

AD-A131 218

AMMUNITION RESUPPLY MODEL VOLUME 2 PROGRAMMER'S MANUAL

1/3

(U) COMBINED ARMS OPERATIONS RESEARCH ACTIVITY FORT

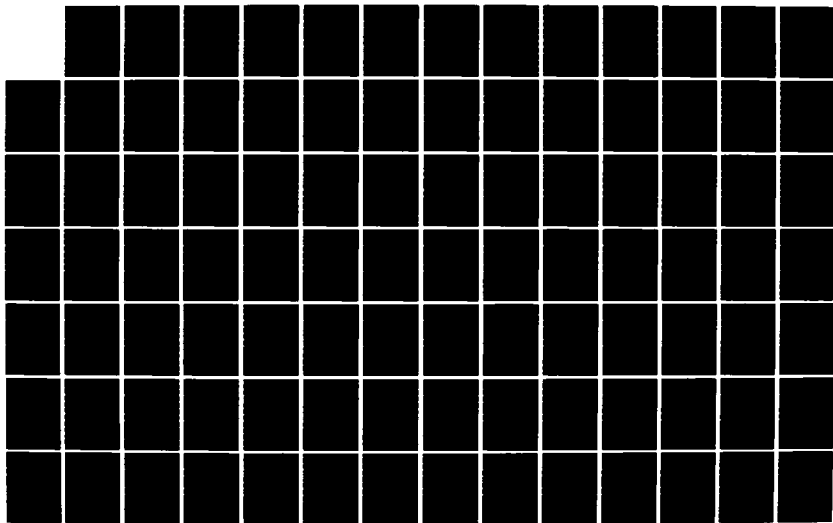
LEAVENWORTH KS A M RESNICK ET AL. MAY 83

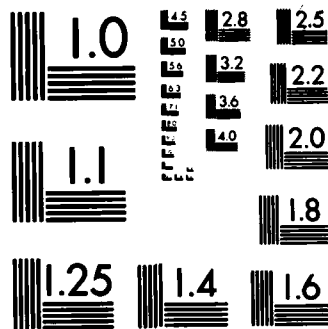
UNCLASSIFIED

CAORA/TR-6/83-VOL-2

F/G 19/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MAY 1983

CAORA/TR-6/83

ACN 65676

**AMMUNITION RESUPPLY MODEL
PROGRAMMER'S MANUAL
VOLUME II**

TECHNICAL REPORT 6/83

**UNITED STATES ARMY
COMBINED ARMS CENTER**

**DIRECTORATE OF STUDIES AND ANALYSIS
COMBINED ARMS OPERATIONS RESEARCH ACTIVITY
FORT LEAVENWORTH, KS 66027**

83-4046

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

83 08 10 003

DTIC FILE COPY

ADA131218



DEPARTMENT OF THE ARMY

COMBINED ARMS OPERATIONS RESEARCH ACTIVITY
FORT LEAVENWORTH, KANSAS 66027

REPLY TO
ATTENTION OF

ATOR-CAS-F

30 June 1983

SUBJECT: Documentation of the Ammunition Resupply Model

TO: SEE DISTRIBUTION

Included find your copy or copies of the latest (May 1983) three volume documentation of the CAORA developed Ammunition Resupply Model. For those not already familiar with the system, this supercedes all previous ARM documentation. For others it extends earlier documentation to include several significant program modifications and the use of a different computer system.

1 Encl

RONALD G. MAGEE
Dir, Studies and Analysis Directorate

DISTRIBUTION:

Commander
USATRADOC
ATTN: ATCD-A, ATCD-S
Ft. Monroe, VA 23651

Commander
Defense Technical Information Center
Cameron Station
Alexandria, VA 22314

Director
USATRASANA
ATOR-TG
White Sands Missile Range, NM 88002

Commander
US Army Logistics Center
ATTN: ATCL-C, ATCL-CF, ATCL-LE, ATCL-OS
Ft. Lee, VA 23801

ATOR-CAS-F
SUBJECT: Documentation of the Ammunition Resupply Model

30 June 1983

DISTRIBUTION (Continued)

Commander
US Army Air Defense Center & Ft. Bliss
ATTN: ATZC-CD
Ft. Bliss, TX 79916

Commander
US Army Aviation Center & Ft. Rucker
ATTN: ATZQ-CD
Ft. Rucker, AL 36362

Commander
US Army Armor Center & Ft. Knox
ATTN: ATSB-CD-S
Ft. Knox, KY 40121

Commander
US Army Engineer Center & Ft. Belvoir
ATTN: ATSEN-CTDK-CS
Ft. Belvoir, VA 22060

Commander
US Army Armament Material Readiness Cmd
ATTN: DRSAR-SAS
Rock Island, IL 61299

Commander
US Army Tank Automotive Command
ATTN: DRSTA-VS
Warren, MI 48090

Commander
US Army Field Artillery School
ATTN: ATSF-CD
Ft. Sill, OK 73503

Commander
US Army Infantry School
ATTN: ATSH-CD-CS
Ft. Benning, GA 31905

Commander
US Army Intelligence Center & School
ATTN: ATSI-CTD-CS
Ft. Huachuca, AZ 86611

ATOR-CAS-F

30 June 1983

SUBJECT: Documentation of the Ammunition Resupply Model

Commander
US Army Missile & Munitions Center & School
ATTN: ATSK-C
Redstone Arsenal, AL 35809

Commander
USA Ordnance Center and School
ATTN: ATSL-CTD-CS
APG, MD 21005

Commander
USA Institute of Military Assistance
DCOMDT Cmt Tng Div
Fort Bragg, NC 28307

Commander
USA Transportation School
ATTN: ATSP-CTD-CS
Fort Eustis, VA 23604

Commander
USA Concepts Analysis Agency
8120 Woodmont Avenue
Bethesda, MD 20014

Commander
USA Combined Arms Combat Developments Activity
ATZL-CAM-I
Fort Leavenworth, KS 66027

Commandant
US Naval Post-Graduate School
ATTN: Dr. Sam Parry
Monterey, CA

Director
USA Human Engineering Laboratory
ATTN: DRXHE-CSS
APG, MD 21005

Errata Sheet

P. 69 of Volume I ARM Methodology and User's Manual should be replaced with the following:

<u>USER RESPONSES AND SYSTEM MESSAGES</u>	<u>REMARKS</u>
ENTER NUMBER OF ACTIVE ATPS <u>6</u>	
ENTER NUMBER OF ACTIVE ASPs <u>4</u>	
TIME = 2160.05 ?	Current simulation time menu option
<u>4</u>	The RETURN option
TIME = 2160.1	This is the end of simulation time
SSG ARMPL.FREE, THEN SSG ARMPL. EDITYES	To remind the user. This now begins a process of editing data for following CI.
<u>@ SSG ARMPL.FREE</u>	Putting ARMPL.FREE into machine readable form.
SSG 20R1 S74T27	Machine reply
<u>SGS</u>	To input information to SSG.
<u>FREE CI(N)</u> <u>@</u>	Tells machine which files to free.
END SSG ERRORS 0/0/0 RETAIN LEVEL 1.2	End of input files are now free and symmed to the printer
<u>@ SSG ARMPL.EDITYES</u>	Putting ARMPL.EDITYES into machine.
SSG 20R1574T27	Above entry processed to input CI nos to SSG.
<u>EDIT CI(N), CI (N+1)</u>	CI(N) to be edited and renamed CI(N+1)
<u>@ XOT ARMPL.EDIT</u> ZERO COUNTERS (YES or NO) <u>YES</u> ???	Execute the EDIT program EDIT menu option. This is now the same as in previous chapter Building and Editing data files.

Technical Report TR 6-83
May 1983

Directorate of Studies and Analysis
US Army Combined Arms Operations Research Activity
Fort Leavenworth, KS 66027

AMMUNITION RESUPPLY MODEL

VOLUME II

PROGRAMMERS MANUAL

by

MAJ Allan Resnick
and
Mr. Rick Cunningham

ACN 65676

Approved by:

Ronald G. Magee
RONALD G. MAGEE
Director, S&AD

John L. Ballantyne
JOHN L. BALLANTYNE
BG, USA
Commander, CAORA

83-4046

A



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report TR 6-83	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ammunition Resupply Model Volume II Programmers Manual		5. TYPE OF REPORT & PERIOD COVERED Final
7. AUTHOR(s) MAJ Allan M. Resnick Mr. Rick D. Cunningham		6. PERFORMING ORG. REPORT NUMBER TR 6-83
9. PERFORMING ORGANIZATION NAME AND ADDRESS		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
12. REPORT DATE May 1983		13. NUMBER OF PAGES 193
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release Distribution Unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Ammunition Resupply War Game Ammunition Consumption Rates		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report is the second of three volumes that describe the Ammunition Resupply Model (ARM). The model was designed to simulate those activities associated with ammunition resupply. This manual contains the FORTRAN code of the ARM simulation and a description of the major event subroutines. Volume I contains the user instructions and methodology of the ARM model. The third volume contains a standard European Heavy division data base.		

ABSTRACT

This manual is the second of three volumes produced to document the Ammunition Resupply Model. The Ammunition Resupply Model (ARM) was designed to simulate those activities associated with ammunition resupply in parallel with the play of existing war games. Its purpose is to assess the capability of a given TOE structure to respond to the logistical demands placed upon it by various numbers of ammunition-expending weaponry and/or to assess the capability of existing or proposed resupply systems (i.e., number, location, or sizes of Ammunition Transfer Points (ATP) and Ammunition Supply Points (ASP)). This report contains the FORTRAN code and a discussion of the major subroutines of the Ammunition Resupply Model. Volume I contains a discussion of model methodology, database development, and operators guide. The third volume contains a description of a standard heavy European Division data base along with data sources.

