME(LP(I,3))
1303   FORMAT (22X, A6, 16X, A9, 18X, A9, 11X, 'TCT')
1304   GO TO 70
1305 60 WRITE (6,115) RNAME(I), CHAM(LP(I,2)), RNAME(LP(I,3))
1306   FORMAT (22X, A6, 16X, A9, 18X, A9, 17X, 'TCS')
1307 70 CONTINUE
1308 C
1309 75 CONTINUE
1310 C
1311 C
1312 C
1313 C
1314 C
1315 C
1316 C
1317 C
1318 C
1319 C
1320 C
1321 C
1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C
1335 C
1336 C
1337 C
1338 C
1339 C
1340 C
1341 C
1342 C
1343 C
1344 C
1345 C
1346 C
1347 C
1348 C
1349 C
1350 C
1351 C
1352 C
1353 C
1354 C
1355 C
1356 C
1357 C
1358 C
1359 C
1360 C
1361 C
1362 C
1363 C
1364 C
1365 C
1366 C
1367 C
1368 C
1369 C
1370 C
1371 C
1372 C
1373 C
1374 C
1375 C
1376 C
1377 C
1378 C
1379 C
1380 C
1381 C
1382 C
1383 C
1384 C
1385 C
1386 C
1387 C
1388 C
1389 C
1390 C
1391 C
1392 C
1393 C
1394 C
1395 C
1396 C
1397 C
1398 C
1399 C
1400 C
1401 C
1402 C
1403 C
1404 C
1405 C
1406 C
1407 C
1408 C
1409 C
1410 C
1411 C
1412 C
1413 C
1414 C
1415 C
1416 C
1417 C
1418 C
1419 C
1420 C
1421 C
1422 C
1423 C
1424 C
1425 C
1426 C
1427 C
1428 C
1429 C
1430 C
1431 C
1432 C
1433 C
1434 C
1435 C
1436 C
1437 C
1438 C
1439 C
1440 C
1441 C
1442 C
1443 C
1444 C
1445 C
1446 C
1447 C
1448 C
1449 C
1450 C
1451 C
1452 C
1453 C
1454 C
1455 C
1456 C
1457 C
1458 C
1459 C
1460 C
1461 C
1462 C
1463 C
1464 C
1465 C
1466 C
1467 C
1468 C
1469 C
1470 C
1471 C
1472 C
1473 C
1474 C
1475 C
1476 C
1477 C
1478 C
1479 C
1480 C
1481 C
1482 C
1483 C
1484 C
1485 C
1486 C
1487 C
1488 C
1489 C
1490 C
1491 C
1492 C
1493 C
1494 C
1495 C
1496 C
1497 C
1498 C
1499 C
1500 C
1501 C
1502 C
1503 C
1504 C
1505 C
1506 C
1507 C
1508 C
1509 C
1510 C
1511 C
1512 C
1513 C
1514 C
1515 C
1516 C
1517 C
1518 C
1519 C
1520 C
1521 C
1522 C
1523 C
1524 C
1525 C
1526 C
1527 C
1528 C
1529 C
1530 C
1531 C
1532 C
1533 C
1534 C
1535 C
1536 C
1537 C
1538 C
1539 C
1540 C
1541 C
1542 C
1543 C
1544 C
1545 C
1546 C
1547 C
1548 C
1549 C
1550 C
1551 C
1552 C
1553 C
1554 C
1555 C
1556 C
1557 C
1558 C
1559 C
1560 C
1561 C
1562 C
1563 C
1564 C
1565 C
1566 C
1567 C
1568 C
1569 C
1570 C
1571 C
1572 C
1573 C
1574 C
1575 C
1576 C
1577 C
1578 C
1579 C
1580 C
1581 C
1582 C
1583 C
1584 C
1585 C
1586 C
1587 C
1588 C
1589 C
1590 C
1591 C
1592 C
1593 C
1594 C
1595 C
1596 C
1597 C
1598 C
1599 C
1600 C
1601 C
1602 C
1603 C
1604 C
1605 C
1606 C
1607 C
1608 C
1609 C
1610 C
1611 C
1612 C
1613 C
1614 C
1615 C
1616 C
1617 C
1618 C
1619 C
1620 C
1621 C
1622 C
1623 C
1624 C
1625 C
1626 C
1627 C
1628 C
1629 C
1630 C
1631 C
1632 C
1633 C
1634 C
1635 C
1636 C
1637 C
1638 C
1639 C
1640 C
1641 C
1642 C
1643 C
1644 C
1645 C
1646 C
1647 C
1648 C
1649 C
1650 C
1651 C
1652 C
1653 C
1654 C
1655 C
1656 C
1657 C
1658 C
1659 C
1660 C
1661 C
1662 C
1663 C
1664 C
1665 C
1666 C
1667 C
1668 C
1669 C
1670 C
1671 C
1672 C
1673 C
1674 C
1675 C
1676 C
1677 C
1678 C
1679 C
1680 C
1681 C
1682 C
1683 C
1684 C
1685 C
1686 C
1687 C
1688 C
1689 C
1690 C
1691 C
1692 C
1693 C
1694 C
1695 C
1696 C
1697 C
1698 C
1699 C
1700 C
1701 C
1702 C
1703 C
1704 C
1705 C
1706 C
1707 C
1708 C
1709 C
1710 C
1711 C
1712 C
1713 C
1714 C
1715 C
1716 C
1717 C
1718 C
1719 C
1720 C
1721 C
1722 C
1723 C
1724 C
1725 C
1726 C
1727 C
1728 C
1729 C
1730 C
1731 C
1732 C
1733 C
1734 C
1735 C
1736 C
1737 C
1738 C
1739 C
1740 C
1741 C
1742 C
1743 C
1744 C
1745 C
1746 C
1747 C
1748 C
1749 C
1750 C
1751 C
1752 C
1753 C
1754 C
1755 C
1756 C
1757 C
1758 C
1759 C
1760 C
1761 C
1762 C
1763 C
1764 C
1765 C
1766 C
1767 C
1768 C
1769 C
1770 C
1771 C
1772 C
1773 C
1774 C
1775 C
1776 C
1777 C
1778 C
1779 C
1780 C
1781 C
1782 C
1783 C
1784 C
1785 C
1786 C
1787 C
1788 C
1789 C
1790 C
1791 C
1792 C
1793 C
1794 C
1795 C
1796 C
1797 C
1798 C
1799 C
1800 C
1801 C
1802 C
1803 C
1804 C
1805 C
1806 C
1807 C
1808 C
1809 C
1810 C
1811 C
1812 C
1813 C
1814 C
1815 C
1816 C
1817 C
1818 C
1819 C
1820 C
1821 C
1822 C
1823 C
1824 C
1825 C
1826 C
1827 C
1828 C
1829 C
1830 C
1831 C
1832 C
1833 C
1834 C
1835 C
1836 C
1837 C
1838 C
1839 C
1840 C
1841 C
1842 C
1843 C
1844 C
1845 C
1846 C
1847 C
1848 C
1849 C
1850 C
1851 C
1852 C
1853 C
1854 C
1855 C
1856 C
1857 C
1858 C
1859 C
1860 C
1861 C
1862 C
1863 C
1864 C
1865 C
1866 C
1867 C
1868 C
1869 C
1870 C
1871 C
1872 C
1873 C
1874 C
1875 C
1876 C
1877 C
1878 C
1879 C
1880 C
1881 C
1882 C
1883 C
1884 C
1885 C
1886 C
1887 C
1888 C
1889 C
1890 C
1891 C
1892 C
1893 C
1894 C
1895 C
1896 C
1897 C
1898 C
1899 C
1900 C
1901 C
1902 C
1903 C
1904 C
1905 C
1906 C
1907 C
1908 C
1909 C
1910 C
1911 C
1912 C
1913 C
1914 C
1915 C
1916 C
1917 C
1918 C
1919 C
1920 C
1921 C
1922 C
1923 C
1924 C
1925 C
1926 C
1927 C
1928 C
1929 C
1930 C
1931 C
1932 C
1933 C
1934 C
1935 C
1936 C
1937 C
1938 C
1939 C
1940 C
1941 C
1942 C
1943 C
1944 C
1945 C
1946 C
1947 C
1948 C
1949 C
1950 C
1951 C
1952 C
1953 C
1954 C
1955 C
1956 C
1957 C
1958 C
1959 C
1960 C
1961 C
1962 C
1963 C
1964 C
1965 C
1966 C
1967 C
1968 C
1969 C
1970 C
1971 C
1972 C
1973 C
1974 C
1975 C
1976 C
1977 C
1978 C
1979 C
1980 C
1981 C
1982 C
1983 C
1984 C
1985 C
1986 C
1987 C
1988 C
1989 C
1990 C
1991 C
1992 C
1993 C
1994 C
1995 C
1996 C
1997 C
1998 C
1999 C
2000 C

```

<PAGE 27>

///// FILE:CREATE /////

>>>> SUBROUTINE DISP <<<<

<PAGE 27>

<PAGE 28>

<PAGE 28>

I DATE:02-05-80,10:40 CYNH:JDSN JILL:CRFATE

15M

```

135 FORMAT ('0', A1, 'MESSAGE', 7X, 'ORIGINATOR', A1, 'NAME', 12X,
1 'REP', 9X, 'MSG DISK', A1, 'DATA DISK', 7X, 'MSG OUT', A1,
2 'MSG OUT', /, 10X, 'NAME', 26X, 'LENGTH', 11X, 'TIME', 6X,
3 'REAR/WITES', 11, 'REAR/WITES', 4X, 'SAMP ROUTE', 4X,
4 'CINR ROUTE', 16X, 'CHARACTERS', 5X, 'MILLSSEC', 5X,
5 'CINR MSG', 5X, 'CINR MSG', /)

```

USER GENERATED MESSAGES

```

IF (NUM -FO. 0) GO TO 165

```

```

DO 160 I = 1, NUM

```

```

  INC1 = MC(I,1)

```

```

  INC3 = MC(I,3)

```

```

  INC4 = MC(I,4)

```

```

  IF (MP(I,1) -NE. 0 -AND. MP(I,2) -NE. 0) WRITE (6,140) NAME(
1 I, IND1, MD(I,2), IND3, IND4, NAME(MP(I,1), NAME(MP(I,2))
  FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I, 10X, A8, 7X, A8)

```

```

  IF (MP(I,1) -NE. 0 -AND. MP(I,2) -FO. 0) WRITE (6,145) NAME(
1 I, IND1, MD(I,2), IND3, IND4, NAME(MP(I,1), NAME(MP(I,2))
  FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I, 10X, A8, 7X, 'NONE')

```

```

  IF (MP(I,1) -EQ. 0 -AND. MP(I,2) -NE. 0) WRITE (6,150) NAME(
1 I, IND1, MD(I,2), IND3, IND4, NAME(MP(I,2))
  FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I, 10X, 'NONE', 11X, A8)

```

```

  IF (MP(I,1) -EQ. 0 -AND. MP(I,2) -FO. 0) WRITE (6,155) NAME(
1 I, IND1, MD(I,2), IND3, IND4
  FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I, 10X, 'NONE', 11X, 'NONE')

```

```

160 CONTINUE

```

```

165 CONTINUE

```

105 GENERATED MESSAGES

```

IF (NUM -FO. 0) GO TO 180

```

```

IF (NUM -NE. 0 -AND. IC -NE. 0) WRITE (6,115)

```

```

  II = NUM + 1

```

```

DO 175 J = 1, II

```

```

  IND1 = MD(I,1)

```

```

  IND3 = MD(I,3)

```

```

  IND4 = MD(I,4)

```

```

  WRITE (6,170) NAME(I), IND1, MD(I,2), IND3, IND4

```

```

  FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I, 10X, 'NONE', 11X, 'NONE')

```

```

175 CONTINUE

```

```

180 CONTINUE

```

WRITE PROCESSOR HEADERS

```

  WRITE (6,10)

```

```

  WRITE (6,115)

```

```

105 FORMAT ('01', '*****', 13X, 'PROCESSOR DATA', 13X, '*****')

```

```

  WRITE (6,15)

```

<PAGE 28>

>>>> SUBROUTINE DISP <<<<

<PAGE 28>

<PAGE 30>

1 DATE:02-05-80,14:40 CMMER:IDSH FILE:CREATP
 1

<PAGE 30>

158

260 FORMAT (1X, 'PRIGADE', 10X, 'MESSAGE', 9X, 'INITIATION', /, 61X,
 1 'TYPE', 14X, 'RATE', /, 76X, '(MSG/HOUR)', /)

JJ = MOD(I, 1)
 JK = MOD(J, 1)

DO 255 I = JJ, JK

PO 250 J = 1, NUN

IRM = RM(1,J,1)

WRITE (6,265) RMAN(I), RMAN(J), IRM

FORMAT (43X, A8, 9X, A8, 11X, 13)

CONTINUE

WRITE (6,415)

255 CONTINUE

IP (JR,PO,MR) GO TO 280

WRITE (6,415)

WRITE (6,260)

MESSAGES FROM BATTALIONS

260 FORMAT (15X, 'PATTALION', 9X, 'PARFMT', 12X, 'MESSAGE', 11X,
 1 'INITIATION', 9X, 'FRCB OP', 12X, 'PROB OP', /, 33X,
 2 'BRIGADE', 12X, 'TYPE', 16X, 'RATE', 12X, 'DEFLECTION', 10X,
 3 'ATTENTION', /, 69X, '(MSG/HOUR)', 9X, 'AT REC', 11X,
 4 'AT DOE', /)

JL = JK + 1

ISPACE = LP(JL,3)

NO 270 I = JL, MR

IP (LP(I,3) - ME, ISPACE) WRITE (6,415)

ISPACE = LP(I,3)

DO 270 J = 1, NUN

IRM = RM(1,J,1)

WRITE (6,265) RMAN(I), RMAN(LP(I,3)), RMAN(J), IRM,

RM(1,J,3), RM(1,J,4)

FORMAT (15X, A8, 10X, A8, 10X, A8, 13X, 11, 15X, F4.2,
 15X, F4.2)

CONTINUE

WRITE (6,415)

275 CONTINUE

280 CONTINUE

IP (BCB,PO,0) GO TO 100

WRITE (6,415)

WRITE (6,265)

MESSAGES FROM CINFUSERS

285 FORMAT (43X, 'OTHER', 12X, 'MESSAGE', 9X, 'INITIATION', /, 43X,
 1 'USPRS', 13X, 'TYPE', 14X, 'RATE', /, 76X, '(MSG/HOUR)')

DO 290 I = 1, NUN

290 CONTINUE

295 CONTINUE

300 CONTINUE

305 CONTINUE

310 CONTINUE

315 CONTINUE

320 CONTINUE

////// FILE:CREATP //

>>>> SUBROUTINE DISP <<<<

<PAGE 30>

<PAGE 30>

<PAGE 11>

1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577