TABLE OF CONTENTS

	<u>Page</u>
CHAPTER 1 - GENERAL	
Introduction.	1-1
General Information	1-1
An ARM Overview	1-1
File Usage.	1-12
Common Blocks	1-12
The Events Queue.	1-13
Vehicle Queue	1-14
CHAPTER 2 - ARM SUBROUTINE DESCRIPTIONS	
MAINARM	2-1
Event Descriptions.	2-1
DEMAND.	2-1
RELOAD.	2-2
UNTARV.	2-2
UNIDEP.	2-2
ATPARV.	2-2
ATP	2-3
ATPAR1.	2-3
CSAARV.	2-3
CSADEP.	2-3
ATPAR2.	2-3
ASPAR1.	2-4
ASPARV.	2-4
ASP	2-4
ASPAR2.	2-4
HELARV.	2-5
HASPAR.	2-5
Special Purpose Routines.	2-5
OPERA	2-5
INTRDK.	2-5
DUALMX.	2-6
LOPWDR.	2-6
ASPCK	2-6
DEPASP.	2-6
SERVER.	2-6
CHAPTER 3 - FORTRAN LISTINGS	
ARM Program	3-1
MAINARM	3-1
ASP	3-4
CSPARV.	3-9
ASPCK	3-12
ASPAR1.	3-14
ASPAR2.	3-19
ATP	3-23
ATPARV.	3-29
ATPAR1.	3-34
ATPAR2.	3-38

TABLE OF CONTENTS (Cont'd)

	<u>Page</u>
CONTRL.	3-40
CREEVT.	3-42
CSAARV.	3-43
CSADep.	3-46
DEMAND.	3-49
DEPASP.	3-53
DUALMX.	3-55
EDITD.	3-58
ENDSIM.	3-62
EVSTOP.	3-63
FINTR.	3-64
GEVEVT.	3-66
GETQUE.	3-67
HASPAR.	3-68
HELARV.	3-69
INIT.	3-71
INTRDK.	3-72
IQ.	3-74
LDPWDR.	3-76
LOOKEV.	3-78
NEXTEV.	3-79
NXTQUE.	3-80
OPERA.	3-81
PUTEVT.	3-83
PUTQUE.	3-86
QINIT.	3-87
RDIEOX.	3-88
RDJIFF.	3-90
READF.	3-95
RELOAD.	3-98
REPORT.	3-102
SCHED.	3-109
SERVER.	3-110
SEVENT.	3-116
SETQUE.	3-118
TRKPUT.	3-119
TRKTIM.	3-121
TRUCK.	3-122
UNTARV.	3-123
UNTDEP.	3-125
GETONE.	3-128
EDIT Program.	3-129
PRINT.	3-131
CONVERT Program.	3-132
ADDEVTV.	3-133
Information.	3-135
DEMAND Generation Program.	3-145
UN /AC Exec. ve Language Runstreams.	3-169

CHAPTER 1

1. INTRODUCTION.

a. This manual is intended for the programmer who has the task of maintaining, transferring or modifying the Ammunition Resupply Model (ARM). This manual contains listings and discussion of the ARM simulation, as well as routines for editing data and events, and a demand generation program.

(1) Chapter one contains general information about the ARM model; including files, common blocks, and subroutine calling sequences.

(2) Chapter two gives a detailed description of the major event subroutines in ARM.

(3) Chapter three lists the FORTRAN code for the ARM model and the UNIVAC executive language runstreams that drive the program.

2. GENERAL INFORMATION.

a. ARM Summary. ARM is a set of FORTRAN subroutines designed to assist an analyst in studying the ammunition flow from the Corps Storage Area (CSA) to the weapons. ARM also models the operations of the Ammunition Transfer Points (ATPs) and Ammunition Supply Points (ASPs). Additional information can be found in volume one-Methodology.

b. Program Specifications.

(1) Language and operating system. ARM is written in standard FORTRAN 77, and runs on the TRADOC Data Processing Field Office (DPFO) UNIVAC 1100/80. The driving runstreams in chapter 3 are written for the UNIVAC Symbolic Stream Generator (SSG processor). Their function is to assign the input and output files for ARM.

(2) Program size. There are 49 subroutines, consisting of about 5000 lines of code, in ARM. This produces about 20000 words of instructions on the UNIVAC, and requires about 50,000 words in the data bank.

(3) Operating Environment. ARM requires an interactive (demand) terminal with a printer and/or a CRT. Output can be routed to a high speed line printer.

3. AN ARM OVERVIEW.

a. Major groups of subroutines perform the following functions:

- (1) Event Processing.
- (2) Vehicle Queue Processing.
- (3) Event Storage and Retrieval.
- (4) Input and Output.

b. Event Processing. These subroutines simulate the flow of ammunition from the Corps Storage Area (CSA) down to the consuming weapon systems. The events are ASP, ASPARV, ASPAR1, ASPAR2, ATP, ATPARV, ATPAR1, ATPAR2, CONTRL, CSAARV, CSADEP, DEMAND, ENDSIM, HASPAR, HELARV, RELOAD, REPORT, UNTARV, and UNTDEP. Additional subroutines that aid in event processing include: ASPCK, DEPASP, DUALMX, INTRDK, LDPWDR, OPERA, RDIEXO, RDJIFF, and SERVER. A brief explanation of each is in chapter 2.

c. Vehicle Queue Processing. These subroutines, in general, keep track of all vehicles used in ARM by performing various functions on them in the queue appropriate to the event being simulated. FINTK, GETONE, GETQUE, NXTQUE, PUTQUE, and SETQUE all perform vehicle queue operations (see chapter 3 for definition of each). A summary of vehicle queue functions is presented in paragraph 7.

d. Event storage and retrieval subroutines are made up of CREEVT, GETEVT, NEXTEVT, PUTEVT, SCHED, and SEVENT. This group of subroutines allow the list of events in ARM (Events Queue) to be kept current and executable. Paragraph 6 explains how events are scheduled.

e. Input and Output. This grouping permits various input/output operations, including editing of data and printing various output.

(1) Initialization, input and output for ARM is performed by EVSTOP, INIT, QINIT, READF, TRKPUT, TRKTIM, TRUCK, and PRINT.

(2) The ARM program may be supplemented by the EDIT program which provides the capability of editing the database between Critical Incidents (CIs). Editing between CIs provides the ability to replicate results of the following CI. The following subroutines make up program EDIT:

EDIT (main)	PRINT
EDITD	PUTQUE
GETONE	READF
GETQUE	SETQUE
NXTQUE	TRKPUT

Note that most of the EDIT program subroutines are also found in the ARM simulation. Additional programs used to edit event parameters, are found in Volume I.

f. Subroutine Calling Sequences.

(1) All of the routines in ARM are directly called by other subroutines or the main program, except for the function IQ (which returns the proper vehicle queue number from parameters). Initially listed are those subroutines which are called by the MAINARM program; subsequently, all subroutines are cross-referenced by calling sequences. Additionally, common blocks that are referenced by calling subroutine are provided.

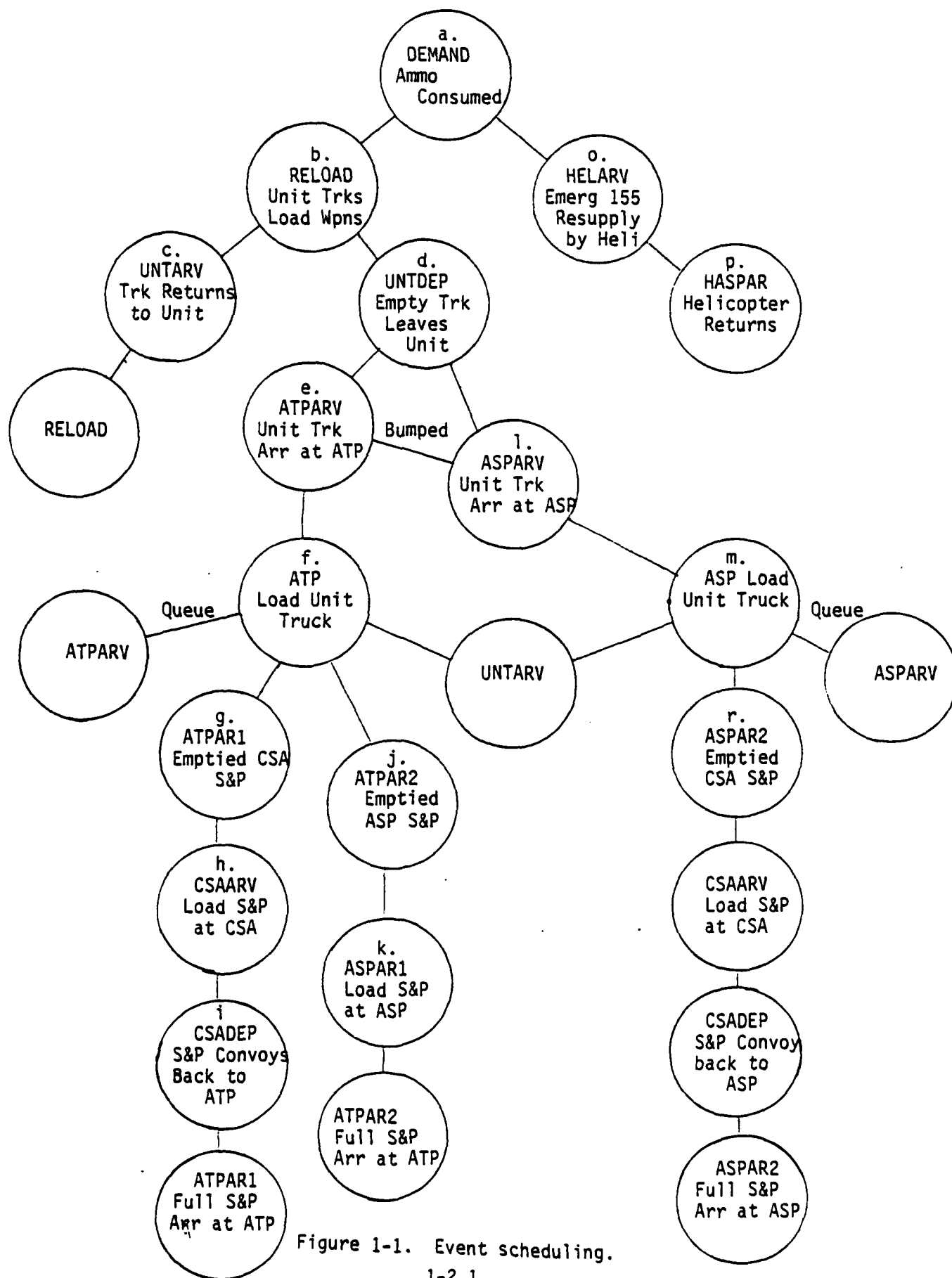


Figure 1-1. Event scheduling.
1-2.1

(2) Events are subroutines that are allowed to be scheduled at some future time, that is, the calling arguments are stored (in a queue) along with the time in the simulation that the event is to happen. When the event time comes to pass, the calling arguments (or parameters) are retrieved from the queue by GET EVT and the event subroutine is called directly by MAINARM. In this way, event execution may be delayed by a given amount of time from when the event was scheduled. The event type (e.g., type 1 is DEMAND) is saved in the passing argument list (IPARM(5)) so that MAINARM can determine which event subroutine to call. The event type is specified at the time the event is scheduled.

(3) Table 1-1 lists all of the ARM subroutines and what they call. Table 1-2 lists the subroutines in the EDIT program and which other subroutines are referenced (called) by each, and which routines reference each. In contrast to Table 1-1, Table 1-3 contains all subroutines and where they were called from (the event type appears at the left column). Tables 1-4 and 1-5 deal with scheduled (as opposed to called directly) events: what routine schedules what; and what routine is scheduled by what, respectively (the numbers in parenthesis are event numbers).

Table 1-1.

EVENT

<u>MAINARM Calls</u>	<u>Common</u>	<u>Calls</u>
INIT	AUNIT EVENTS LOG QUENUM QUEPUT	QINIT, SEVENT, SCHED, TRKT IM, CONTRL, RDJIFF
NEXTEV		
LOOKEV	LOG	
1 DEMAND	LOG	RDIEXO, SCHED, OPERA
2 RELOAD	LOG	IQ, DUALMX, FIN TK, INTRDK, SCHED, OPERA
3 UNTDEP	LOG	OPERA, INTRDK, SCHED
4 ATPARV	LOG	SCHED, IQ, FIN TK, PUTQUE, GETQUE, OPERA, INTRDK
5 ASPARV	LOG	SCHED, IQ, ASP CK, GETQUE, PUTQUE
6 ATP	LOG	IQ, FIN TK, SCHED, OPERA, LDPWPR, INTRDK
7 ASP	LOG	IQ, FIN TK, SCHED, OPERA, INTRDK
8 UNTARV	LOG	DEPASP, SERVER, IQ, PUTQUE, SCHED
9 CSAARV	LOG	PUTQUE, SCHED, GETQUE
10 ATPAR1	LOG	OPERA, SCHED, IQ, PUTQUE, GETQUE, INTRDK
11 ATPAR2	LOG	IQ, PUTQUE, OPERA, INTRDK, SCHED
12 ASPAR1	LOG	IQ, PUTQUE, OPERA, SCHED, FIN TK, GETQUE, INTRDK
13 ASPAR2	LOG	PUTQUE, GETQUE, OPERA, INTRDK, SCHED, IQ, SERVER

Table 1-1. (Cont'd)

<u>EVENT</u>		
<u>MAINARM Calls</u>	<u>Common</u>	<u>Calls</u>
14 HELARV	LOG	OPERA, SCHED
15 HASPAR	LOG	
16 CSADEP	LOG	OPERA, INTRDK, SCHED, PUTQUE, GETQUE
17 REPORT	AUNIT LOG	TRUCK
18 CONTRL		EDITD, REPORT, SCHED, CREEVT
19 ENDSIM	AUNIT LOG	
	QUENUM	
	QUEPUT	
EVSTOP	EVENTS	
<u>Subroutine</u>	<u>Common</u>	<u>Calls</u>
ASPCK	LOG	SCHED, OPERA
CREEVT		READF, SCHED
DEPASP	LOG	GETQUE, SCHED, PUTQUE
DUALMX	LOG	IQ, FINTK, INTRDK, SCHED
EDITD	AUNIT LOG	READF
FINTK	LOG	GETQUE, PUTQUE
GETEVT	EVENTS	
GETONE	QUENUM	
	QUEPNT	
GETQUE	QUENUM	
	QUEPNT	
INTRDK	LOG	
IQ		
LDPWDR	LOG	IQ, FINTK, PUTQUE
OPERA	LOG	
PUTEVT	EVENTS	
PUTQUE	QUENUM	
	QUEPNT, LOG	
QINIT	EVENTS	
RDIEXO	LOG	SCHED
RDJIFF	AUNIT LOG	SCHED
READF		
SCHED		LOOKEV, PUTEVT, CONTRL
SERVER	LOG	PUTQUE, SCHED, GETQUE, IQ, FINTK, OPERA
SEVENT		SCHED
TRKPUT		READF, NXTQUE, GETQUE, PUTQUE, SETQUE, GETONE
TRKTIM	LOG	
TRUCK	AUNIT LOG	
MAINARM	AUNIT LOG	
	QUENUM	
	QUEPNT	

TABLE 1-2

<u>Routine</u>	<u>Called by</u>	
EDITD	EDIT	
GETONE	TRKPUT	
GETQUE	EDIT, PRINT, TRKPUT	
NXTQUE	PRINT, TRKPUT	
PUTQUE	EDIT, PRINT, TRKPUT	
READF	EDITD, TRKPUT	
SETQUE	TRKPUT	
TRKPUT	EDIT	
<u>Routine</u>	<u>Common</u>	<u>Calls</u>
EDIT	AUNIT LOG QUENUM QUEPNT	EDITD, GETQUE, PRINT, PUTQUE, TRKPUT
EDITD	AUNIT LOG QUENUM QUEPNT	READF
GETQUE	QUENUM QUEPNT	
NXTQUE	QUENUM QUEPNT	
PRINT	QUENUM QUEPNT	GETQUE, NXTQUE, PUTQUE
PUTQUE	LOG QUENUM QUEPNT	
READF		
SETQUE	QUENUM QUEPNT	
TRKPUT		GETQUE, NXTQUE, PUTQUE, READF, SETQUE

Table 1-3.

<u>Event Number</u>	<u>Subroutine</u>	<u>Called by</u>
7	ASP	MAINARM
12	ASPAR1	MAINARM
13	ASPARA2	MAINARM
5	ASPARV	MAINARM
	ASPCK	ASPARV
6	ATP	MAINARM
10	ATPAR1	MAINARM
11	ATPAR2	MAINARM
4	ATPARV	MAINARM
18	CONTRL	INIT, MAINARM, SCHED
	CREEVT	CONTRL
9	CSAARV	MAINARM
16	CSADEP	MAINARM
1	DEMAND	MAINARM
	DEPASP	UNTARV
	DUALMX	RELOAD
	EDITD	CONTRL
19	ENDSIM	MAINARM
	EVSTOP	MAINARM
	FINTK	ASP, ASPAR1, ATP ATPARV, DUALMX, LDPWDR, RELOAD, SERVER
	GETEVT	NEXTEV
	GETONE	TRKPUT

Table 1-3. (Cont'd)

<u>Event Number</u>	<u>Subroutine</u>	<u>Called by</u>
	GETQUE	ASPAR1 ASPAR2, ASPARV, ATPAR1, ATPARV, CSAARV, CSADEP, DEPASP, FINTK, SERVER
15	HASPAR	MAINARM
14	HELARV	MAINARM
	INIT	MAINARM
	INTRDK	ASP, ASPAR1, ASPAR2, ATP, ATPAR1, ATPAR2, ATPARV, CSADEP, DUALMX, RELOAD, UNTDEP
	IQ	ASP, ASPAR1, ASPAR2, ASPARV, ATP, ATPAR1, ATPAR2, ATPARV, DUALMX, LDPWDR, RELOAD, SERVER UNTARV
	LDPWDR	ATP
	LOOKEV	MAINARM
	NEXTEV	MAINARM
	OPERA	ASP, ASPAR1, ASPAR2, ASPCK, ATP, ATPAR1, ATPAR2, ATPARV, CSADEP, DEMAND, HELARV, RELOAD, SERVER, UNTDEP
	PUT EVT	SCHED
	PUTQUE	ASPAR1, ASPAR3, ASPARV, ATPAR1, ATPAR2, ATPARV, CSAARV, CSADEP, DEPASP, FINTK, LDPWDR, SERVER, UNTARV
	QINIT	INIT
	RDIE XO	DEMAND
	RDJIFF	INIT

Table 1-3. (Cont'd)

<u>Event Number</u>	<u>Subroutine</u>	<u>Called by</u>
	READF	CREEVT, EDITD, TRKPUT
2	RELOAD	MAINARM
17	REPORT	CONTRL, MAINARM
	SCHED	ASP, ASPAR1, ASPAR2, ASPARV, ASPCK, ATP, ATPAR1, ATPAR2, ATPARV, CONTRL, CREEVT, CSAARV, CSADep, DEMAND, DEPASP, DUALMX, HELARV, INIT, RDIE XO, RDJIFF, RELOAD, SERVER, SEVENT, UNTARV, UNTDEP
	SERVER	ASPAR2, UNTARV
	SEVENT	INIT
	TRKPUT	CONTRL
	TRKTIM	INIT
	TRUCK	REPORT
8	UNTARV	MAINARM
3	UNTDEP	MAINARM