```

DO 290 J = 1, NMH
  IMM = MM(I,J,1)
  WRITE (6,245) MMAR(I), MMAR(J), IMM
290 CONTINUE
C
295 WRITE (6,415)
295 CONTINUE
C
300 CONTINUE
  IP (MMH - EQ. MP) GO TO 155
  JJ = MMH + 1
  WRITE (6,305)
C
C 105 GENERATED MESSAGES
C
305 FORMAT (4X, ' ** TOS GENERATED MESSAGES **', //, 43X,
1 'TERMINUS', 4X, 'MESSAGE', 7X, 'OUTPUT', 4X, 'OUTPUT', /,
2 56X, 'TYPE', 9X, 'RATIO A', 1X, 'RATIO B', /)
C
C 105 MESSAGES TO BRIGADES
C
  IP (MMDF - EQ. 0) GO TO 340
  II = MMH + 1
  JJ = MMH + 1
  DO 320 I = II, JJ
    DO 315 J = JJ, MM
      WRITE (6,310) MMAR(I), MMAR(J), MM(I,J,1), MM(I,J,2)
      FORMAT (43X, A6, 4X, A6, 6X, F5.1, 5X, F5.1)
315 CONTINUE
    WRITE (6,415)
320 CONTINUE
C
325 CONTINUE
C
C 105 MESSAGES TO BATTALIONS
C
  IP (MMH + MMDF - EQ. MM) GO TO 340
  WRITE (6,415)
  JJ = JJ + 1
  ISPACE = IP(JJ,3)
C
  DO 335 I = JJ, MM
    IF (IP(I,1) - NE. ISPACE) WRITE (6,415)
    ISPACE = IP(I,3)
C
    DO 330 J = JJ, MM
      WRITE (6,310) MMAR(I), MMAR(J), MM(I,J,1), MM(I,J,2)
330 CONTINUE
C
335 WRITE (6,415)
335 CONTINUE
C
340 CONTINUE
C
C 105 MESSAGES TO OTHER USERS

```

<PAGE 11>

////// FILE:CREATE /////

>>>> SUBROUTINE DISP <<<<

<PAGE 11>

DATE:02-05-80,14:40 CMMER:105H FILE:CREATE

158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195

<PAGE 32>

 I DATE:02-05-80,14:40 CYMER:YOSH FILE:CREATE
 I-----I

<PAGE 32>

```

1578 C      IP (MCU -FO. 0) GO TC 355
1579      WRITE (6,415)
1580 C
1581 C      DO 350 I = 1, NCU
1582 C
1583 C          CO 345 J = 1J, N4
1584 C              WRITE (6,310) RMAN(I), RMAN(J), RMAN(I,J,1), RMAN(I,J,2)
1585 C          CONTINUE
1586 C
1587 C              WRITE (6,415)
1588 C          CONTINUE
1589 C
1590 C          340 CONTINUE
1591 C
1592 C          355 CONTINUE
1593 C
1594 C              WRITE THE ERROR RATE ARRAY
1595 C
1596 C              WRITE (6,10)
1597 C              WRITE (6,360)
1598 C              FORMAT (30X, '*****', 11X, 'ERROR RATES', 15X, '*****')
1599 C              WRITE (6,15)
1600 C              WRITE (6,365)
1601 C              365 FORMAT (15X, 'SENDING', 7X, 'RECEIVING', 8X, 'ERROR',
1602 C                  1 22X, 'SENDING', 7X, 'RECEIVING', 8X, 'ERROR', /,
1603 C                  2 16X, 'MODE', 11X, 'MODE', 11X, 'PATH', 24X, 'MODE', 11X,
1604 C                  3 'MODE', 11X, 'MODE', /)
1605 C              IP (NCP -FO. 0) GO TC 195
1606 C              JJ = NCU + NDD
1607 C              IJ = PCU + 1
1608 C
1609 C          DO 375 I = 1J, JJ
1610 C              WRITE (6,370) RMAN(I), LD(I,1), RMAN(I), LD(I,2)
1611 C              FORMAT (15X, AB, 11X, 'PPE', 11X, P5.1, 22X, 'FPE', 12X, AB,
1612 C                  1 8X, P5.1)
1613 C          CONTINUE
1614 C
1615 C          380 CONTINUE
1616 C              IP (JA -FO. 0) GO TC 395
1617 C              WRITE (6,415)
1618 C              JJ = NCU + NDD + 1
1619 C              ISPACE = LP(JJ,3)
1620 C
1621 C          DO 390 I = 1J, N4
1622 C              IP (LP(I,1) -NE. ISPACE) WRITE (6,415)
1623 C              ISPACE = LP(I,3)
1624 C              WRITE (6,385) RMAN(I), RMAN(LP(I,3)), LD(I,1), RMAN(LP(I,3)),
1625 C                  1 RMAN(I), LD(I,2)
1626 C              385 FORMAT (15X, AB, 7X, AB, 8X, P5.1, 22X, AB, 7X, AB, 8X, P5.1)
1627 C          CONTINUE
1628 C
1629 C          395 CONTINUE
1630 C              IP (MCU -FO. 0) GO TC 405
1631 C              WRITE (6,415)
1632 C
1633 C          DO 400 I = 1, NCU
1634 C              WRITE (6,370) RMAN(I), LD(I,1), RMAN(I), LD(I,2)
1635 C          CONTINUE

```

<PAGE 32>

///// FILE:CREATE /////

>>>> SUBROUTINE DISP <<<<

<PAGE 32>

<PAGE 33>

ISM

235
236
237
238
239
240

DATE: 07-05-80, 10:40 CMMER:IDSW FILE:CMRATP

405 CONTINUE
WRITE (6,410)
410 FORMAT (' ')
415 FORMAT (' ')
RETURN
END

<PAGE 31>

1636
1637
1638
1639
1640
1641

<PAGE 33>

>>>> SUBROUTINE DISE <<<<

//// FILE:CMRATP ////

<PAGE 31>

<PAGE 1>

DATE:02-05-80,14:01 CYNPR:IOSH FILE:DISPLAY

ISH

```

1 C
2 C
3 C
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C

***** F - I - S - P - I - A - V *****
***** TOS FILE WRITING ROUTINE *****

THIS PROGRAM WILL READ THE TOS SIMULATION
INPUT FILE CREATED BY PROGRAM CREATE OR
PROGRAM MODIFY AND WRITE THE DATA IN AN
FAST TO REAL FORMAT.

THE INPUT FILE IS READ FROM DEVICE NUMBER 2

      DEK VRI API-3 23 JAN 80

      READ INPUT FILE HERE

      REAL*8 RMAR(50), MMAR(50), CHAN(50)
      REAL IC(50,2), ME(50,4)
      INTEGER RP(50,2), PE(50), LP(50,3)
      DIMENSION DN(2), GPE(11,2), CD(10,4), MP(50,2), RN(50,50,4),
1      IPD(50)

      INDEXING AND COUNTING INFO

      READ (2,10) RN, MM, MM, MM, NC, NP, NL, MUQ, NUDE, IFG
10 FORMAT (10(I3,1X))
      DO 20 I = 1, NR
          MPAT (2,15) RMAR(I), RP(I,1), MP(I,2), IPD(I)
15      FORMAT (A9, 3(1X,1))
20 CONTINUE

      MESSAGE INFO

      F00LE INFO

      DO 30 I = 1, NM
          READ (2,25) MMAR(I), PPI(I,1), ME(I,2), MD(I,1), MD(I,2),
1          PDI(I,1), MD(I,4)
25      FORMAT (A9, 2(1X,13), 4(1X,P9.3))
30 CONTINUE

      CHANNEL INFO

      DO 40 I = 1, NC
          READ (2,35) CHAN(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)
35      FORMAT (A8, 4(1X,P9.3))
40 CONTINUE

      NUMBER OF TEMPLATES

      DO 50 I = 1, NF
          MPAT (2,45) PDI(I)
45      FORMAT (2X, 13)
50 CONTINUE

      ITER INFO

```

<PAGE 1>

>>>> MAIN PROGRAM <<<<

///// FILE:DISPLAY /////

<PAGE 1>

<PAGE 2>

```

1  DATE:02-05-80,14:41  CUSER:IDSH  FILE:DISPLAY
DO 6C I = 1, NI
  READ (2,55) LP(I,1), IP(I,2), LF(I,3), LD(I,4), LO(I,5), LD(I,6,1X)
60 CONTINUE
C
C      FOURTH MESSAGE AREA
C
  IF (IP(I,2) .EQ. 1) READ (2,65) FLAG
  IF (FLAG .EQ. 1) GO TO 85
  IF (IP(I,2) .EQ. 1) GO TO 85
  DO 75 J = 1, NR
    READ (2,70) RM(I,J,1), RM(I,J,2), RM(I,J,3), RM(I,J,4)
    FORMAT (4(E9.3,1X))
70 CONTINUE
75 CONTINUE
80 CONTINUE
85 CONTINUE
C
C      FISH ACCESS TIMES
C
  READ (2,90) GE(I), ED(2)
  FORMAT (2(E9.3,1X))
90 CONTINUE
C
C      GENERAL PROCESSOR DATA
C
  DO 95 I = 1, 11
    READ (2,90) GPE(I,1), GDE(I,2)
95 CONTINUE
C
C      END OF READ SECTION
C
  WRITE SYSTEM CONFIGURATION HEADERS
C
  WRITE (6,100)
100 FORMAT ('-1', 17X, 50('**'), '/', 10X, '*****', 40X, '*****')
105 FORMAT (10X, '*****', 40X, '*****', '/', 30X, 50('**'), '/')
  WRITE (6,110)
110 FORMAT (10X, '*****', 10X, 'SYSTEM CONFIGURATION', 10X, '*****')
  WRITE (6,105)
  IF (NPEP .EQ. 0) GO TO 160
C
C      CONFIGURATION OF BRIGADES
C
  11 = FHC + 1
  JJ = NPEP + MPC
  WRITE (6,115)
115 FORMAT ('0', 50X, '*** MANEUVER BRIGADES ***', //, 22X, 'UNIT',
1 16X, 'COMMUNICATIONS', 13X, 'PROCESSOR', 16X, 'NUMBER',
2 10X, //, 22X, 'NAME', 17X, 'LINK #', 17X, 'TYPE',
3 10X, 'TELETYPE', '/')
  DO 135 I = 1, JJ
    IF (IP(I,2) .EQ. 1) GO TO 125
    WRITE (6,120) RM(I,1), RM(I,2), PD(I)
    FORMAT (22X, A8, 15X, A8, 10X, 'TCT', 22X, 12)
    GO TO 135
125 WRITE (6,130) RM(I,1), RM(I,2), PD(I)
130 FORMAT (22X, A8, 15X, A8, 10X, 'TCS', 22X, 12)
135 CONTINUE

```

<PAGE 2>

///// FILE:DISPLAY /////

>>>> MAIN PROGRAM <<<<

<PAGE 2>

<PAGE 3>

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

<PAGE 3>

158

```

-----
1 DATE:02-05-80,14:41  CMMER:IOSH  FILE:DISFIAY
-----

```

C

C

CONFIGURATION OF BATTALIONS

IF (JJ .EQ. 0) GO TO 160

WRITE (6,140)

140 FORMAT ('0', 49X, '** RANHEVER BATTALIONS **', //, 22X, 'UNIT',

1 14X, 'COMMUNICATIONS', 15X, 'PARANT', 17X, 'PROCESSOR', /,

2 22X, 'NAME', 16X, 'LINK W/ BRIGADE', 14X, 'BRIGADE', 18X,

3 'TYPE', /)

JJ = JJ + 1

ISPACE = IP(JJ,3)

DO 160 I = JJ, NR

IF (IP(I,3) .NE. ISPACE) WRITE (6,140)

ISPACE = IP(I,3)

IF (IP(I,1) .EQ. 1) GO TO 150

WRITE (6,145) RNAME(I), CNAME(I,2), RNAME(I,3)

145 FORMAT ('2X, AB, 16X, AB, 10X, AB, 17X, 'TCT')
GO TO 160

150 WRITE (6,155) RNAME(I), CNAME(I,2), RNAME(I,3)

155 FORMAT ('22X, AB, 16X, AB, 18X, AB, 17X, 'TCS')
160 CONTINUE

C

C

CONFIGURATION OF OTHER TCS USERS

IF (MNO .EQ. 0) GO TO 185

WRITE (6,165)

165 FORMAT ('0', 51X, '** OTHER TCS USERS **', //, 38X, 'UNIT', 14X,

1 14X, 'COMMUNICATIONS', 12X, 'PROCESSOR', /, 30X, 'NAME', 10X,

2 'LINK', 19X, 'TYPE', /)

DO 185 I = 1, NUC

IF (IP(I,1) .EQ. 1) GO TO 175

WRITE (6,170) RNAME(I), CNAME(I,2)

170 FORMAT ('18X, AB, 11X, AB, 10X, 'TCT')
GO TO 185

175 WRITE (6,180) RNAME(I), CNAME(I,2)

180 FORMAT ('18X, AB, 11X, AB, 10X, 'TCS')
185 CONTINUE

C

C

WRITE DATA ABOUT THE COMMUNICATIONS NET

WRITE (6,190)

WRITE (6,190)

190 FORMAT ('18X, *****', 9X, 'COMMUNICATIONS SYSTEM', 10X, '*****')

WRITE (6,195)

WRITE (6,195)

195 FORMAT ('0', 29X, 'CHANNEL', 9X, 'NUMBER', 7X, 'MISSION', 9X,

1 'RISF', 7X, 'UNAVAILABLE', /, 31X, 'NAME', 10X, 'OF LINES',

2 9X, 'RATE', 12X, 'TIME', 9X, 'CAPACITY', /, 59X,

3 'CHARA/SEC', 6X, '(SECONDS)', 9X, '(FRACTION)', /)

DO 205 I = 1, NC

ICD = CD(I,4)

JCC = CD(I,2)

WRITE (6,200) CNAME(I), ICD, JCC, CD(I,1), CD(I,3)

200 FORMAT ('29X, AB, 10X, I3, 11X, I4, 12X, F4.2, 11X, F4.2')
205 CONTINUE

C

C

WRITE DATA ABOUT THE MESSAGES

<PAGE 3>

///// FILE:DISPLAY /////

>>>> MAIN PROGRAM <<<<

<PAGE 3>

175
176
177
178
179
180
181
182

DATE: 02-05-00, 14:01 CENTER: USN FILE: DISPLAY

<PAGE 4>

ISI

```

WRITE (6,100)
WRITE (6,210)
FORMAT (21X)
WRITE (6,105)
WRITE (6,215)
FORMAT (10X, 'MESSAGE', 7X, 'ORIGINATOR', 4X, 'MEAN', 12X,
1 'MSG', 9X, 'MSG DISK', 6X, 'DATA DISK', 7X, 'MSG OUT', 8X,
2 'MSG OUT', /, 10X, 'NAME', 26X, 'LENGTH', 11X, 'TIME', 6X,
3 'READS/WRITES', 1X, 'READS/WRITES', 4X, 'SAME ROUTE', 4X,
4 'OTHER ROUTE', 16X, '(CHARACTERS)', 5X, '(MILLISEC)', 5X,
5 '(PER MSG)', 5X, '(PER MSG)', /)

```

USPP GENERATED MESSAGES

```

IF (M04, .EQ. 0) GO TO 240
NO 240 I = 1, NUN
IM0 = MP(I, 1)
IM03 = MP(I, 3)
IM04 = MD(I, 4)
IM05 = MD(I, 5)
IF (MP(I, 1) - NE. 0 -AND. MP(I, 2) - NE. 0) WRITE (6, 220) MNUM(I),
1 IM0, IM03, MP(I, 2), IM03, IM04, MNUM(MP(I, 1)), MNUM(MP(I, 2))
220 FORMAT (9X, A9, A9, 'USFR', 10X, I4, 12X, F6.3, 10X, I2, 12X,
3 I3, 10X, A9, 7X, A9)
IF (MP(I, 1) - NE. 0 -AND. MP(I, 2) - EQ. 0) WRITE (6, 225) MNUM(I),
1 IM0, IM03, MD(I, 2), IM03, IM04, MNUM(MP(I, 1))
225 FORMAT (9X, A9, A9, 'USFR', 10X, I4, 12X, F6.3, 10X, I2, 12X,
3 I3, 10X, A9, 7X, 'MCN0')
IF (MP(I, 1) - EQ. 0 -AND. MP(I, 2) - NE. 0) WRITE (6, 230) MNUM(I),
1 IM0, MP(I, 2), IM03, IM04, MNUM(MP(I, 2))
230 FORMAT (9X, A9, A9, 'USFR', 10X, I4, 12X, F6.3, 10X, I2, 12X,
3 I3, 10X, 'NONE', 11X, A9)
IF (MP(I, 1) - EQ. 0 -AND. MP(I, 2) - EQ. 0) WRITE (6, 235) MNUM(I),
1 IM0, MP(I, 2), IM03, IM04
235 FORMAT (9X, A9, A9, 'USFR', 10X, I4, 12X, F6.3, 10X, I2, 12X,
3 I3, 10X, 'NONE', 11X, 'NONE')
NO 240 CONTINUE

```

105 GENERATED MESSAGES

```

17 IN05 =0, NF) GO TO 250
18 IF (MOD .NF, 0 .AND. IC .NE. 0) WRITE (6,400)
19 IT = A04 + 1
20 2C I = II, NN
    IREI = NE(1,1)
    IREJ = NE(1,3)
    IREK = ME(1,3)
    IRE4 = ME(1,4)
    WRITE (6,205) KRAM(1), INDI, MD(1,2), INDI, INDI4
    SORTAT (9, A0, 9X, 'COS', 11X, 14, 12X, F6.3, 10X, 12, 12X,
    13, 10X, 'NONE', 11X, 'NONE')
50 CONTINUE

```

ITF PROCESS HEADERS

```
WRITE (5,100)
WRITE (6,255)
55, FORMAT (1PX, ....., 11X, 'PROCESSOR DATA', 11X, .....)
WRITE (5,105)
WRITE (6,260)
```

<PAGE 4>

PIFF:DISPLAY ////

>>>> MAIN PROGRAM <<<<

<PAGE 4>

<PAGE 5>

DATE:02-05-80,14:41 CMMER:105H FILE:DISPLAY

<PAGE 5>

ISM

```

233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

<PAGE 5>

FILE:DISPLAY

MAIN PROGRAM

<PAGE 5>

<PAGE 6>

```

1 DATE:02-05-80,14:41 CYMR:IDSN FILE:DISPLAY

```

<PAGE 6>

```

291 DO 315 I = 1, JK
292 EC 310 J = 1, NM
293   IIR = RM(I,J,1)
294   WRITE (6,305) RMAM(I), MMAM(J), IIR
295   FORMAT (A1X, A8, 9X, A8, 11X, 13)
296   CONTINUE
297   WRITE (6,440)
298   CONTINUE
299   IF (JF - EQ. N5) GO TO 315
300   WRITE (6,440)
301   WRITE (6,320)
302
303   C
304   C
305   C
306   C
307   C
308   C
309   C
310   C
311   C
312   C
313   C
314   C
315   C
316   C
317   C
318   C
319   C
320   C
321   C
322   C
323   C
324   C
325   C
326   C
327   C
328   C
329   C
330   C
331   C
332   C
333   C
334   C
335   C
336   C
337   C
338   C
339   C
340   C
341   C
342   C
343   C
344   C
345   C
346   C
347   C
348   C
349   C
350   C
351   C
352   C
353   C
354   C
355   C
356   C
357   C
358   C
359   C
360   C
361   C
362   C
363   C
364   C
365   C
366   C
367   C
368   C
369   C
370   C
371   C
372   C
373   C
374   C
375   C
376   C
377   C
378   C
379   C
380   C
381   C
382   C
383   C
384   C
385   C
386   C
387   C
388   C
389   C
390   C
391   C
392   C
393   C
394   C
395   C
396   C
397   C
398   C
399   C
400   C
401   C
402   C
403   C
404   C
405   C
406   C
407   C
408   C
409   C
410   C
411   C
412   C
413   C
414   C
415   C
416   C
417   C
418   C
419   C
420   C
421   C
422   C
423   C
424   C
425   C
426   C
427   C
428   C
429   C
430   C
431   C
432   C
433   C
434   C
435   C
436   C
437   C
438   C
439   C
440   C
441   C
442   C
443   C
444   C
445   C
446   C
447   C
448   C
449   C
450   C
451   C
452   C
453   C
454   C
455   C
456   C
457   C
458   C
459   C
460   C
461   C
462   C
463   C
464   C
465   C
466   C
467   C
468   C
469   C
470   C
471   C
472   C
473   C
474   C
475   C
476   C
477   C
478   C
479   C
480   C
481   C
482   C
483   C
484   C
485   C
486   C
487   C
488   C
489   C
490   C
491   C
492   C
493   C
494   C
495   C
496   C
497   C
498   C
499   C
500   C
501   C
502   C
503   C
504   C
505   C
506   C
507   C
508   C
509   C
510   C
511   C
512   C
513   C
514   C
515   C
516   C
517   C
518   C
519   C
520   C
521   C
522   C
523   C
524   C
525   C
526   C
527   C
528   C
529   C
530   C
531   C
532   C
533   C
534   C
535   C
536   C
537   C
538   C
539   C
540   C
541   C
542   C
543   C
544   C
545   C
546   C
547   C
548   C
549   C
550   C
551   C
552   C
553   C
554   C
555   C
556   C
557   C
558   C
559   C
560   C
561   C
562   C
563   C
564   C
565   C
566   C
567   C
568   C
569   C
570   C
571   C
572   C
573   C
574   C
575   C
576   C
577   C
578   C
579   C
580   C
581   C
582   C
583   C
584   C
585   C
586   C
587   C
588   C
589   C
590   C
591   C
592   C
593   C
594   C
595   C
596   C
597   C
598   C
599   C
600   C
601   C
602   C
603   C
604   C
605   C
606   C
607   C
608   C
609   C
610   C
611   C
612   C
613   C
614   C
615   C
616   C
617   C
618   C
619   C
620   C
621   C
622   C
623   C
624   C
625   C
626   C
627   C
628   C
629   C
630   C
631   C
632   C
633   C
634   C
635   C
636   C
637   C
638   C
639   C
640   C
641   C
642   C
643   C
644   C
645   C
646   C
647   C
648   C
649   C
650   C
651   C
652   C
653   C
654   C
655   C
656   C
657   C
658   C
659   C
660   C
661   C
662   C
663   C
664   C
665   C
666   C
667   C
668   C
669   C
670   C
671   C
672   C
673   C
674   C
675   C
676   C
677   C
678   C
679   C
680   C
681   C
682   C
683   C
684   C
685   C
686   C
687   C
688   C
689   C
690   C
691   C
692   C
693   C
694   C
695   C
696   C
697   C
698   C
699   C
700   C
701   C
702   C
703   C
704   C
705   C
706   C
707   C
708   C
709   C
710   C
711   C
712   C
713   C
714   C
715   C
716   C
717   C
718   C
719   C
720   C
721   C
722   C
723   C
724   C
725   C
726   C
727   C
728   C
729   C
730   C
731   C
732   C
733   C
734   C
735   C
736   C
737   C
738   C
739   C
740   C
741   C
742   C
743   C
744   C
745   C
746   C
747   C
748   C
749   C
750   C
751   C
752   C
753   C
754   C
755   C
756   C
757   C
758   C
759   C
760   C
761   C
762   C
763   C
764   C
765   C
766   C
767   C
768   C
769   C
770   C
771   C
772   C
773   C
774   C
775   C
776   C
777   C
778   C
779   C
780   C
781   C
782   C
783   C
784   C
785   C
786   C
787   C
788   C
789   C
790   C
791   C
792   C
793   C
794   C
795   C
796   C
797   C
798   C
799   C
800   C
801   C
802   C
803   C
804   C
805   C
806   C
807   C
808   C
809   C
810   C
811   C
812   C
813   C
814   C
815   C
816   C
817   C
818   C
819   C
820   C
821   C
822   C
823   C
824   C
825   C
826   C
827   C
828   C
829   C
830   C
831   C
832   C
833   C
834   C
835   C
836   C
837   C
838   C
839   C
840   C
841   C
842   C
843   C
844   C
845   C
846   C
847   C
848   C
849   C
850   C
851   C
852   C
853   C
854   C
855   C
856   C
857   C
858   C
859   C
860   C
861   C
862   C
863   C
864   C
865   C
866   C
867   C
868   C
869   C
870   C
871   C
872   C
873   C
874   C
875   C
876   C
877   C
878   C
879   C
880   C
881   C
882   C
883   C
884   C
885   C
886   C
887   C
888   C
889   C
890   C
891   C
892   C
893   C
894   C
895   C
896   C
897   C
898   C
899   C
900   C
901   C
902   C
903   C
904   C
905   C
906   C
907   C
908   C
909   C
910   C
911   C
912   C
913   C
914   C
915   C
916   C
917   C
918   C
919   C
920   C
921   C
922   C
923   C
924   C
925   C
926   C
927   C
928   C
929   C
930   C
931   C
932   C
933   C
934   C
935   C
936   C
937   C
938   C
939   C
940   C
941   C
942   C
943   C
944   C
945   C
946   C
947   C
948   C
949   C
950   C
951   C
952   C
953   C
954   C
955   C
956   C
957   C
958   C
959   C
960   C
961   C
962   C
963   C
964   C
965   C
966   C
967   C
968   C
969   C
970   C
971   C
972   C
973   C
974   C
975   C
976   C
977   C
978   C
979   C
980   C
981   C
982   C
983   C
984   C
985   C
986   C
987   C
988   C
989   C
990   C
991   C
992   C
993   C
994   C
995   C
996   C
997   C
998   C
999   C
1000   C