TABLE 1-4

<u>NUMBER</u>	<u>EVENT</u>	<u>NAME</u>	<u>SCHEDULED BY</u>
1		DEMAND	RDIEX0,RDJIFF
2		RELOAD	DEMAND,UNPARV
3		UNTDEP	DUALMX,RELOAD
4		ATPARV	ATP,ATPARV,UNTDEP
5		ASPARV	ASP,ASPARV,ASPCK, ATP,ATPARV,DUALMX, RELOAD,UNTDEP
6		ATP	ATPARV,DEPASP,SERVER
7		ASP	ASPARV,DEPASP,SERVER
8		UNPARV	ASP,ASPAR1,ATP, DUALMX,RELOAD,SERVER
9		CSAARV	ASPAR2,ASPAR1
10		ATPAR1	ATP,ATPAR1,CSADEP
11		ATPAR2	ASPAR1,ASPAR2,ATP
12		ASPAR1	ASPAR1,ASPAR2,SERVER
13		ASPAR2	ASP,ASPAR1,ASPAR2, ATPAR1,CSADEP,SERVER
14		HELARV	DEMAND
15		HASPAR	DEMAND,HELARV
16		CSADEP	CSAARV
17		REPORT	
18		CONTRL	CONTRL
19		ENDSIM	CONTRL, INIT

Table 1-5

<u>EVT#</u>	<u>Routine</u>	<u>Schedules</u>
(7)	ASP	(5) ASPARV, (8) UNTARV, (13) ASPAR2
(12)	ASPAR1	(12) ASPAR1, (13) ASPAR2, (8) UNTARV, (11) ATPAR2
(13)	ASPAR2	(9) CSAARV, (13) ASPAR2, (11) ATPAR2
(5)	ASPARV	(5) ASPARV, (7) ASP
	ASPCX	(5) ASPARV
(6)	ATP	(4) ATPARV, (5) ASPARV, (8) UNTARV, (10) ATPAR1, (11) ATPAR2
(10)	ATPAR1	(13) ASPAR2, (10) ATPAR2, (9) CSAARV
(11)	ATPAR2	(12) ASPAR1
(4)	ATPARV	(5) ASPARV, (6) ATP, (4) ATPARV
	CREVT	SCHEDULES ANY EVENT
(18)	CONTRL	(18) CONTRL, (19) ENDSIM
(9)	CSAARV	(16) CSADEP
(16)	CSADEP	(10) ATPAR1, (13) ASPAR2
(1)	DEMAND	(2) RELOAD, (15) HASPAR, (14) HELARV
	DEPASP	(7) ASP, (6) ATP
	DUALMX	(5) ASPARV, (3) UNTDEP, (8) UNTARV
(14)	HELARV	(15) HASPAR
	INIT	(19) ENDSIM
	RDIEXO	(1) DEMAND
	RDJIFF	(1) DEMAND
(2)	RELOAD	(5) UNTDEP, (8) UNTARV

<u>EVT#</u>	<u>Routine</u>	<u>Schedules</u>
	SERVER	(8) UNTARV, (6) ATP, (7) ASP, (12) ASPAR1, (13) ASPAR2
	SEVENT	SCHEDULES ANY EVENT
(8)	UNTARV	(2) RELOAD
(3)	UNTDEP	(5) ASPARV, (4) ATPARV

4. File Usage

a. The ARM program use the following files (see Volume I for details):

- 2 Audit trail (output), a listing of all events that were scheduled or executed.
- 3 Database input at start of CI.
- 4 Database output at the end of the CI.
- 7 Events file, input, to be executed during this CI (generated by previous CI).
- 8 Events file, output, to be executed during the next CI.
- 9 Demand (input), the expenditure rates for the unit weapons.
- 11 Additional events (input) to be executed this CI. This file is manually built in program ADDEVT (see Volume I).
- 14 Unit status report output.

b. Program EDIT uses these files:

- 2 Printout of truck queues.
- 3 Database input to be edited.
- 4 Database output after editing.
- 13 Distance file, input, to be stored in the IUNIT array distance attributes. The distances must change to reflect unit movement (see Volume I).

5. COMMON BLOCKS.

a. EVENTS. EVENTS is the events queue. It is accessed from: INIT, EVSTOP, GET EVT, PUT EVT, QINIT. An explanation of how it is used can be found in paragraph 6. The common declaration is referenced with the "include events" statement:

```
COMMON/EVENTS/JSTAT(6), JEVDS(2048, 4), IEVS(5, 2048)
```

b. QUENUM. Contains IHEAD, the list of last item in queue for vehicle queues (see paragraph 6):

```
COMMON/QUENUM/IHEAD(176)
```

c. QUEPNT. The ITEMS in the vehicle queue are held in common QUEPNT. QUEPNT is used in association with QUENUM (see paragraph 6):

```
COMMON/QUEPNT/ITEMS(1400)
```

d. AUNIT. Contains the alphabetic unit names:

```
COMMON/AUNIT/AUNIT(75,2)
```

e. LOG. Contains the database for the ARM model (see volume I for the array definitions):

```
COMMON/LOG/IATP(10,53), IASP(10,110), IUNIT(75,142),  
Z ITRUCK(1400,15), ITYPE(9,6), IMIX(91,32), INTER(10),  
Z IRSTME(23,3), IATPSD(5), IDAY, TIME,  
Z IATPAM(10,40), ICSA(3,32), LPPAR(10), IASPAM(10,120),  
Z LUOUT, TCIST, TCILNG, LOOK(19), JUNIT(8,24),  
Z JATP(10,6), JASP(10,9), IATPSP(10,22),  
Z IASPSP(10,30), IAMLVL(2,30), ISERV(10)
```

6. THE EVENTS QUEUE.

a. The flexibility of ARM comes from the ability of its subroutines to schedule events to happen at some future time. On the most fundamental level, this is done by sorting the events in chronological order and storing them in the events queue (list) for future processing.

b. Events are scheduled by calling subroutine SCHED and passing the event type (see table 1-1), calling arguments and the time the event is to take place. SCHED calls PUTEVT to put the event in the event queue.

c. The parameters (arguments) associated with the event at the time it was scheduled are the thing that makes that event unique. For example, if a truck is scheduled for an event at some future time, the first of five parameters is generally the truck number. The parameter list for an event is usually given the name IPARM.

d. If an event is scheduled at the current time, (i.e., no delay is assessed at time of scheduling) then that event will be the first event in queue (next to be executed), even if there were events scheduled for that time previously.

e. The structure of the events queue may be found in figure 1-2. In general, JEVDS is a doubly linked list of events, sorted by time. The double linkage allows insertion from either the front or rear. IEVS contains the parameters (IPARMS) associated with the event when it was scheduled JSTAT(1) contains the subscript to JEVDS and IEVS of the next event in queue (front). Subsequent events are found by following the subscript to the next event (found in JEVDS (IFIRST,1), and so on).

JEVDS (2048,4)				JSTAT (6)	
JFORE	JBACK	JTIME(1)	JTIME(2)		
				JFIRST	1
				JLAST	2
				JEMPTY	3
				NUMEVT	4
				NEMPTY	5
				MAXEVT	6

Event		IEVS (5,2048)		
IPARM				
1				
2				

Forward Pointer		Backward Pointer	

NOTE: First subscript of JEVDS is the second subscript of IEVS.

Event IPARM	IEVS (5,2048)
1	
2	
3	
4	
5	

Figure 1-2. Events Queue.

7. VEHICLE QUEUES.

a. Because several vehicles may wait in the same place at a given time, a vehicle queueing system has been set up for ARM. To put a vehicle (NTRUCK) in queue (NQUEUE), a subroutine must "CALL PUTQUE (NTRUCK, NQUEUE)." Similarly to pull a vehicle from queue, a "CALL GETQUE (NTRUCK, NQUEUE)" is executed.

b. There are two arrays that make up the vehicle queues (see figure 1-3), IHEAD and ITEMS. IHEAD is found in common block QUENUM; ITEMS is in QUEPNT. IHEAD (NQUEUE) contains the number of the last vehicle in queue "NQUEUE." The vehicle in front of NTRUCK in queue is given by ITEMS(NTRUCK). In order to get to the first item in queue, it is necessary to traverse through the pointers until you reach the element of array ITEMS that contains a zero.

c. Since IHEAD is dimensioned to 176, there are 176 possible queues. ITEMS is dimensioned to 1400, so there are 1400 vehicles that can be in any combination of the 176 queues.