```

<PAGE 6>

```

111111 FILE:DISPLAY

```

```

111111 MAIN PROGRAM

```

<PAGE 6>

<PAGE 8>

807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827

```

815 CONTINUE
  IF (JJ .EQ. NR) GO TO 825
  WRITE (6,440)
  JJ = JJ + 1
  ISPACE = LP(JJ,1)
  DO 825 I = JJ, NR
    IF (LP(I,1) .NE. ISPACE) WRITE (6,440)
    ISPACE = LP(I,1)
    WRITE (6,420) RNAME(I), RNAME(LP(I,1)), LD(I,1), RNAME(LP(I,1)),
      1 NAME(I), LD(I,2)
    820 FORMAT (15X, A6, 7X, A6, 9X, P5.1, 22X, A6, 7X, A6, 9X, P5.1)
    825 CONTINUE
  IF (NR .EQ. 0) GO TO 810
  WRITE (6,440)
  DO 810 I = 1, NRC
    WRITE (6,410) RNAME(I), LD(I,1), RNAME(I), LD(I,2)
  810 CONTINUE
  WRITE (6,415)
  815 FORMAT (11)
  820 FORMAT (11)
  END

```

<PAGE 8>

158

1 DATE:02-05-80,18:41 GENE:IDSH FILE:DISPLAY

241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260

<PAGE 8>

///// FILE:DISPLAY /////

>>>> MAIN PROGRAM <<<<

<PAGE 8>

<PAGE 1>
ISN

DATE:02-05-80,14:42 CUNEB:LDSE FILE:MODIFY

```

1  C
2  C
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C

PROGRAM MODIFY
THIS PROGRAM PRODUCES A NEW INPUT FILE TO RUN THE SIMULATION
PROGRAM USING AN EXISTING INPUT FILE AT A MODEL.

RUN WITH I/O UNITS:
1-EXISTING INPUT FILE 2=NEW INPUT FILE(CREATED BY THIS PROGRAM)

FORMAT(' *** PROGRAM MODIFY ***')
FORMAT(' DO YOU WISH TO ALTER THE CONFIGURATION? (YES OR NO)')
FORMAT(' DO YOU WISH TO ALTER THE COMMUNICATIONS SYSTEM?')
FORMAT(' DO YOU WISH TO ALTER THE MESSAGE DATA?')
FORMAT(' DO YOU WISH TO ALTER THE PROCESSOR DATA?')
FORMAT(' DO YOU WISH TO ALTER THE SYSTEM LOADS?')
FORMAT(' DO YOU WISH TO ALTER THE ERROR RATES?')
FORMAT(' DO YOU WISH TO MAKE ANY ADDITIONAL CHANGES?')
FORMAT(' ***A NEW INPUT FILE HAS BEEN CREATED***')

INTEGER ANSWER, YES, NO, MC, NM
INTEGER MC, NM, NI, NP, NR, NB, NHQ, NUM, NREF
SPRINT BEHAF(5), DNAME(10,5), NAME(15), CMAN(10)
REAL IN(50,2), ND(50,4)
INTEGER IP(50,2), RNUM, IN(50), BD(10), BA(10,5), NO(15)
DIMENSION NNM(5), LINK(10,5), LINK(15), LP(50,3),
*ND(2), GPD(1,2), CD(10,4), NP(50,2), NI(50,50,4), IP(50)
COMMON/INT/ MC, NM, NI, NP, NR, NB, NHQ, NUM, NREF
COMMON/BEHAF, DNAME, NAME, CMAN
COMMON IN, ND
COMMON IP, RNUM, BD, BE, BN, HQ
COMMON NNM, LINK, LINK, LINK, IP,
*ND, GPD, CD, NP, NI, IPD

READ IN EXISTING DATA
CALL READP
ALTER DATA
WRITE(6,10)
CALL PRAC(5,'S:',ANSWER,1)
IF (ANSWER -PC- NO) GO TO 100
CALL CCMFIC
WRITE(6,10)
CALL PRAC(5,'S:',ANSWER,1)
IF (ANSWER -PC- NO) GO TO 110
CALL CCMFIC
WRITE(6,40)
CALL PRAC(5,'S:',ANSWER,1)
IF (ANSWER -PC- NO) GO TO 120
CALL P4C
WRITE(6,50)
CALL PRAC(5,'S:',ANSWER,1)

```

<PAGE 1>

>>>> MAIN PROGRAM <<<<

FILE:MODIFY

<PAGE 1>

LINE	CODE	TEXT	DATE: 02-05-80, 14:42	TYPE: 105W	FILE: MODIFY
59		IF (ANSWER .EQ. NO) GO TO 130			
60		CALL TRCC			
61	130	WRITE(6,60)			
62		CALL READ(5,'S:',ANSWER,1)			
63		IF (ANSWER .EQ. NO) GO TO 140			
64		CALL INFACT			
65	140	WRITE(6,70)			
66		CALL READ(5,'S:',ANSWER,1)			
67		IF (ANSWER .EQ. NO) GO TO 150			
68		CALL NOISE			
69	150	WRITE(6,80)			
70		CALL READ(5,'S:',ANSWER,1)			
71		IF (ANSWER .EQ. YES) GO TO 800			
72	C				
73	C	WRITE NEW DATA FILE			
74	C				
75	C	CALL WRITEF			
76					
77		WRITE(6,90)			
78		STOP			
79		END			

```

<PAGE 2>
      //////////////////////////////////
      >>>> MAIN PROGRAM <<<<
<PAGE 2>

```


<PAGE 4>

DATE:02-05-80, 14:42 CMH:ICSM FILE:MODIFY

158

```

90 106 1-1, MC
100-CELL, 4)
100-CELL, 2)
WRITE(6,33) CHAR(1), ICD, JCE, CC(1,1), CU(1,1)
FORMAT(29E, A9, 10E, 13, 11E, 14, 12E, F4, 2, 11E, F4, 2)
156 CONTINUE
C
600 WRITE(6,60)
CALL PRPAR(5, 'S', ANSWER, 1)
IF (ANSWER .EQ. YES) GO TO 900
PRTOUT
END

```

<PAGE 4>

138
139
140
141
142
143
144
145
146
147
148
149

<PAGE 4>

>>>> SUBROUTINE CONSIST <<<<

///// FILE:MODIFY /////

<PAGE 4>

<PAGE 5>

1 DATE:02-05-80,14:42 CMMER:1050 FILE:MODIFY 1

13M

1

2

3

SUBROUTINE CONFIG
RETURN
END

<PAGE 5>

150
151
152

<PAGE 5>

>>>> SUBROUTINE CONFIG <<<<

////// FILE:MODIFY //

<PAGE 5>

• (9 454d)

<PAGE 6>

<PAGE 7>

194
195
196

SUBROUTINE MSG
RETURN
END

DATE: 02-05-80, 14:42 CMMER: IDSN FILE: MODIFY

ISM

1
2
3

<PAGE 7>

<PAGE 7>

////// FILE: MODIFY //

>>>> SUPROUTINE MSG <<<<

<PAGE 7>

<PAGE 9>

DATE:02-05-80, 14:42 CMMER:IDSH FILE:MODIFY

<PAGE 9>

ISM

```

285      'MODP', 11X, 'HATE', /)
286      IF (NOC.FO.O) GOTO 162
287      JJ=NHQ*NDNF
288      LJ=NRQ*1
289      DO 162 I=1,JJ
290      WRITE(6,101)NHAM(I),LD(I,1),NHAM(I),LD(I,2)
291      FORMAT(15X,A8,7X,'FEF',11X,F8.6,21X,'PER',12X,A8,6X,F8.6)
292      CONTINUE
293      IF (JJ.FO.NR) GOTO 161
294      WRITE(6,104)
295      JJ=JJ*1
296      ISPACE=IP(JJ,3)
297      DO 163 I=JJ,NR
298      IF (LP(I,3).NE.ISPACE) WRITE(6,104)
299      ISPACE=I(I,3)
300      WRITE(6,302)NHAM(I),NHAM(LP(I,3)),LD(I,1),NHAM(LP(I,3)),
301      NHAM(I),I(I,2)
302      FORMAT(15X,A8,7X,A8,6X,F8.6,21X,A8,7X,A8,6X,F8.6)
303      CONTINUE
304      IF (NHC.FO.O) GOTO 164
305      WRITE(6,304)
306      DO 164 I=1,NHC
307      WRITE(6,301)NHAM(I),LD(I,1),NHAM(I),LD(I,2)
308      CONTINUE
309      WRITE(6,315)
310      NHAM(I,1)
311      FORMAT(' ')
312      C
313      DO 200 WRITE(6,50)
314      CALL PRFAC(5,'S:',ANSWER,1)
315      IF (ANSWER.FO.YES) GO TO 900
316      RETURN
317      END

```

<PAGE 9>

///// FILE:MODIFY /////

>>>> SUBROUTINE NOISP <<<<

<PAGE 9>

<PAGE 10>

278
279
280

SUBROUTINE PROC
RETURN
END

DATE: 02-05-80, 14:47 CYNFB: TCSH FILE: MODIFY

<PAGE 10>

ISN

1
2
3

<PAGE 12>

////// FILE: MODIFY //

>>>> SUBROUTINE PROC <<<<

<PAGE 10>

LINE	STATEMENT	DATE: 02-05-80, 14:42	CANAL: 105H	FILE: MODIFY	ISN
281	1				1
282	2				2
283	3				3
284	4				4
285	5				5
286	6				6
287	7				7
288	8				8
289	9				9
290	10				10
291	11				11
292	12				12
293	13				13
294	14				14
295	15				15
296	16				16
297	17				17
298	18				18
299	19				19
300	20				20
301	21				21
302	22				22
303	23				23
304	24				24
305	25				25
306	26				26
307	27				27
308	28				28
309	29				29
310	30				30
311	31				31
312	32				32
313	33				33
314	34				34
315	35				35
316	36				36
317	37				37
318	38				38
319	39				39
320	40				40
321	41				41
322	42				42
323	43				43
324	44				44
325	45				45
326	46				46
327	47				47
328	48				48
329	49				49
330	50				50
331	51				51
332	52				52
333	53				53
334	54				54
335	55				55
336	56				56
337	57				57
338	58				58
339	59				59
340	60				60

<PAGE 1>

ISN

DATE: 02-05-80, 14:42 CMMER: 105N FILE: CCMFUTE

*** TOS ANALYSIS PACKAGE PROGRAM NUMBER 4: CCMFUTE *****
PROGRAM COMPUTES SUMMARY STATISTICS AND PERFORMS
USF SPECIFIED OUTPUT FORMATTING

GM VRI ARI-3 1/7/79

REAL*8 RMAN(50), CMAN(10), NMAN(50)
REAL*8 LARG, RM(50,50,4), LD(50,2), MD(50,4), CD(10,4),
\$ RM(50,50), MC(60,1), MOUT(50,3), POUT(50,3), POUT(50,6),
\$ MON1,T(7,2,2), CS(10), GPD(11,2), POUT(1,5), CONT(10,6),
INTERP C1,C2,PD(50),MP(50,2),LPD(50), LP(50,3), MP(50,2)

INPUT AND INITIALIZATION MODULE BEGINS

USF INPUTS THE BLOCKING NUMBER AND RETAINED MESSAGE COPY SWITCH

WRITE(*,990)
FORMAT(' INPUT BLOCKING: NUMBER AND RMC SWITCH (0=>USF RMC, 1=>DO NOT)')

CALL MTS FREE FORMAT READ SUBROUTINE

CALL FREAD(6,'21:',MULOCK,IRMC)

READ INPUT FILE

FREAD(7,990) NR,NR,NM,NM,NM,MC,MP,ML
DO 30 I=1,NR

READ(7,901) RMAN(I),RP(I,1),ME(I,2),LPD(I)

CONTINUE

FORMAT(A9,3(1X,13))

DO 31 I=1,NM

READ(7,902) NMAN(I),MP(I,1), MP(I,2),MD(I,1),MD(I,2),MD(I,3),ME(I,4)

PT(I,2)=MP(I,2)/60000.

CONTINUE

FORMAT(A9,2(1X,13),4(1X,E9.3))

DO 32 I=1,MC

FREAD(7,904) CMAN(I),CD(I,1),CD(I,2),CD(I,3),CD(I,4)

CI(I,1)=CI(I,1)/60.