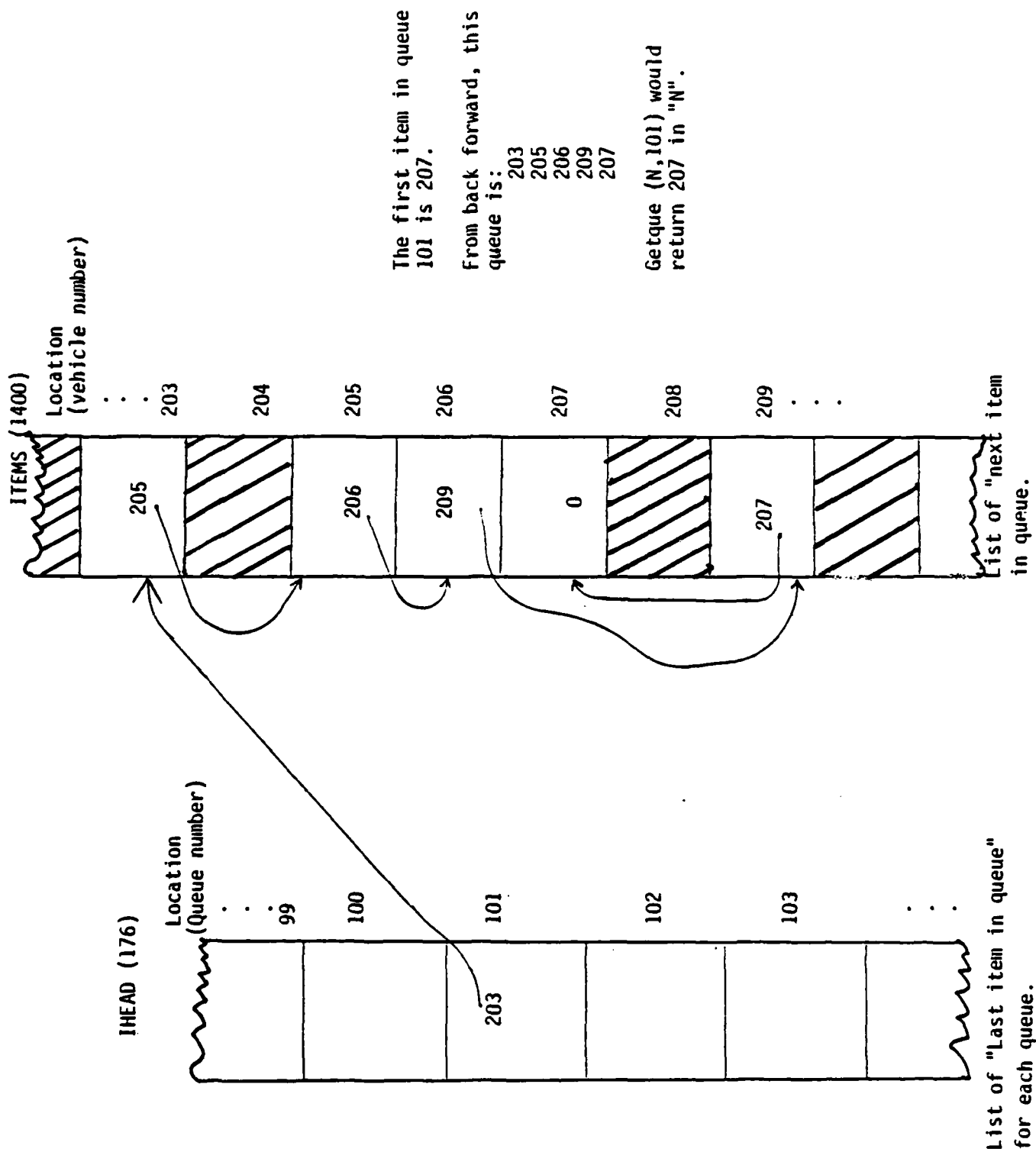


Figure 1-3. Truck Queues.

CHAPTER 2

1. MAINARM.

a. The general logic of MAINARM is as follows:

- (1) Initialize, read database.
- (2) The following is repeated until simulation end:
 Get an event.
 Process the event.
- (3) Write out the database for the next CI.
- (4) Stop.

b. Initialization Routines.

(1) INIT. INIT reads the database from file 3. The database is held in common for the rest of the Critical Incident (CI). This allows all of the events to modify the data. The list of the arrays in the database and their definitions can be found in Volume I. INIT gives the operator access to the ARM menu by calling CONTRL.

(2) INIT calls RDJIFF which reads the demand file (file 9) and updates the demand attributes in the IUNIT array for each ammo type. RDJIFF schedules the first DEMAND pulse (for each unit) for the CI.

c. Event Processing: MAINARM processes events as they appear in the 'events queue' (common events, see paragraph 6, Volume II). NEXTEV retrieves the next scheduled event by calling GET EVT. MAINARM calls LOOKEV to print the event message on the audit trail (file 2).

Event subroutines are called from MAINARM dependent on the event type (IPARM(5), see Table 1-1 for explanation of the codes), until the event ENDSIM is processed.

2. EVENT DESCRIPTIONS.

a. DEMAND. DEMAND simulates the ammunition consumption at the unit level. DEMAND calls RDIEXO to fire that portion of the CI's ammunition demand allocated to this pulse (an instantaneous expenditure of ammunition as represented by the scenario and demand generation process). For a maneuver unit, all of the demand is fired (at the end of the CI), because they are only pulsed once per CI. For multipulse units, RDIEXO schedules the next DEMAND event for this CI. For artillery units, the DEMAND event is scheduled every 60 minutes. For 155mm units, if current supply plus ammunition on trucks is less than the critical resupply level, an emergency helicopter resupply event

(HELARV) is scheduled. RDIEXO expends the ammunition by subtracting part of the demand from the current supply. If current supply per weapon falls below the routine resupply level for at least one ammunition type, a RELOAD event is scheduled immediately.

b. RELOAD.

(1) RELOAD transfers ammunition from the unit trucks to the weapons. The number of rounds to be taken from each truck is determined by taking the total requirements of the unit divided by the number of weapons needing ammunition (from IUNIT array). Transfer is accomplished by multiplying the percent of the trucks load (ITRUCK(N,6)) by the quantities in the IMIX array for the particular mix the truck is carrying (ITRUCK(N,5)).

(2) After completion of the reload, the truck is scheduled to return to the combat trains or assembly area. If the truck is empty it will return through a UNTDEP (unit departure) to the ASP or ATP for another load (see paragraph 2d. below). Otherwise, the truck will wait at the unit for the next RELOAD event.

c. UNTARV (arrival of a unit truck at the unit). UNTARV brings the truck back to the unit combat trains (if it is a maneuver unit) or the assembly area (for an artillery unit). The ammunition-on-trucks attributes in the IUNIT array are incremented by the ammunition load of the returning truck. UNTARV may schedule a RELOAD event if there is an outstanding demand for the type of ammo on the returning truck.

d. UNTDEP (departure of a unit truck from the assembly area or combat trains). If upon completion of a RELOAD event the truck is empty, a unit departure is scheduled. This event checks for the most needed type at the unit. If that ammunition type is stocked at the ATP, an ATP arrival (ATPARV) event is scheduled. The arrival time is based upon the distance from the unit to the ATP and the average speed of the truck. If the needed ammunition type is not stocked at the ATP, then the arrival time to the ASP is calculated and an ASP arrival (ASPARV) event is scheduled.

e. ATPARV (arrival of a unit truck at the ATP). When the ATPARV event is executed, the ATPARV subroutine checks to see that the ammunition type required by the arriving truck is on-hand at the servicing ATP. If the needed ammunition type is available at this ATP, ATPARV looks in queue for a free server (a piece of Materiel Handling Equipment, MHE, i.e. a forklift or a crane), and a S&P with the proper ammunition mix. If either the server or the S&P is unavailable (busy or not present), the unit truck is placed in queue to wait for a free server. When both a server and an S&P with the proper mix are available, ATPARV schedules an ATP event to load the unit truck (the server and unit truck number are passed as IPARMS when the event is scheduled), and the S&P is put in queue. If the needed ammunition type is not available at this ATP, then the truck is bumped to the associated ASP by scheduling an ASPARV.

f. ATP (service of a unit truck at the ATP). The ATP event simulates the transloading of ammunition from an S&P to a unit truck. This is accomplished by decrementing the load on the S&P and incrementing the unit truck load, using the IMIX array (for number of rounds) and the ITRUCK array (ITRUCK 6 which is the percent loaded and ITRUCK 5 which is the mix on the vehicle). If the S&P runs out of ammunition before the unit truck is full, ATP looks for another S&P to finish loading the unit truck (if none are available, the unit truck goes through an ATPARV). The empty S&P goes through an ATPAR1 (on the way to the CSA) if it is a CSA-ATP truck. If the empty S&P belongs to the ASP, it is scheduled for an ATPAR2 (for an eventual reload). When the unit truck finishes loading, it is scheduled for a UNTARV to return it to the unit. The UNTARV routine is also used (in the ATP subroutine) to schedule the return of a server to the proper queue in the ATP.

g. ATPAR1 (arrival of a CSA-ATP S&P at the ATP).

(1) The major function of ATPAR1 is to accept S&Ps coming into the ATP. This includes full trucks coming from the CSA and empty (or partially empty) trucks returning to queue from reloading a unit truck in the ATP event.

(2) ATPAR1 also handles the redistribution function of the Division Ammunition Officer (DAO). S&P trucks arriving at the DAO from the CSA (IPARM(4) will be 333) are scheduled into the ATP that needs the ammunition the most (another ATPAR1). This represents the diversion of these trucks to where they are needed the most.

Empty trucks are sent to the CSA through a CSAARV. Partially loaded trucks are put in queue.

h. CSAARV (arrival of an empty S&P at the CSA). CSAARV increments the throughput at the CSA, and sets the load attribute of the S&P to a full load (ITRUCK(N,6)=10000). If the S&P is from an ASP, then CSAARV checks each ASP to find the ASP and ammunition type that is needed the most. A CSADEP event is scheduled to return the S&P to the ATP or ASP.

i. CSADEP (departure of a full S&P from the CSA). S&Ps travel from the CSA forward in convoys. CSA to ATP convoys contain three trucks; a CSA to ASP convoy is seven trucks. The S&Ps wait in queue until there are enough going to the same place to form a convoy. When a convoy is formed, trucks are dispatched to arrive at their destination one minute apart. ASP convoys go through the ASPAR2 event. ATP convoys go to the DAO by scheduling an ATPAR1 with IPARM(4) set to 333 (see g above).

j. ATPAR2 (arrival of an ASP S&P truck at the ATP). ATPAR2 processes ASP S&P trucks at the ATP. Empty ATP-ASP S&P trucks are scheduled for an ASPAR1 to load from stocks at the ASP. Full S&P trucks returning from the ASP, as well as partially full S&P trucks (returning from loading unit trucks in the ATP), are placed in the S&P queue at the ATP, by this routine.

k. ASPAR1. ASPAR1 simulates the arrival of an ASP-ATP S&P truck at the ASP.

(1) The S&P tractor will exchange an empty trailer for a full one, if one of the right mix is available. Otherwise, ASPAR1 attempts to locate a server to load the empty trailer from the ground. If no servers are available, the tractor and empty trailer wait in queue for a returning server. If there is not sufficient ammunition available on trailers or on the ground, the S&P will be "bumped" to the rear ASP by scheduling another ASPAR1. In the case when there is no ammunition at the rear ASP, the S&P truck waits in queue.