CI(I,2)=MULOCK/(60*CD(I,2))

CONTINUE

FORMAT(A9,4(1X,E9.3))

DO 33 I=1,MP

FREAD(7,905) PD(I)

CONTINUE

FORMAT(2X,13)

DO 36 I=1,NL

FREAD(7,906) LP(I,1),LP(I,2),LP(I,3),LP(I,1),LP(I,2)

CONTINUE

FORMAT(3(1X,1X),2(1X,6,1X))

DO 37 I=1,NR

DO 38 J=1,NM

FREAD(7,907) RM(I,J,1),RM(I,J,2),RM(I,J,3),RM(I,J,4)

IF (J.EC.NM) RM(I,J,1)=RM(I,J,1)/60

CONTINUE

FORMAT(4(E9.3,1X))

////// FILE:CCMUTE //

>>>> MAIN PROGRAM <<<<

<PAGE 1>

<PAGE 1>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

19.2

19.4

19.6

20

21

22

23

24

25

26

27

28

29

30

301

31

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

```

<PAGE 2>
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113

DATE:02-05-80,14:42 CUSER:IOSH FILE:COMPUT
ISN
39
40
41
42
43
44
45
46
47

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

82
83
84

<PAGE 2>

```

```

<PAGE 3>                                     <PAGE 3>
I DATE:02-05-80,14:42 CHANNEL:IDSH FILE:COMPUTE I
ISM
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

C POINTERS TO THE MESSAGE TYPE OF RESPONSES ON THIS ROUTE
C TO THIS TYPE OF USER MESSAGE
C
C      I1=RP(I,1)
C
C POINTER TO THE MESSAGE TYPE OF RESPONSES ON ALL OTHER ROUTES
C TO THIS TYPE OF USER MESSAGE
C
C      I2=RP(1,2)
C
C INCREMENT THE RATE OF OUTPUTS OF THE APPROPRIATE TYPE ON THE
C INCORPORATING ROUTE
C
C      IF (I1.EQ.(0)) GOTO 17
C      RMO(J,I1)=RMC(J,I1)*RM(J,I,1)*(1.-RM(J,I,1))
C      CCNTIMEP
C
C INCREMENT RATES OF OUTPUTS OF THE APPROPRIATE TYPE ON ALL OTHER ROUTES
C
C      IF (I2.EQ.(0)) GOTO 13
C      TO 14 N=1,NH
C      IF (N.EQ.J) GOTO 14
C      RMO(K,I2)=RMO(K,I2)*RMO(J,I,1)*RM(K,I2,2)*(1.-RM(J,I,1))
C      CCNTIMEP
C      CONTINUE
C
C TRAFFIC FLOW MODULE ENDS
C OPERATING STATISTICS MODULE BEGINS
C
C COMPUTE THE AGGREGATE ARRIVAL RATES FROM THE ARRIVAL RATES
C TO EACH ROUTE
C
C DO 21 I=1,NR
C     TO 22 J=1,RH
C
C POINTERS TO THE LINK(S) ON THE ROUTE
C
C      I1=RP(I,1)
C      I2=RP(1,2)
C
C SET POINTERS TO THE NODES AND CHANNEL OF THE FIRST LINK,
C AND SET THE APPROPRIATE ARRIVAL RATES
C
C      M1=LP(I1,1)
C      C1=LP(I1,2)
C      M3=LP(I1,3)
C      M2=RP+C1
C      JPE=TPE(M1)
C
C COMPUTE THE RATE AT WHICH MESSAGES OF TYPE J TRAVEL TO
C OR FROM USER I
C
C      A=RMO(1,J)*(1.+RN(I,J,3)+RN(I,J,4))
C
C COMPUTE THE RATE AT WHICH MESSAGES OF TYPE J TRAVEL TO
C OR FROM THE ECC TO USER I
C

```

<PAGE 1>

>>>> MAIN PROGRAM <<<<

////// FILE:COMPUTR. //

<PAGE 1>

```

172 P=MOD(1,3)*(1-16*(L,J,3))
173
174 SKIP IF NO SECOND LINK, ELSE SET POINTERS TO NODES AND CHANNEL
175 OF THE SECOND AND SUM THE APPROPRIATE ARRIVAL RATES
176
177
178 IF (12,PG,(0)) GOTO 23
179 C2=LP(12,2)
180 M5=LP(12,3)
181 M4=M2+C2
182 CONTINUE
23
183
184 ASSIGN SOME TEMPORARY PARAMETERS
185
186 MESSAGE LENGTH
187 MCJ1=MCJ1,1)
188
189 PROCESSOR CPU TIMES FOR TASKS
190
191 ORIGINATE MESSAGE OF LENGTH = MCJ1 AT A TCS
192 T(1,1,1)=GPD(1,1)*GPD(1,2)*MDJ1
193 ORIGINATE MESSAGE OF LENGTH = 16 CHARACTERS AT A TCS
194 T(1,2,1)=GPD(1,1)*GPD(1,2)*16
195 SEND MESSAGE OF LENGTH = MCJ1 FROM A TCS
196 T(2,1,1)=GPD(2,1)*GPD(2,2)*MDJ1
197 SEND MESSAGE OF LENGTH = 16 CHARACTERS FROM A TCS
198 T(2,2,1)=GPD(2,1)*GPD(2,2)*16
199 RECEIVE MESSAGE OF LENGTH = MCJ1 AT A TCS
200 T(1,1,1)=GPD(1,1)*GPD(1,2)*MDJ1
201 RECEIVE MESSAGE OF LENGTH = 16 CHARACTERS AT A TCS
202 T(1,2,1)=GPD(1,1)*GPD(1,2)*16
203 CHECK A MESSAGE OF LENGTH = MCJ1 AGAINST ONE TEMPLATE AT A TCS
204 T(4,1,1)=GPD(4,1)*GPD(4,2)*MDJ1
205 TERMINATE A MESSAGE OF LENGTH = MCJ1 AT A TCS
206 T(5,1,1)=GPD(5,1)*GPD(5,2)*MDJ1
207 TERMINATE AT THE REP A MESSAGE OF LENGTH = MCJ1
208 T(6,1,1)=GPD(6,1)*GPD(6,2)*MDJ1
209 ORIGINATE A MESSAGE OF LENGTH = MCJ1 AT A TCT
210 T(1,1,2)=GPD(1,1)*GPD(1,2)*MDJ1
211 ORIGINATE A MESSAGE OF LENGTH = 16 CHARACTERS AT A TCT
212 T(1,2,2)=GPD(1,1)*GPD(1,2)*16
213 SEND A MESSAGE OF LENGTH = MCJ1 FROM A TCT
214 T(2,1,2)=GPD(2,1)*GPD(2,2)*MDJ1
215 SEND A MESSAGE OF LENGTH = 16 CHARACTERS FROM A TCT
216 T(2,2,2)=GPD(2,1)*GPD(2,2)*16
217 RECEIVE A MESSAGE OF LENGTH = MCJ1 AT A TCT
218 T(3,1,2)=GPD(3,1)*GPD(3,2)*MDJ1
219 RECEIVE A MESSAGE OF LENGTH = 16 CHARACTERS AT A TCT
220 T(3,2,2)=GPD(3,1)*GPD(3,2)*16
221 TERMINATE A MESSAGE OF LENGTH = MCJ1 AT A TCT
222 T(4,1,2)=GPD(4,1)*GPD(4,2)*MDJ1
223 CHECK A MESSAGE AGAINST A TEMPLATE AT A TCT
224 T(4,1,2)=0
225 ORIGINATE A MESSAGE OF LENGTH = MCJ1 AT THE REP
226 T(7,1,1)=GPD(1,1)*GPD(1,2)*MDJ1
227
228 COMPUTE THE FIRST AND SECOND MOMENTS
229 OF THE DISTRIBUTION OF THE NUMBER OF TRANSMISSIONS

```

```

PAGE 55
I DATE:02-05-80,14:42 SOURCE:IOSH FILE:COMPUTE
ISM
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

<PAGE 6>
1 DATE: 02-05-80, 10:42  CYNEX: 105P  FILE: CONDUIT
1
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

<PAGE 6>

>>>> MAIN PROGRAM <<<<

///// FILE: CONDUIT /////

<PAGE 6>

```

<PAGE 7>
1 DATE:02-05-80,14:47 CYNER:IOSH FILE:COMPUTF
1
1SM
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

NO	CONTINUE	COMPUTATIONS FOR MULTI-SERVER CHANNELS	NO	CONTINUE	COMPUTATIONS FOR MULTI-SERVER CHANNELS
21	C	DO 29 I=1,NC	21	C	DO 29 I=1,NC
22	C	RC(NC+I,1)=NO(NC+I,1)/CG(I,M)	22	C	RC(NC+I,1)=NO(NC+I,1)/CG(I,M)
23	C	IF (RC(NC+I,1) * LI(I,3))	23	C	IF (RC(NC+I,1) * LI(I,3))
24	C	MS=IPTR(CD(I,M))	24	C	MS=IPTR(CD(I,M))
25	C	FACT=1.	25	C	FACT=1.
26	C	SP4=1.	26	C	SP4=1.
27	C	DO 129 J=1,MS	27	C	DO 129 J=1,MS
28	C	FACT=FACT*J	28	C	FACT=FACT*J
29	C	SUP=SUM*(MSGOUT)**J/FACT	29	C	SUP=SUM*(MSGOUT)**J/FACT
30	C	CONTINUE	30	C	CONTINUE
31	C	SUM=SUM*OUTS**MS/(FACT*(1.-UT))	31	C	SUM=SUM*OUTS**MS/(FACT*(1.-UT))
32	C	CS(I)=1./SUM	32	C	CS(I)=1./SUM
33	C	CONTINUE	33	C	CONTINUE
34	C	COMPUTE COEFFICIENT OF VARIATION FOR EACH MODE	34	C	COMPUTE COEFFICIENT OF VARIATION FOR EACH MODE
35	C	DO 111 I=1,MH	35	C	DO 111 I=1,MH
36	C	IF (NO(I,1)-PO-(0.)) GOTO 111	36	C	IF (NO(I,1)-PO-(0.)) GOTO 111
37	C	X=NO(I,1)*NO(I,2)/NO(I,1)**2-1.	37	C	X=NO(I,1)*NO(I,2)/NO(I,1)**2-1.
38	C	IF (X-LT-(1./LARG)) X=0.	38	C	IF (X-LT-(1./LARG)) X=0.
39	C	NO(I,3)=SQRT(X)	39	C	NO(I,3)=SQRT(X)
40	C	CONTINUE	40	C	CONTINUE
41	C	OPERATING STATISTICS MODIFIER ENDS	41	C	OPERATING STATISTICS MODIFIER ENDS
42	C	PERFORMANCE MEASURE MODIFIER BEGINS	42	C	PERFORMANCE MEASURE MODIFIER BEGINS
43	C	COMPUTE FCC COMPONENT SUMMARY STATISTICS	43	C	COMPUTE FCC COMPONENT SUMMARY STATISTICS
44	C	DO 110 I=1,I	44	C	DO 110 I=1,I
45	C	MI=MI+1	45	C	MI=MI+1
46	C	IF (NO(MI,2)-PO-(0.)) GOTO 110	46	C	IF (NO(MI,2)-PO-(0.)) GOTO 110
47	C	EXPECTED DELAY (MIN)	47	C	EXPECTED DELAY (MIN)
48	C	FCUT(I,1)=NO(MI,1)*NO(MI,1)*(1.-NO(MI,3)**2)/(2*NO(MI,2)*(1.-NO(MI,3)))+NO(MI,1)/NO(MI,2)	48	C	FCUT(I,1)=NO(MI,1)*NO(MI,1)*(1.-NO(MI,3)**2)/(2*NO(MI,2)*(1.-NO(MI,3)))+NO(MI,1)/NO(MI,2)
49	C	EXPECTED QUEUE LENGTH	49	C	EXPECTED QUEUE LENGTH
50	C	FCUT(I,2)=NO(MI,1)*(1.-NO(MI,3)**2)/(2*(1.-NO(MI,3)))	50	C	FCUT(I,2)=NO(MI,1)*(1.-NO(MI,3)**2)/(2*(1.-NO(MI,3)))
51	C	TRAFFIC RATE (MSG/HR)	51	C	TRAFFIC RATE (MSG/HR)
52	C	FCUT(I,3)=60*NO(MI,2)	52	C	FCUT(I,3)=60*NO(MI,2)
53	C	UTILIZATION	53	C	UTILIZATION
54	C	FCUT(I,M)=NO(MI,1)	54	C	FCUT(I,M)=NO(MI,1)
55	C	CAPACITY OF FCC COMPONENTS (MSG/HR)	55	C	CAPACITY OF FCC COMPONENTS (MSG/HR)
56	C	FCUT(I,5)=NO*FCUT(I,3)/FCUT(I,M)	56	C	FCUT(I,5)=NO*FCUT(I,3)/FCUT(I,M)
57	C	MAKE ADJUSTMENTS IF UTILIZATION >= 1	57	C	MAKE ADJUSTMENTS IF UTILIZATION >= 1
58	C	MAKE ADJUSTMENTS IF UTILIZATION >= 1	58	C	MAKE ADJUSTMENTS IF UTILIZATION >= 1

```

<PAGE 9>
110
C
462 IF (NO(N1,1)-LT.(1.)) GOTO 110
463 FCNT(1,1)=1ARG
464 FCNT(1,2)=1ARG
465 FCNT(1,2)=1ARG
466 CONTINUE
467
C
468 DOCC COMPONENT SUMMARY STATISTICS COMPUTATIONS ARE COMPLETED
469
C
470 CCRUTP CHANNEL SUMMARY STATISTICS
471
C
472 NO 120 I=1,NC
473 NI=NPAT
474 IF (NO(N1,2)-EQ.(0.)) GOTO 120
475 NON1=NO(N1,1)*CD(L,I)
476
C
477 EXPECTED DELAY AT CHANNEL (PIE)
478
C
479  $CD(L,1) = NO(N1,1) * CD(L,4) / NO(N1,2) + CS(1) * NON1 * (1. + NO(N1,3) * 2) / (2 * NO(N1,2) * CD(L,4) * (1. - NON1) * 2)$ 
480
C
481 EXPECTED QUEUE LENGTH
482
C
483  $CCNT(L,2) = CS(1) * NON1 * (1. + NO(N1,3) * 2) / (2 * (1. - NON1) * 2)$ 
484
C
485 TRAFFIC RATE (MSG/HR)
486
C
487  $CCNT(L,1) = 60 * NO(N1,2)$ 
488
C
489 UTILIZATION
490
C
491  $CCNT(L,4) = NON1$ 
492
C
493 CAPACITY OF CHANNELS (MSG/HR)
494
C
495  $CCNT(L,5) = 4 * CCNT(L,1) / CCNT(L,4)$ 
496
C
497 MAKE ADJUSTMENTS IF UTILIZATION >= 1
498
C
499 IF (NON1.LT.(1.)) GOTO 120
500 FCNT(1,1)=1ARG
501 FCNT(1,2)=1ARG
502 CONTINUE
503
C
504 CHANNEL SUMMARY STATISTICS ARE COMPUTE
505
C
506 COPEPTE PROCESSOR SUMMARY STATISTICS
507
C
508 DO 170 I=1,NP
509 IF (NO(I,2)-EQ.(0.)) GOTO 170
510
C
511 EXPECTED DELAY (PIE)
512
C
513  $FCNT(I,1) = NO(I,1) / NO(I,2) + NO(I,1) * NO(I,3) * 2 / (2 * NO(I,2) * (1. + NO(I,1)))$ 
514
C
515 EXPECTED QUEUE LENGTH
516
C
517  $FCNT(I,2) = NO(I,1) * (1. + NO(I,3) * 2) / (2 * (1. - NO(I,1)))$ 
518
C
519 TRAFFIC RATE (MSG/HR)
520
C
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

<PAGE 9>

>>>> MAIN PROGRAM <<<<

///// FILE:COMPUTE /////

<PAGE 9>

```

<PAGE 10>
I DATE:02-05-80,10:42  OWNER:IDSH  FILE:COMPUT  I
15M
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```


[illegible]

```

<PAGE 14>
1 DATE:02-05-80, 14:42 CNAME:IOSU FILE:COMPUP 1
ISM

```

WRITE PROCESSOR OUTPUT TABLE

```

752 C
753 C
754 C
755
756 WRITE(8,970)
757 WRITE(8,924)
758 WRITE(8,924)
759 WRITE(8,924)
760 WRITE(8,924)
761 WRITE(8,924)
762 WRITE(8,924)
763 WRITE(8,924)
764 WRITE(8,924)
765 WRITE(8,924)
766 WRITE(8,924)
767 WRITE(8,924)
768 WRITE(8,924)
769 WRITE(8,924)
770 WRITE(8,924)
771 WRITE(8,924)
772 WRITE(8,924)
773 WRITE(8,924)
774 WRITE(8,924)
775 WRITE(8,924)
776 WRITE(8,924)
777 WRITE(8,924)
778 WRITE(8,924)
779 WRITE(8,924)
780 WRITE(8,924)
781 WRITE(8,924)
782 WRITE(8,924)
783 WRITE(8,924)
784 WRITE(8,924)
785 WRITE(8,924)
786 WRITE(8,924)
787 WRITE(8,924)
788 WRITE(8,924)
789 WRITE(8,924)
790 WRITE(8,924)
791 WRITE(8,924)
792 WRITE(8,924)
793 WRITE(8,924)
794 WRITE(8,924)
795 WRITE(8,924)
796 WRITE(8,924)
797 WRITE(8,924)
798 WRITE(8,924)
799 WRITE(8,924)
800 WRITE(8,924)
801 WRITE(8,924)
802 WRITE(8,924)
803 WRITE(8,924)
804 WRITE(8,924)
805 WRITE(8,924)
806 WRITE(8,924)
807 WRITE(8,924)
808 WRITE(8,924)
809 WRITE(8,924)

```

***** TABLE 3: PROCESSOR SUMMARY STATISTICS *****

```

901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****

```

EXPECTED EXCEEDED TRAFFIC

```

970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****

```

DELAY QUEUE RATE CAPACITY

```

1000 *****
1001 *****
1002 *****
1003 *****
1004 *****
1005 *****
1006 *****
1007 *****
1008 *****
1009 *****
1010 *****
1011 *****
1012 *****
1013 *****
1014 *****
1015 *****
1016 *****
1017 *****
1018 *****
1019 *****
1020 *****
1021 *****
1022 *****
1023 *****
1024 *****
1025 *****
1026 *****
1027 *****
1028 *****
1029 *****
1030 *****
1031 *****
1032 *****
1033 *****
1034 *****
1035 *****
1036 *****
1037 *****
1038 *****
1039 *****
1040 *****
1041 *****
1042 *****
1043 *****
1044 *****
1045 *****
1046 *****
1047 *****
1048 *****
1049 *****
1050 *****
1051 *****
1052 *****
1053 *****
1054 *****
1055 *****
1056 *****
1057 *****
1058 *****
1059 *****
1060 *****
1061 *****
1062 *****
1063 *****
1064 *****
1065 *****
1066 *****
1067 *****
1068 *****
1069 *****

```

LENGTH (MSG/HR) UTILIZATION (MSG/HR) RANK

```

1070 *****
1071 *****
1072 *****
1073 *****
1074 *****
1075 *****
1076 *****
1077 *****
1078 *****
1079 *****
1080 *****
1081 *****
1082 *****
1083 *****
1084 *****
1085 *****
1086 *****
1087 *****
1088 *****
1089 *****
1090 *****
1091 *****
1092 *****
1093 *****
1094 *****
1095 *****
1096 *****
1097 *****
1098 *****
1099 *****

```

IF (IC,EO,(0)) GOTO 244

```

1100 IF 245,1=1,NP11
1101 IF (IFIX(POINT(J,6)),FO,1) K=J
1102 CONTINUE
1103 CONTINUE
1104 WRITE(8,905) NAME(K),POINT(K,1),POINT(K,2),POINT(K,3),POINT(K,4),POINT(K,5),POINT(K,6)
1105 IF (FIX(1/5),AE,1/5,.) GOTO 243
1106 WRITE(8,924)
1107 CONTINUE
1108 FORMAT(' ',AR,4(1X,FO,1),2X,2(J1,PA,1))
1109 GOTO 249

```

CHAPTER SUMMARY STATISTICS TABLE

```

1110 RANK CHANNELS
1111 CONTINUE
1112 DO 261 I=1,NC
1113 X=CC(1,I,IP)
1114 IF=0
1115 DO 262 J=1,NC
1116 IF (COUT(J,10),GT,X) IP=IP+1
1117 IF ((COUT(J,10),EQ,X).AND.(J,1E-1)) IP=IP+1
1118 CONTINUE
1119 COUT(I,6)=IP
1120 CONTINUE

```

WRITE CHANNEL OUTPUT TABLE

```

1121 WRITE(8,970)
1122 WRITE(8,924)
1123 WRITE(8,924)

```

<PAGE 14>

<PAGE 15>

 I DATE: 02-05-90, 14:42 CNAME: JDSH FILE: COMPUTE I
 I-----I

<PAGE 15>

LSM

 910
 911
 912
 913
 914
 915
 916
 917
 918
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967

 WRITE(9,924)
 WRITE(9,936)
 WRITE(9,937)
 WRITE(9,961)
 WRITE(9,917)
 WRITE(9,916)
 FORMAT('')
 WRITE(9,924)
 WRITE(9,924)
 WRITE(9,924)
 WRITE(9,942)
 WRITE(9,943)
 WRITE(9,944)
 FORMAT(' CHANNEL
 WRITE(9,924)
 DO 263 I=1,NC
 K=I
 IF (IC.EQ.(0)) GOTO 264
 IF 265 J=1,NC
 IF (IEX(COUT(J,6)).EQ.I) K=J
 CONTINUE
 CONTINUE
 WRITE(9,945) CHAN(K),COUT(K,1),COUT(K,2),COUT(K,3),COUT(K,4),COUT(K,5),COUT(K,6)
 IF (FLOAT(I/5).NE.1/5.) GOTO 263
 WRITE(9,924)
 CONTINUE
 GOTO 949

***** TABLE 4: CHANNEL SUMMARY STATISTICS *****

	(MIN)	LENGTH	(MSG/HR)	UTILIZATION	(MSG/HR)	RANK
--	-------	--------	----------	-------------	----------	------

***** TABLE 5: DCC COMPONENT SUMMARY STATISTICS *****

	EXPECTED	EXPECTED	QUEUE	RATE	UTILIZATION	(MSG/HR)	'
--	----------	----------	-------	------	-------------	----------	---

<PAGE 15>

///// FILE: COMPUTE /////

>>>> MAIN PROGRAM <<<<

<PAGE 15>

<PAGE 16>

DATE: 02-05-80, 14:42 CMMES: 10:50 FILE: COMPTIF

<PAGE 16>

```

656      CONTINUE, 44, 5 (F4, 3, 54)
657      WRITE(10, 107) ECUT(44, 1), POUT(MI, 2), ECUT(MI, 2), POUT(MI, 5)
658      CONTINUE, 44, 5 (F4, 3, 54)
659      GO TO 295
660
661      C      TERMINATE
662      C
663      C      CONTINUE
664      C      STOP
665      C      END

```

<PAGE 16>

DATE: 02-05-80, 14:42 CMMES: 10:50 FILE: COMPTIF

<PAGE 16>

```

656      CONTINUE, 44, 5 (F4, 3, 54)
657      WRITE(10, 107) ECUT(44, 1), POUT(MI, 2), ECUT(MI, 2), POUT(MI, 5)
658      CONTINUE, 44, 5 (F4, 3, 54)
659      GO TO 295
660
661      C      TERMINATE
662      C
663      C      CONTINUE
664      C      STOP
665      C      END

```

<PAGE 16>

>>>> MAIN PROGRAM <<<<

///// FILE: COMPTIF /////

<PAGE 16>

<PAGE 17>

1 DATE: 02-05-80, 14:42 CMMER: 105H FILE: COMPTF 1

<PAGE 17>

ISM

```

878 SUBROUTINE TO COMPUTE THE FIRST AND SECOND MOMENTS OF THE
879 DISTRIBUTION OF THE NUMBER OF TRANSMISSIONS TO SUCCESS.
880
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C

```

FM = FIRST MOMENT
 SM = SECOND MOMENT
 EQ = ERROR RATE AFTER MAJORITY AND BEFORE HAMMING CODE
 L = NUMBER OF CHARACTERS IN THE MESSAGE
 NB = PICKING NUMBER
 TRAC = FRC SWITCH, 0=>ON
 SCRATCH PAD
 PRCM = THE PROBABILITY THAT ALL COPIES OF A GIVEN CHARACTER
 IN A TRANSMISSION (NB CHARACTERS) EACH HAVE 2 OR MORE
 BITS IN ERROR
 FSA = FRC SWITCH, 0 OR 1
 EN = COUNTER
 PGLTM = PROBABILITY OF SUCCESS IN LESS THAN N TRIES
 PGLFM = PROBABILITY OF SUCCESS IN LESS THAN OR EQUAL TO N TRIES
 PACK = PROBABILITY OF A ACK TO A TRANSMISSION OF A MESSAGE

GW ADI-3 1/7/79

```

SUBROUTINE TRANS(FM,SM,PR,L,RD,TRAC)
  PRCM=(1-((1-FR)**12+12*FR*(1-FR)**11))*ND
  PACK=(1-PRCM)**L
  FM=10.**10
  IF (EACK.GT.(10.**(-10))) FM=1/PACK
  SM=2**10*FM-FM
  IF (11*PRC.EQ.(0)).AND.(FM.LT.(10.**10))) GOTO 200
  RPTURN
  EN=1.0
  PGLTM=0.0
  PGLFM=0.0
  FM=0.0
  SM=0.0
  CONTINUE
  PGLER=(1-PRCM**EN)**L
  FM=FM+FM*(PGLFM-PGLTM)
  SM=SM+EN*FM*(PGLFM-PGLTM)
  PGLTM=PGLER
  FM=FM+1
  IF (EGLFM.LT.(0.999)) GOTO 100
  RETURN
END

```

200
 100

<PAGE 17>

///// FILE:COMPTF /////

>>>> SUBROUTINE TRANS <<<<

<PAGE 17>

APPENDIX B: MERGED VARIABLE CROSS-REFERENCE LISTING

This appendix contains a merged variable cross-reference listing for all of the programs which make up the TOS Simulation Package. These programs are CREATE, DISPLAY, MODIFY, and COMPUTE. The merged cross-reference lists all the occurrences of each variable in the programs which make up the TOS Simulation Package.