(2) The filled truck returns to the ATP through an ATPAR2. The server is released by scheduling a UNTARV. If the ATP S&P switched trailers with a CSA S&P truck, the CSA truck is returned by scheduling an ASPAR2.

l. ASPARV (arrival of a unit truck at the ASP). ASPARV works much the same way as ATPARV, in that it sets up the unit truck for loading (at the ASP). ASPARV locates a server and schedules an ASP event to load the truck from the ground or a waiting S&P. If the needed type is out of stock at this ASP, then ASPCK will find another ASP that has the needed type. The truck must wait in queue if no ASP has the needed ammunition. MLRS trucks are self-loading and require no server. Only three MLRS may load at one time; additional MLRS must wait in queue.

m. ASP. The ASP event reloads a unit truck at the ASP. ASP will get an S&P from the queue if one is available, to load the unit truck. If the S&P runs out of ammunition, ASP schedules the return of the empty S&P truck (ASPAR2), and gets another from queue to service unit trucks needs. When no more S&Ps are available, ASP will load the unit truck from stocks on the ground.

n. ASPAR2. ASPAR2 handles ASP-CSA S&P trucks.

(1) Empty S&Ps are put into queue until seven are ready to be sent to the CSA in convoy. Trucks in a convoy are scheduled to arrive at the CSA (CSAARV) one minute apart.

(2) Full S&P trucks arriving from the CSA look for an empty ATP-ASP S&P waiting in queue for their ammunition type. If a match is found they will switch trailers, the ATP-ASP truck (full) returns to the ATP (ATPAR2). The ASP-CSA truck (empty) is scheduled through an ASPAR2 to go to the CSA. If no ATP trucks are waiting then ASPAR2 looks for a server, and puts the S&P in queue to be offloaded. If a server was found, ASPAR2 calls SERVER to offload the trucks in queue.

(3) Partially full S&Ps returning from loading unit trucks in the ASP event are scheduled for an ASPAR2 to be put in their proper queue.

o. HELARV (arrival of a helicopter (CH-47) at an artillery unit). When 155mm artillery units become critically short of ammunition, DEMAND will schedule a HELARV event. This subroutine increments the ammunition supply in the IUNIT array, and schedules a HASPAR that returns the helicopter to the ASP. It is assumed that the helicopter was idle and the cargo bags were preloaded at the ASP for the helicopter to pick up.

p. HASPAR (return of a helicopter to the ASP). HASPAR simulates the return of the helicopter to the available queue and its eventual reload in preparation for another aerial resupply. Subroutine SERVER is called to load the cargo bags of the helicopter at the ASP.

3. Special purpose routines.

a. OPERA.

(1) Each time a truck is scheduled to move (or for MHE, scheduled to be used), the routine scheduling this action calls subroutine OPERA to check the operational availability of that vehicle. Each vehicle has its own clock that keeps track of the vehicle's time to next failure (in minutes). Time to next failure is maintained (stored) in ITRUCK (N,7), and is exponentially distributed about the mean time to fail (MTTF) for the vehicle type.

(2) When a vehicle is scheduled to move from one location to another (or is in use for a period of time), the travel time is subtracted from the time to next failure. If the time to fail reaches zero, then the vehicle is delayed by a repair time, which is a random variable, log normally distributed about the mean time to repair (MTTR) for the given vehicle type.

b. INTRDK (vehicle interdiction).

(1) Whenever a subroutine schedules an event that causes a vehicle to move, it calls INTRDK to find out if the vehicle completes the move. The division area of operations is divided into two zones for interdiction purposes. Zone one extends from line of contact to the brigade rear boundary. Zone two is the area from the brigade rear boundary back to the corps storage area.

(2) The operator inputs parameters to the database in array INTER, which are derived from the scenario. For each zone, there is a maximum number of interdictions, a modulo number, and a counter which accumulates number of trucks who have travelled this CI (see array definition of INTER for more information). If the number of trucks traveling through the zone this CI is evenly divisible by the modulo for the zone (using modular arithmetic), then this truck will be killed. If a unit truck is interdicted, the time of interdiction is stored in the IUNIT array (IUNIT(N,6)) to evenly distribute the total interdictions for a CI over all units with trucks on the road.

c. DUALMX (reload mortars and 25 mm ammunition. Since mortar and 25 mm ammunition are carried on the same unit truck, and are fired at different rates, RELOAD cannot handle reloading these weapons. RELOAD calls DUALMX to reload mortars and 25 mm weapons. DUALMX finds which type has the greatest need, and loads both weapons to the same percentage of truck load. This is done so the percent load in the ITRUCK array will represent both ammunition types. As in RELOAD, partially loaded trucks are scheduled into a UNTARV. Empty trucks go through UNTDEP to the ASP.

d. LDPWDR (load powder on artillery truck at the ATP). LDPWDR is called by ATP to finish loading an artillery (unit) truck with powder and fuzes.

e. ASPCK (check alternate ASPs for ammo). ASPCK checks for the needed ammunition at the sister and rear ASPs, and schedules the unit truck into an ASPARV if ammo is available. If ammunition is unavailable at alternate ASPs, the truck is kept at the current ASP and ASPCK returns a 777 in IPARM(4) to the calling routine.

f. DEPASP (departure of a MLRS truck from the ASP or ATP). Since only three MLRS trucks may load at any one time, DEPASP brings the next MLRS truck in when one finishes loading. This is accomplished by scheduling a false UNTARV event for the newly loaded unit truck (IPARM (2)=0 for ASP; -1 for ATP). An ATP or ASP event will be scheduled for the new truck.

g. SERVER (release of a server at the ATP or ASP). SERVER is called from UNTARV to process a server when it comes available. It looks for a unit truck in queue to load in an ATP or ASP event. At the ASP, SERVER may find a CSA S&P to offload or an ATP S&P to load. When there are no more trucks to be serviced, the server is put in queue.

```

a.      SUBROUTINE MAINARM
        CHARACTER*10 AUNIT
C
        INCLUDE LOG,LIST
        INCLUDE QUENUM,LIST
        INCLUDE QUEPNT,LIST
        INCLUDE AUNIT,LIST
        DIMENSION IPARM(5)
C**** MAIN, H. JONES    DEC 78
C
C**** UNIT 2    REPORT FILE
C**** UNIT 3    INPUT DATA, COMMONS LOG, QUENUM, QUEPNT
C**** UNIT 4    OUTPUT DATA, COMMONS LOG, QUENUM, QUEPNT
C**** UNIT 5    KEYBOARD
C**** UNIT 6    DISPLAY
C**** UNIT 7    INPUT DATA, COMMON EVENTS
C**** UNIT 8    OUTPUT DATA, COMMON EVENTS
C**** UNIT 9    INPUT DATA, DEMAND FILE
C**** UNIT 11   INPUT DATA, CONVOY EVENTS
C
C          DICTIONARY
C          -----
C      DIST      The distance between two specified locations. See TULTIM.
C      IOWNER     The owner number of the vehicle; attribute(4) of the vehicle
C                  in the ITRUCK array.
C      ITKTYP     The vehicle type; attribute(1) of the vehicle in the ITRUCK
C                  array.
C      LOAD()     The number of rounds of the specified type loaded on the
C                  vehicle. see also ND().
C      MIX        The mix of ammunition on vehicle; attribute(5) of the ITRUCK
C                  array. For a given mix number, the number of rounds for
C                  each ammunition type is listed in the IMIX array.
C      MIXGET     The MIX of ammunition that an empty unit/trains truck is
C                  being sent to get from either the ATP or the ASP.
C      NASP       The ASP number from the IASP array. (1..6)
C      NATP       The ATP number from the IATP array. (1..6)
C      ND()       The number of rounds of the specified ammunition type needed,
C                  demanded, or required. See also LOAD().
C      NRONTK     The total number of rounds on the vehicle.
C      NUMAM      Ammunition type; usually used in DO LOOP control.
C      NUMRD      The total number of rounds needed, demanded or required.
C      NUMSP      The S&P number from the ITRUCK array.
C      NUMSVR     The server number from the ITRUCK array.
C      NUMTK      The truck number from the ITRUCK array.
C      NUNIT      The unit number from the IUNIT array for either units or
C                  battalions.
C      TDELA      The time delay for vehicle (or server) providing ammunition.
C      TFAIL      The duration of maintenance failure for a vehicle.
C      TGNULD     The time to load truck receiving ammunition from stocks on
C                  the ground.
C      TLOAD      The time to load vehicle receiving ammunition.
C      TMIND      The duration of non-availability due to enemy interdiction.
C      TOFFLD     The time to off-load ammunition to the ground from a truck.
C      TOTIM      The sum of all loadings or delay times.

```

```

C
C**** INITIALIZE SIMULATION
      CALL INIT
C
C**** GET AND PROCESS EACH EVENT
      5 CALL NEXTEV (KIND, IPARM, TIME)
      IF (KIND .LT. 1 .OR. KIND .GT. 19) WRITE(6,7) KIND
      7 FORMAT(' EVENT TYPE ',I5,' OUT OF RANGE')
      CALL LOOKEV (KIND+0, IPARM, TIME+0., 1)
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
      Z 150,160,170,180,190), KIND
C
C**** CHECKS AMMO DEMAND OF UNIT
      10 CALL DEMAND (IPARM)
      GO TO 5
C
C**** REPLACES ROUNDS OF AMMO AT UNIT WEAPONS
      20 CALL RELOAD (IPARM)
      GO TO 5
C
C**** DEPARTURE OF TRUCK FROM UNIT
      30 CALL UNTDEP (IPARM)
      GO TO 5
C
C**** ARRIVAL OF UNIT TRUCK AT ATP
      40 CALL ATPARV (IPARM)
      GO TO 5
C
C**** ARRIVAL OF UNIT TRUCK AT ASP
      50 CALL ASPARV (IPARM)
      GO TO 5
C
C**** SERVICE OF UNIT TRUCK FROM QUEUE AT ATP
      60 CALL ATP (IPARM)
      GO TO 5
C
C**** SERVICE OF UNIT TRUCK FROM QUEUE AT ASP
      70 CALL ASP (IPARM)
      GO TO 5
C
C**** ARRIVAL OF TRUCK AT UNIT
      80 CALL UNTARV (IPARM)
      GO TO 5
C
C**** ARRIVAL OF TRUCK AT CSA (FROM ATP OR ASP)
      90 CALL CSAARV (IPARM)
      GO TO 5
C
C**** ARRIVAL OF ATP S & P FROM CSA
      100 CALL ATPAR1 (IPARM)
      GO TO 5
C
C**** ARRIVAL OF ASP S & P AT ATP
      110 CALL ATPAR2 (IPARM)

```

```

C
C**** ARRIVAL OF ASP S & P AT ASP (FROM ATP)
  120 CALL ASPAR1 (IPARM)
      GO TO 5
C
C**** ARRIVAL OF CSA S & P AT ASP
  130 CALL ASPAR2 (IPARM)
      GO TO 5
C
C**** ARRIVAL OF HELICOPTER AT UNIT
  140 CALL HELARV (IPARM)
      GO TO 5
C
C**** ARRIVAL OF HELICOPTER AT ASP
  150 CALL HASPAR (IPARM)
      GO TO 5
C
C**** DEPARTURE OF S & P FROM CSA
  160 CALL CSADEP (IPARM)
      GO TO 5
C
C**** REPORT
  170 CALL REPORT (IPARM)
      GO TO 5
C
C**** INTERACTIVE CONTROL (MENU)
  180 CALL CONTRL (TIME)
      GO TO 5
C
C**** SIMULATION END
  190 CALL ENDSIM(IPARM)
      CALL EVSTOP
      STOP
      END

```

```

b.  SUBROUTINE ASP
      SUBROUTINE ASP (IPARM)
C**** EVENT ASP -- SERVICE OF TRUCK FROM QUEUE AT ASP.
C      MAY LOAD UNIT TRUCK FROM A S&P OR GROUND.
C      ALL EXCEPT MLRS NEED SERVERS.
C      EVENT TYPE 7
C      CALLED BY MAINARM
C      CALLS IQ, FINTK, SCHED, OPERA, INTRDK
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- 1 = ROUTINE QUEUE, 2 = MLRS QUEUE
C**** IPARM(2) -- ASP NUMBER
C**** IPARM(3) -- UNIT TRUCK NUMBER
C**** IPARM(4) -- SERVER NUMBER (FOR MLRS, SAME AS TRUCK)
C
C**** SCHEDULES -- UNTARV, ARRIVAL OF TRUCK AT UNIT
C                  OR SERVER
C                  ASPAR2, RELEASE OF S&P
C                  ASPARV, INTERDICTION UNIT TRUCK
C
C**** SCHEDULED BY  ASPARV, DEPAASP, SERVER
C
C**** (2) CALCULATES LOAD TIME AS FUNCTION OF LOAD MIX
C****      NUMBER AND NUMBER OF SERVERS ACTIVE FOR THIS QUEUE.
C
C**** NOTE !!! IN THIS ROUTINE ITRUCK(X,3) CONTAINS THE TIME THE
C**** THAT TRUCK NUMTK ARRIVED AT THE ASP !!!
C**** CHECKS -- DELAY DUE TO MTBF AND INTERDICTION.
C
      INCLUDE LOG,LIST
      DIMENSION IPARM(5),IIPARM(5)
C      LOCAL VARIABLE DEFINITION
C      NASP0 - QUEUE NUMBER TO BE SERVICED
C      NUMTK -TRUCK TO BE SERVED
C      NUMSP - S&P TRK FROM CSA TO ASP
C      MIX -AMMO MIX INDEX ON UNIT TRUCKS
C      MIXSP - MIX ON CSA TRUCK
C      NRONTK - NO. OF ROUNDS ON CSA TRUCK
C      TLOAD - TIME FOR A SERVER TO LOAD ONE TRUCK
C      ITKTYP - TYPE OF TRUCK TO BE SERVED
C      NUNIT - UNIT NUMBER OWNING THE TRUCK
C      DIST - DISTANCE BETWEEN ASP AND UNIT OR CSA AND ASP
C      TVLTIM - TRAVEL TIME
C      TFAIL - TIME LOST DUE TO FAILURE
C      TIND - TIME LOST DUE TO INTERDICTION
C      TOTIM -TIME OF SCHEDULED EVENT
C
      IQTYPE = IPARM(1)
      NASP = IPARM(2) - 10
      NUMTK = IPARM(3)
      NUMSRV = IPARM(4)
      IPARM(3) = 0
      IPARM(4) = 0

```

```

TLOAD = 0.
IFLAG = 0
NFLAG = 0

C
C FIND THE VALUE FOR THE QUEUE: 1 FOR ROUTINE, 4 FOR MIRS
KVAL = 1
IF(IQTYPE .EQ. 2) KVAL = 4
C FIND THE QUEUE WAIT TIME AND ADD IT TO TOTAL QUEUE TIME
NTQUE = TIME - ITRUCK(NUMTK,3)
ITRUCK(NUMTK,12) = ITRUCK(NUMTK,12) + NTQUE
JASP(NASP,KVAL+1) = JASP(NASP,KVAL+1) + NTQUE
C IF THIS IS THE LONGEST QUEUE WAIT RECORD IT
IF(NTQUE.GT.JASP(NASP,KVAL+2)) JASP(NASP,KVAL+2)=NTQUE
C ADD ONE TO THE NUMBER SERVED
JASP(NASP,KVAL) = JASP(NASP,KVAL) + 1
C
MIX = ITRUCK(NUMTK,5)
C**** FIND AMMO TYPE WANTED. ASSUME ONLY ONE TYPE.
DO 30 NUMAM = 1,LPPAR(1)
IF(IMIX(MIX,NUMAM) .LE. 0)GO TO 30
C**** RECORD NUMBER OF ROUNDS NEEDED -NRNDSN AND TYPE OF RDS
NRNDSN = IMIX(MIX,NUMAM)
C**** CHECK IF THERE IS AMMO ON CSA TRUCKS
NASPQ = IQ(7,NASP)
50 CALL FINTK(NASPQ,NUMAM,NUMSP,0)
C IF NO TRUCK, GO TO GROUND LD UP
IF(NUMSP .EQ. 0) GO TO 70
C FIND THE NUMBER OF ROUNDS ON NUMSP. IF SUFFICIENT, DECREMENT
C AMMO, SCHEDULE UNTARV, PUT TRUCK BACK IN ASP Q.
C IF INSUFFICIENT, EMPTY S&P TRUCK,SEND TO CSA, DECREMENT
C THE NUMBER OF ROUNDS REQUIRED, FIND ANOTHER TRUCK WITH
C PROPER AMMO OR COMPLETE LOADING FROM STOCK.
C UPDATE PER CENT ROUNDS ON THE TRUCK.
MIXSP = ITRUCK(NUMSP,5)
NRONTK = (IMIX(MIXSP,NUMAM)*ITRUCK(NUMSP,6)+9999)/10000
ITRUCK(NUMSP,13) = ITRUCK(NUMSP,13) + 1 @ LOAD FROM CSA S&P
C IF INSUFFICIENT ROUNDS GO TO 60
IF(NRNDSN .GT. NRONTK) GO TO 60
C SUFFICIENT AMMO ON TRUCK, DECREMENT AMMO ON TRUCK
ITRUCK(NUMSP,6) = 10000*(NRONTK - NRNDSN)/IMIX(MIXSP,NUMAM)
IF(ITRUCK(NUMSP,6) .EQ. 0) GO TO 60
IF(IFLAG .EQ. 1)THEN
    RNDS = NRNDSN
    RNDAV = FRNN
    TDELAY = TDELAY + IMIX(MIX,32) * (RNDS/RNDAV)
ELSE
    TDELAY = IMIX(MIX,32)
END IF
C SCHEDULE RELEASE OF S & P
IIPARM(1) = NASP + 10
IIPARM(2) = NUMSP
IIPARM(3) = 0
IIPARM(4) = 999
CALL SCHED(13,IIPARM,TIME + TDELAY) @ ASPAR2