Referencing of statements that a variable is used in is done using internal statement numbers (see the discussion of internal statement numbers in the previous description of appendix A.) Since internal statements are numbered beginning at one within each routine, each reference to internal statement numbers also requires specification of the routine in which the statement is located. This is accomplished by first listing the variable being cross-referenced, then the routine in which occurrences of this variable appear, and to the far right, a list of internal statement numbers referring to statements in this program module at which the variable is referenced. If the variable is referenced multiple times within a single statement, only one reference appears in the cross-reference listing. Some internal statement references in the listing may be immediately followed by a symbol indicating the usage of the variable in this statement. The following key defines the usage of these symbols:

- * A variable or a function is changed either through an assignment READ, ASSIGN statement, or is used as a DO index.
- ? A variable may be changed because it is used as a simple argument to a subroutine or function.
- D A subprogram is defined by the SUBROUTINE, FUNCTION, ENTRY, or EXTERNAL statement. A statement function is defined by the statement function definition. A variable is declared in a

type or DIMENSION statement. For units, "D" stands for DEFINE FILE statements.

E A variable appears in an EQUIVALENCE statement.

C A variable appears in a COMMON statement.

These suffixes are especially useful since they allow distinguishing the use of variables in declarations from their use in executable statements, and, in the latter case, the distinguishing between just the referencing of a variable versus its updating. These distinctions can be made, for the most part, just from the suffixes, without the need for looking up the statement in the program listing.

Certain other useful pieces of information are also provided in these cross-reference listings. The type of each variable is indicated. The list of variable types included are "I*4" for integer variables, "R*4" for real variables, "L*4" for logical variables, and "CHAR" for character variables. The field in the cross-reference listing with the heading "ATTR" distinguishes between scalars and arrays. This field is left blank for scalars while it contains "ARRAY" for arrays. It has been found useful to include references to subroutines and functions in these cross-reference listings, in addition to references to just variables. These are distinguished from variables by having "SUBR" or "FCN" in the ATTR field. Another type of item included in these cross-references are RETURN or STOP statements within each program module. These are referenced in the listing using <EXIT> as the name of the "variable".

***** REFUTED VARIABLE CROSS REFERENCES *****				*****			
NAME	ROUTINE	TYPE	ATTR	CORREL	REFERENCES		
<EXIT>	CHECK			4	5		
	CHECK1		SUBR	7			
	CHECK2		SUBR	9	17		
	CHECK3		SUBR	6			
	CHECK4		SUBR	6			
	COMPUTE		SUBR	542			
	CONSYS		SUBR	25	54		
	DISP		SUBR	239			
	LOACT		SUBR	20			
	MODIFY		SUBR	55			
A	MOUSE		SUBR	23	74		
	READ		SUBR	47			
	TRANS		SUBR	8	21		
	WRITE		SUBR	47			
	COMPUTE	(R+4)		107*	145	153	166
	COMPUTE	(R+4)		140*	144	146	147
	COMPUTE			169*	171	173	174
	COMPUTE			192*	194	196	197
	COMPUTE			218*	220	222	223
	COMPUTE	(R+4)		141*	144	146	150*
AP1	COMPUTE			208			
	COMPUTE	(R+4)		161*	165	167	168
	CONSYS			2C	352	36	522
	MODIFY	T+4		110	272	28	312
	MODIFY			48	512	52	512
	MOUSE	T+4		30	362	37	722
	COMPUTE			108*	171	172	173
	CREATE	T+4		50	242	252	1002
	CREATE	R+4		20	212	22	26
	CREATE			192	650	653	
B	CREATE	T+4		50	402	412	1152
	CREATE			20	382	39	75
	CREATE	R+4		3C	20*	21*	22*
	COMPUTE			209	210	217	234
	COMSYS	(R+4)		90	15C	123	46
	CREATE	(R+4)		60	5432	5462	5492
	DISP	(R+4)		1	50	60	61
	DISP	(R+4)		50	17*	95	96
	READ	(R+4)		70	13C	25*	97
	WRITE	(R+4)		70	13C	25	
C	CHECK		SUBR	10			
	CREATE		SUBR	25	41	56	101
	CHECK1		SUBR	10			
	CREATE		SUBR	78	85	93	134
	CHECK2		SUBR	1C			
	CREATE		SUBR	264	268	309	310
	CHECK3		SUBR	10			
	CREATE		SUBR	349	353	399	402
	CHECK4		SUBR	10			
	CREATE		SUBR	621	630		
CHAM	CREATE	R+4	PERAY	1	24	5	
	COMPUTE	R+4	ATRAY	2C	20*	505	
	CONSYS	R+4	ATRAY	5C	11C	11	48
	CREATE	R+4	ATRAY	2C	662	782	852
	CREATE		ATRAY	547	618	627	6722
	DISP	R+4	ATRAY	1	20	19	22

[illegible]

IA	COMPUTE (1.00)	1332	334	345	342	384	387	388	429	432	433	473	476
IP	COMPUTE (1.00)	1332	338	341									
IC	COMPUTE (1.00)	477	332	415	460	500							
IC	DISP (1.00)	121	48										
IC	COMES (1.00)	46*	62										
IC	DISP (1.00)	95*	97										
IC	CREATE (1.00)	1302	331	333	6227	677	699	701					
IC	CREATE (1.00)	396	397	398	399	400	401	402	405	407	408	409	410
IC	CREATE (1.00)	413	414*	415	644*	649*	652	655					
IC	DISP (1.00)	1	11*	121									
IC	DISP (1.00)	6	28	30	148	150							
IC	COMPUTE (1.00)	239	171	417	462	502							
IC	DISP (1.00)	13*	17	88*	89	173*	175						
IC	DISP (1.00)	50*	54	122*	123	200*	202						
IC	CREATE (1.00)	34*	35	77*	78	111*	112	203*	204	647*	648		
IC	CREATE (1.00)	248*	249	276*	299*	300	432*	433					
IC	DISP (1.00)	169*	176	190	199	213*	214						
IC	DISP (1.00)	196*	201	216	224	238*	239						
IC	DISP (1.00)	48*	49										
IC	DISP (1.00)	71*	76	78	80	82	90*	93					
IC	DISP (1.00)	100*	111	113	115	117	124*	127					
IC	DISP (1.00)	78*	76	78	80	82	91*	93					
IC	DISP (1.00)	109*	111	113	115	117	125*	127					
IC	DISP (1.00)	75*	76	78	80	82	92*	93					
IC	DISP (1.00)	110*	111	113	115	117	126*	127					
IC	CREATE (1.00)	20	2612	2642	2672	2682	1082	1092	1102				
IC	CREATE (1.00)	139*	141*	142*	344	385*	387*	390					
IC	COMPUTE (1.00)	478*	476*	477*	479								
IC	COMPUTE (1.00)	40	11*	106									
IC	CREATE (1.00)	60	105*	198*	212*	218	219*	220*	225	226	6727	680	
IC	DISP (1.00)	1	50	18	33	45							
IC	DISP (1.00)	50*	54	55	70	91							
IC	READ (1.00)	70	130	17*									
IC	WRITE (1.00)	70	130	17									
IC	DISP (1.00)	133*	134	149*	150	162*	163						
IC	DISP (1.00)	162*	163	178*	179	190*	191						
IC	COMPUTE (1.00)	77	1332	1342	1362	1372							
IC	TRANS (1.00)	1	7										
IC	DISP (1.00)	29*	31	12*	144*	146	147*	186*	188	189*	224	225*	225*
IC	DISP (1.00)	66*	68	69*	173*	175	176*	212*	214	215*	246*	248	249*
IC	DISP (1.00)	56*	58	59*									
IC	CREATE (1.00)	218*	220	225*	456	6727							
IC	DISP (1.00)	222*	225*	227	400	522	6727						
IC	DISP (1.00)	1	107	226*	227	400	522	6727					
IC	DISP (1.00)	1	111										
IC	CREATE (1.00)	107	11	29	92	57	67	68	241	242	252	269	273
IC	CREATE (1.00)	314	460	484	500	550	570	592	622	641			
IC	CREATE (1.00)	6712	672										
IC	CREATE (1.00)	6757	676	89									
IC	CREATE (1.00)	86*	88										
IC	CREATE (1.00)	189*	191	94									
IC	CREATE (1.00)	87*	91	94									
IC	CREATE (1.00)	190*	191	200									
IC	CREATE (1.00)	34*	35*	36	51*	52	54*	55	59*	60	64*	65	69*
IC	COMPUTE (1.00)	70	74*	75	79*	80	84*	85	89	93	94	99*	107
IC	COMPUTE (1.00)	108	114	119	144	146	147	152	154	155	165	167	168

[illegible]

[illegible]

APPENDIX C: STATEMENT LABEL REFERENCES

This Appendix presents statement label reference listings for each program in the analysis package. The statement labels are expressed in terms of internal statement numbers. The program listings in Appendix A can be consulted to look up any referenced statement. Within each program module, for each FORTRAN statement label which appears, four types of information are provided in the label reference listing: the label being referenced; the internal statement number where it is defined (i.e., the statement in which it is a label); a type indicator ("FMT" if it refers to a format statement, and blank otherwise); and a list of statement in the routine in which this statement label is referenced.

*** ADPL DICTIONARY -- CREATE ***

INDEX	DEPN	TYPE	REFERENCES
10	9	PMT	8
11	12	PMT	11
13	14	PMT	13
19	20	PMT	19
22	23	PMT	22
39	27	PMT	39
50	30	PMT	50
103	31	PMT	103
36	37	PMT	36
35	43	PMT	35
33	44	PMT	33
119	45	PMT	119
46	47	PMT	46
51	52	PMT	51
50	58	PMT	50
127	55	PMT	127
60	61	PMT	60
64	65	PMT	64
63	69	PMT	63
75	76	PMT	75
74	79	PMT	74
71	80	PMT	71
72	93	PMT	72
136	86	PMT	136
81	87	PMT	81
70	91	PMT	70
160	94	PMT	160
89	96	PMT	89
29	99	PMT	29
98	102	PMT	98
97	104	PMT	97
42	114	PMT	42
113	117	PMT	113
112	118	PMT	112
109	120	PMT	109
57	122	PMT	57
121	128	PMT	121
67	131	PMT	67
129	133	PMT	129
68	134	PMT	68
163	142	PMT	163
140	145	PMT	140
139	146	PMT	139
151	152	PMT	151
150	155	PMT	150
147	156	PMT	147
148	157	PMT	148
135	165	PMT	135
162	166	PMT	162
95	167	PMT	95
158	173	PMT	158
159	174	PMT	159
88	178	PMT	88
170	182	PMT	170
169	185	PMT	169
178	186	PMT	178

287	187			177	
285	196			191	
290	212			204	
295	218			201	202
303	215			198	
305	221			217	
310	226			224	
315	229			227	
320	232			231	
325	235			236	
333	241			237	
335	244			236	
340	250			249	
345	253			247	
353	254			245	289 303
355	259			258	277 314
360	262			263	
365	266			265	
370	270			257	
375	273			256	
383	281			275	
385	283			241	
390	286			245	300
395	288			244	
403	290			242	
405	292			291	
410	295			252	
415	302			298	
423	304			269	
425	307			306	
430	311			305	
435	317			273	
443	322			321	
445	326			320	
450	327			272	282 312 318
455	329			328	
460	332			311	
465	340			319	365
473	347			346	397
475	351			350	400
480	355			338	
485	356			355	427
490	360			317	345
495	361			316	354
500	363			334	
505	373			372	384 414 403
513	378			371	
515	379			370	
523	385			383	
525	390			382	
533	393			391	417
535	406			405	
540	411			404	
545	412			403	
550	422			416	
555	423			415	
563	424			392 413	
565	434			413	
573	439			411	

535	447				
543	446				
545	449				
590	451	FMT			
595	454	FMT			
600	456	FMT			
605	462	FMT			
610	465	FMT			
615	468	FMT			
620	471	FMT			
625	474				
630	478				
635	475				
640	482	FMT			
645	486	FMT			
651	489	FMT			
655	492	FMT			
660	495				
665	495				
671	501				
675	503	FMT			
680	506	FMT			
685	509				
693	511	FMT			
695	514	FMT			
700	517	FMT			
705	520	FMT			
710	524	FMT			
715	526				
721	532				
725	535	FMT			
731	539				
735	542	FMT			
740	545	FMT			
745	548	FMT			
751	551				
755	553				
760	556	FMT			
765	558				
770	560				
775	562	FMT			
780	565	FMT			
785	568	FMT			
790	571				
795	572				
800	576	FMT			
805	578				
810	579				
815	582				
820	584				
825	590	FMT			
830	593				
835	594				
840	595	FMT			
845	598				
850	603	FMT			
855	605				
860	606				
865	608	FMT			
442	441				
443	443				
450	450				
453	453				
457	457				
461	461				
464	464				
467	467				
470	470				
476	476				
475	475				
473	473				
481	481				
485	485				
488	488				
491	491				
490	490				
496	496				
494	494				
502	502				
505	505				
460	460				
510	510				
511	511				
516	516				
519	519				
523	523				
494	494				
500	500				
524	524				
508	508				
511	511				
544	544				
547	547				
580	580				
550	550				
555	555				
554	554				
552	552				
561	561				
564	564				
567	567				
560	560				
570	570				
575	575				
574	574				
572	572				
581	581				
580	580				
589	589				
588	588				
585	585				
594	594				
592	592				
602	602				
601	601				
599	599				
607	607				
362	425	400			
522					
527					
529					
531					
517					

070	610		597
075	615		611
080	616		611
085	615	PMT	618
090	621		617
095	625		622
000	628	PMT	627
005	641		626
010	642		628
015	646	PMT	635
020	645	PMT	638
025	642		644
030	651	PMT	650
035	654	PMT	653
040	656		648
045	657		695
050	655		641
055	662	PMT	661
060	665	PMT	664
065	667		660
070	668		641
075	670	PMT	669
080	674	PMT	673
085	674	PMT	677
090	681	PMT	680
095	682		679
100	685	PMT	684
105	686		683
110	689	PMT	688
115	690		687
120	692	PMT	692
125	694		691
130	697	PMT	696
135	698		695
140	700	PMT	699
145	705	PMT	704
150	706		701
155	707		702
160	708		701
165	710	PMT	709
170	713		711
175	714		676

644

658

*** LABEL DICTIONARY -- CHECK ***

LABEL	OPN	TYPE	REFERENCES
10	3		9
15	7	PMT	6

*** LABEL DICTIONARY -- CHECK1 ***

LABEL	OPN	TYPE	REFERENCES
10	3		12
15	7		4
20	10	PMT	5

*** LABEL DICTIONARY -- CHECK2 ***

LABEL	DEFN	TYPE	REFERENCES
13	4		14
15	10		6
20	12	FMT	7
25	15		5

*** LABEL DICTIONARY -- CHECK3 ***

LABEL	DEFN	TYPE	REFERENCES
10	2		7
15	4	FMT	7

*** LABEL DICTIONARY -- CHECK4 ***

LABEL	DEFN	TYPE	REFERENCES
13	2		7
15	4	FMT	3

*** LABEL DICTIONARY -- CLSP ***

LABEL	DEFN	TYPE	REFERENCES					
13	7	FMT	6	53	65	97	115	205
15	8	FMT	11	56	68	100	118	208
20	10	FMT	9					
25	10	FMT	15					
30	20	FMT	14					
35	22	FMT	18					
40	23	FMT	22					
45	24	FMT	17	21				
50	27	FMT	26					
55	35	FMT	34					
60	37	FMT	31					
65	38	FMT	37					
70	35		30	36				
75	40		12	25				
80	43	FMT	42					
85	47	FMT	46					
90	45		45					
95	50	FMT	49					
100	51		44	40				
105	52		41					
110	55	FMT	54					
115	58	FMT	57					
120	63	FMT	62					
125	64		59					
130	67	FMT	66					
135	70	FMT	69					
140	77	FMT	76					
145	74	FMT	74					
150	81	FMT	80					

155	P3	PMT	112	163	
163	P4		111		
165	P5		101		
171	P6	PMT	100		
175	P7		108		
180	P8		105		
185	P9	PMT	104		
190	P10	PMT	103		
195	P11	PMT	102		
200	P12	PMT	101		
205	P13	PMT	100		
210	P14	PMT	108		
215	P15	PMT	107		
220	P16	PMT	112		
225	P17	PMT	111		
230	P18	PMT	116		
235	P19	PMT	119		
240	P20	PMT	124		
245	P21	PMT	127		
			114	163	
250	P22		112		
255	P23		111		
260	P24	PMT	101		
265	P25	PMT	100		
270	P26		108		
275	P27		105		
280	P28		104		
285	P29	PMT	103		
290	P30	PMT	102		
295	P31	PMT	101		
300	P32		100		
305	P33	PMT	122	156	
310	P34	PMT	121		
315	P35		177	191	200
320	P36		176		
325	P37		175		
330	P38		170		
335	P39		167		
340	P40		172	181	
345	P41		169		
350	P42		168		
355	P43		121	168	196
360	P44	PMT	206		
365	P45	PMT	209		
370	P46	PMT	215	233	
375	P47	PMT	214		
380	P48		226		
385	P49	PMT	223		
390	P50		211	219	
395	P51		212		
400	P52		230		
405	P53		236		
410	P54	PMT	31	87	123
415	P55	PMT	193	197	202
					220
					224
					228
					231
					146
					153
					157
					165
					180
					184
					188

*** LABEL DICTIONARY -- DISPLAY ***

LABEL	WPM	TYPE	REFERENCES
13	7	FMT	6
15	10	FMT	9
20	11		8
25	14	FMT	13
30	15		12
35	18	FMT	17
40	19		16
45	22	FMT	21
50	23		20
55	26	FMT	25
60	27		24
65	29	FMT	28
70	34	FMT	33
75	35		32
80	36		31
85	37		30
90	39	FMT	41
95	42		40
100	44	FMT	43
105	45	FMT	48
110	47	FMT	46
115	53	FMT	52
120	57	FMT	56
125	58		55
130	60	FMT	59
135	61		64
140	64	FMT	63
145	72	FMT	71
150	74		70
155	75	FMT	74
160	76		69
165	79	FMT	78
170	83	FMT	82
175	85		81
180	86	FMT	85
185	87		77
190	90	FMT	89
195	93	FMT	92
200	98	FMT	97
205	99		94
210	102	FMT	101
215	105	FMT	104
220	112	FMT	111
225	114	FMT	113
230	116	FMT	115
235	118	FMT	117
240	119		106
245	124	FMT	127
250	129		120
255	132	FMT	131
260	135	FMT	134
265	137	FMT	136
270	139	FMT	138
275	141	FMT	140
			147
			144
			133
			100
			103
			130
			147
			230
			233
			67
			73
			62
			80
			84
			107
			123

*** LABEL DICTIONARY -- MODITY ***

LABEL	DEFN	TYPE	REFERENCES
10	2	PMT	25
21	3	PMT	26
31	4	PMT	30
40	5	PMT	34
50	6	PMT	38
60	7	PMT	42
71	8	PMT	46
81	9	PMT	50
90	10	PMT	54
101	11	PMT	28
111	12	PMT	32
120	13	PMT	16
131	14	PMT	40
141	15	PMT	44
151	16	PMT	48
161	17	PMT	52

*** LABEL DICTIONARY -- CONSYS ***

LABEL	DEFN	TYPE	REFERENCES
10	16	PMT	23
20	17	PMT	26
31	18	PMT	28
40	19	PMT	29
50	20	PMT	30
61	21	PMT	31
71	22	PMT	31
81	23	PMT	31
91	24	PMT	31
101	25	PMT	31
111	26	PMT	31
121	27	PMT	31
131	28	PMT	31
141	29	PMT	31
151	30	PMT	31
161	31	PMT	31
171	32	PMT	31
181	33	PMT	31
191	34	PMT	31
201	35	PMT	31
211	36	PMT	31
221	37	PMT	31
231	38	PMT	31
241	39	PMT	31
251	40	PMT	31
261	41	PMT	31
271	42	PMT	31
281	43	PMT	31
291	44	PMT	31
301	45	PMT	31
311	46	PMT	31
321	47	PMT	31
331	48	PMT	31
341	49	PMT	31
351	50	PMT	31
361	51	PMT	31
371	52	PMT	31
381	53	PMT	31
391	54	PMT	31
401	55	PMT	31
411	56	PMT	31
421	57	PMT	31
431	58	PMT	31
441	59	PMT	31
451	60	PMT	31
461	61	PMT	31
471	62	PMT	31
481	63	PMT	31
491	64	PMT	31
501	65	PMT	31
511	66	PMT	31
521	67	PMT	31
531	68	PMT	31
541	69	PMT	31
551	70	PMT	31
561	71	PMT	31
571	72	PMT	31
581	73	PMT	31
591	74	PMT	31
601	75	PMT	31
611	76	PMT	31
621	77	PMT	31
631	78	PMT	31
641	79	PMT	31
651	80	PMT	31
661	81	PMT	31
671	82	PMT	31
681	83	PMT	31
691	84	PMT	31
701	85	PMT	31
711	86	PMT	31
721	87	PMT	31
731	88	PMT	31
741	89	PMT	31
751	90	PMT	31
761	91	PMT	31
771	92	PMT	31
781	93	PMT	31
791	94	PMT	31
801	95	PMT	31
811	96	PMT	31
821	97	PMT	31
831	98	PMT	31
841	99	PMT	31
851	100	PMT	31

*** LABEL DICTIONARY -- IMPACT ***

LABEL	DEFN	TYPE	REFERENCES
11	14	PMT	15
100	19	PMT	17
			10

*** LABEL DICTIONARY -- NOISE ***

LABEL	DEFN	TYPE	REFERENCES
11	16	PMT	21

20	17	FMT	24
35	18	FMT	27
40	19	FMT	35
50	20	FMT	31
60	21		29
70	22		26
80	23		46
90	24		53
	25		65
	26		
	27		
	28		
	29		
	30		
	31		
	32		
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	44		
	45		
	46		
	47		
	48		
	49		
	50		
	51		
	52		
	53		
	54		
	55		
	56		
	57		
	58		
	59		
	60		
	61		
	62		
	63		
	64		
	65		
	66		
	67		
	68		
	69		
	70		
	71		
	72		
	73		
	74		
	75		
	76		
	77		
	78		
	79		
	80		
	81		
	82		
	83		
	84		
	85		
	86		
	87		
	88		
	89		
	90		
	91		
	92		
	93		
	94		
	95		
	96		
	97		
	98		
	99		
	100		

*** IAPPL DICTIONARY -- READP ***

IAPPL	TYPE	REFERENCES
27	18	16
28	22	20
29	26	24
30	30	28
31	34	32
32	38	36
33	42	40
34	46	44
35	50	48
36	54	52
37	58	56
38	62	60
39	66	64
40	70	68
41	74	72
42	78	76
43	82	80
44	86	84
45	90	88
46	94	92
47	98	96
48	102	100
49	106	104
50	110	108
51	114	112
52	118	116
53	122	120
54	126	124
55	130	128
56	134	132
57	138	136
58	142	140
59	146	144
60	150	148
61	154	152
62	158	156
63	162	160
64	166	164
65	170	168
66	174	172
67	178	176
68	182	180
69	186	184
70	190	188
71	194	192
72	198	196
73	202	200
74	206	204
75	210	208
76	214	212
77	218	216
78	222	220
79	226	224
80	230	228
81	234	232
82	238	236
83	242	240
84	246	244
85	250	248
86	254	252
87	258	256
88	262	260
89	266	264
90	270	268
91	274	272
92	278	276
93	282	280
94	286	284
95	290	288
96	294	292
97	298	296
98	302	300
99	306	304
100	310	308

*** IAPPL DICTIONARY -- WRITE ***

IAPPL	TYPE	REFERENCES
27	18	16
28	22	20
29	26	24
30	30	28
31	34	32
32	38	36
33	42	40
34	46	44
35	50	48
36	54	52
37	58	56
38	62	60
39	66	64
40	70	68
41	74	72
42	78	76
43	82	80
44	86	84
45	90	88
46	94	92
47	98	96
48	102	100
49	106	104
50	110	108
51	114	112
52	118	116
53	122	120
54	126	124
55	130	128
56	134	132
57	138	136
58	142	140
59	146	144
60	150	148
61	154	152
62	158	156
63	162	160
64	166	164
65	170	168
66	174	172
67	178	176
68	182	180
69	186	184
70	190	188
71	194	192
72	198	196
73	202	200
74	206	204
75	210	208
76	214	212
77	218	216
78	222	220
79	226	224
80	230	228
81	234	232
82	238	236
83	242	240
84	246	244
85	250	248
86	254	252
87	258	256
88	262	260
89	266	264
90	270	268
91	274	272
92	278	276
93	282	280
94	286	284
95	290	288
96	294	292
97	298	296
98	302	300
99	306	304
100	310	308

903	27	FMT	25
904	31	FMT	29
905	35	FMT	33
906	41	FMT	38
907	43	FMT	42
			45

*** LABEL DICTIONARY -- COMPUTE ***

LABEL	DEFN	TYPE	REFERENCES	
			CO	
10	57		50	
11	56		54	
12	97		63	
13	56		84	91
14	95		92	93
17	90		88	
21	235		98	
22	234		99	
23	112		109	
24	118		115	
25	185		139	
26	177		148	
27	224		144	
28	217		200	
29	248		236	
33	12		10	
34	17		14	
35	23		19	
36	27		25	
37	48		44	
38	11		29	
39	16		33	
40	17		34	
41	62		58	
42	67		63	
43	72		68	
44	77		73	
45	53		51	
46	61		59	
47	66		64	
48	71		69	
49	76		74	
50	82		78	
51	266		255	263
52	254		249	250
53	275		267	269
54	245		242	276
55	290		280	287
56	316		291	
57	202		300	
58	312		298	305
59	315		313	309
60	321		317	
61	325		318	
62	326		322	323
63	81		79	
64	530		538	
65	336		335	
66	305		337	
67	143		380	
68	195		369	377
69	175		371	
70	174		372	
71	142		315	

FMT

961	400	PM	407		
962	406	PM	405		
973	346	PM	147	392	417
990	6	PM	5		511
1000	364	PM	363		
1001	400	PM	409		
1002	525	PM	526		

*** TABUL DICTIONARY -- TRANS ***

JAPN	TYFM	TYFF	REFERENCES,
100	14		20
200	9		7