```

```

C      INSUFFICIENT AMMO OR EXACTLY ENOUGH AMMO ON S3P
C      TIME TO SHIFT PARTIAL LOAD
50  FRNN= NRNDN
    TLOAD = IMIX(MIX,32) * NRNTK / FRNN
    IF (IDAY .EQ. 0) TLOAD = 1.54*TLOAD
    TDELAY = TDELAY + TLOAD + 5.0
    NRNDN = NRNDN - NRNTK
    ITRUCK(NUMSP,6) = 0
    IIPARM(1) = NASP + 10
    IIPARM(2) = NUMSP
    IIPARM(3) = 0
    IIPARM(4) = 555
    CALL SCHED(13,IIPARM,TIME + TDELAY)      @ ASPAR2
    IFLAG = 1
C      IF EXACTLY ENOUGH ROUNDS ON TRUCK, SEND BACK TO UNIT
    IF(NRNDN .EQ. 0) GO TO 80
C      GO GET ANOTHER S3P TRUCK TO COMPLETE THE LOAD
C      OR COMPLETE FROM ON THE GROUND STOCK
    GO TO 50
70  NRST = NRNDN
    TTLOAD = TLOAD + (IMIX(MIX,32))*(NRST / IMIX(MIX,NUMAM))
    IF (IDAY .EQ. 0) TTLOAD = 1.54*TTLOAD
    TDELAY = TDELAY + TTLOAD
C**** DECREMENT AMMO AVAILABLE
80  CONTINUE
    IASP(NASP,NUMAM*3+18) = IASP(NASP,NUMAM*3+18) - IMIX(MIX,NUMAM)
    IF(IASP(NASP,NUMAM*3+18).LT. 0)PRINT*,INT(TIME),' NO AMMO',NUMAM,
    &      ' AT ASP',NASP
    IASPAM(NASP,MIX) = IASPAM(NASP,MIX) + 1 @ INCR # TRUCKS SERVICED
C * * * DECREMENT 'DEMAND'
    IASP(NASP,NUMAM*3+19) = IASP(NASP,NUMAM*3+19) - IMIX(MIX,NUMAM)
C**** IF ARTY AMMO DECREMENT POWDER AND FUZES
    IF(NUMAM .GT. 3 .AND. NUMAM .LT. 6 .OR. NUMAM .GT. 10 .AND. NUMAM
    Z  .LT. 14)THEN
        IASP(NASP,27) = IASP(NASP,27) - IMIX(MIX,NUMAM)
        IASP(NASP,78) = IASP(NASP,78) - IMIX(MIX,NUMAM)
    END IF
    IF(NUMAM .GT.5 .AND. NUMAM .LT. 8 .OR. NUMAM .EQ. 15)THEN
        IASP(NASP,42) = IASP(NASP,42) - IMIX(MIX,NUMAM)
        IASP(NASP,79) = IASP(NASP,79) - IMIX(MIX,NUMAM)
    END IF
90  CONTINUE
C * * * DECREMENT NUMBER TRUCK IN QUEUE
    IF(NFLAG .EQ. 1) GO TO 100
    IASP(NASP,IQTYPE+11) = IASP(NASP,IQTYPE+11) - 1
    IF(IASP(NASP,IQTYPE+11) .LT. 0)PRINT*,INT(TIME),' QUE',NASP+10,
    Z  IASP(NASP,IQTYPE+11),' IN THE ASP'
C      FIND TYPE OF TRUCK AND COMPUTE TRAVEL TIME
    ITRKTP = ITRUCK(NUMTK,1)
    NUNIT = ITRUCK(NUMTK,4)
    DIST = IUNIT(NUNIT,5)
    IF(((NASP + 10) - IUNIT(NUNIT,3)) .EQ. 1 .OR. ((NASP + 10)
    Z  - IUNIT(NUNIT,3)) .EQ. -1)DIST = DIST + 15.
    IF(((NASP + 10) - IUNIT(NUNIT,3)) .EQ. 2)DIST = DIST +

```

```

TULTIM = 60. * DIST / ITYPE(ITKTYP, IDAY+1)
ITRUCK(NUMTK, 3) = 4
CALL INTRDK(NUMTK, TMIND)
IF(TMIND .LE. 0.) THEN
    CALL OPERA(NUMTK, TULTIM, TFAIL)
ELSE
    TFAIL = 0.
END IF
TOTIM = TIME + TDELAY
C SCHEDULE UNTARV
IPARM(1) = NUNIT
IPARM(2) = NUMTK
TOTIM = TIME + TDELAY + TMIND + TFAIL + TULTIM
C IF NO INTERDICTION, BYPASS.
IF(TMIND .LE. 0) GO TO 100
TOTIM = TOTIM + TDELAY
NFLAG = 1 @ TO CHARGE 2D LOAD OF AMMO
GO TO 80
100 CONTINUE
C ADD ONE TO THE NUMBER OF TRUCKS TRAVELING FROM ASP TO UNIT
ITKTYP = IUNIT(NUNIT, 1)
JUNIT(ITKTYP, 17) = JUNIT(ITKTYP, 17) + 1
C ADD TO THE NUMBER OF KILLED OR FAILED ON THIS TRIP
IF(TMIND .GT. 0) JUNIT(ITKTYP, 18) = JUNIT(ITKTYP, 18) + 1
IF(TFAIL .GT. 0) JUNIT(ITKTYP, 19) = JUNIT(ITKTYP, 19) + 1
C ADD THE WAIT AND LOAD AND TRAVEL TIME TO THE TOTAL TIME
JUNIT(ITKTYP, 20) = JUNIT(ITKTYP, 20) + NTQUE + TTLOAD + TULTIM
C
IF(TMIND .GT. 0) THEN
    IPARM(3) = IUNIT(NUNIT, 3)
    CALL SCHED(5, IPARM, TMIND + TIME) @ ASPARV
    ITRUCK(NUMTK, 6) = 0
ELSE
    CALL SCHED(8, IPARM, TOTIM) @ UNTARV
    ITRUCK(NUMTK, 6) = 10000
END IF
IF(KVAL .EQ. 1) THEN
C**** SCHEDULE THE AVAILABILITY OF THE SERVER
C
    CALL OPERA(NUMSRV, TDELAY, TTFAIL)
    DTIME = TIME + TDELAY + TTFAIL
    IPARM(1) = NASP + 10
    IPARM(2) = NUMSRV
    IPARM(3) = 1
    CALL SCHED(8, IPARM, DTIME) @ UNTARV
    ITRUCK(NUMSRV, 6) = ITRUCK(NUMSRV, 6) + TDELAY
    IF(ITRUCK(NUMTK, 1) .EQ. 1) THEN
        ITRUCK(NUMSRV, 12) = ITRUCK(NUMSRV, 12) + 1 @ 10TON COUNTER
    ELSE
        ITRUCK(NUMSRV, 11) = ITRUCK(NUMSRV, 11) + 1 @ 5TON COUNTER
    END IF
    ELSE
C
C**** SCHEDULE THE DEPARTURE OF THE MLRS TRUCK, THIS ALLOWS

```

```
DTIME = TIME + TDELAY
IPARM(1) = NASP + 10
IPARM(2) = 0
IPARM(3) = 0
CALL SCHED (8,IPARM,DTIME)      @ UNTARV
END IF
RETURN
END
```

```

C.      SUBROUTINE ASPARV
        SUBROUTINE ASPARV (IPARM)
C**** EVENT ASPARV -- ARRIVAL OF UNIT TRUCK AT ASP
C          FIND A SERVER, S&P --> SCHED ASP
C          EVENT TYPE 5
C
C          CALLED BY MAINARM
C          CALLS SCHED, IQ, ASPCK, GETQUE, PUTQUE
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- ASP NUMBER
C**** IPARM(4) -- MIX ON TRUCK COMING INTO PROGRAM
C**** THIS EVENT PUTS TRUCK IN PROPER ASP QUEUE.
C
C          SCHEDULES -- ASP SERVICE OF UNIT TRUCK FROM QUEUE AT ASP
C                      (IF ASP SERVICE FOR THIS QUEUE IS IDLE)
C                      ASPARV, (ITSELF) AS A FALSE EVENT
C          SCHEDULED BY ASP, ASPARV, ASPCK, ATP, ATPARV, DUALMX, RELOAD, UNTDEP
C*****CHANGES----IPARM(4) TO SERVER NUMBER
C          INCLUDE LOG,LIST
C          DIMENSION IPARM(5)
C
C          LOCAL VARIABLES
C          MIX - THE INDEX OF THE AMMO TYPE FROM IMIX
C          NASPQ - QUEUE FOR TRUCK
C          NUMQ - SERVER QUEUE
C          IND - INDEX TO COUNT TRUCKS QUEUE
C          NASP - ASP NUMBER
C
C          NUNIT = IPARM(1)
C          NUMTK = IPARM(2)
C          NASP = IPARM(3) - 10
C          IPARM(4) = 0
C
C          DETERMINE AMMO MIX INDEX.
C          MIX = ITRUCK(NUMTK,5)
C *** IF UNIT TRUCK ENTERING ASPARV IS FROM INTERDICTION OR BREAK-
C DOWN AND ASP # HAS CHANGED, SCHED ASPARV 30 MIN LATER.
C          IF(ITRUCK(NUMTK,2) .NE. 9) THEN @ IF EQ., OLD ASP INT'D; SKIP
C              IOWNER = ITRUCK(NUMTK,4)
C              LASP = IUNIT(IOWNER,3)
C              IF(NASP + 10 .NE. LASP) THEN
C                  IPARM(3) = LASP
C                  IPARM(4) = MIX
C                  CALL SCHED(5,IPARM,TIME + 30.)      @ ASPARV
C                  RETURN
C              END IF
C          END IF
C          ASSUME NO MLRS IN MIX
C          NASPQ = IQ(8,NASP)
C          IND = 12

```

```

C **** IF TRUCK WAS INTERDICTIONED, ADD AMMO TO UNIT ROUNDS DUE IN
  IF(ITRUCK(NUMTK,3) .EQ. 7) THEN
    DO 1 I = 1,LPPAR(6)
      NAM = I * 13 - 5
      IF(IUNIT(IPARM(1),NAM) .GT. 0) THEN
        IUNIT(NUNIT,NAM+12) = IUNIT(NUNIT,NAM+12) +
          IMIX(MIX,IUNIT(NUNIT,NAM))
      Z
    END IF
  1
  END IF
  CONTINUE
END IF
C * * * DETERMINE AMMO TYPE
  NUMAM = MIX
  IF(MIX .GT. LPPAR(7)) NUMAM = MIX - LPPAR(7)
C
C * * * IF .EQ. 9, TRUCK ARRIVED FROM ANOTHER ASP
  IF(ITRUCK(NUMTK,2) .EQ. 9) IASP(NASP,NUMAM * 3 + 19) =
    IASP(NASP,NUMAM * 3 + 19) - IMIX(MIX,NUMAM)
C
C * * * DETERMINE IF SUFFICIENT AMMO ON HAND (IF 'ON HAND' - 'DEMAND'
  < 'NEEDED')
  IF(IASP(NASP,NUMAM*3+18) - IASP(NASP,NUMAM*3+19) .LT.
    IMIX(MIX,NUMAM)) THEN
    IF(ITRUCK(NUMTK,2) .EQ. 9) GO TO 10
    CALL ASPCK(IPARM)
    IF(IPARM(4) .NE. 777) RETURN
  ELSE
    ITRUCK(NUMTK,2) = 1
  END IF
C
C SET ITRUCK(N,3) EQUAL TO THE TIME OF QUEUE ENTRY, THIS
C WILL ALLOW QUEUE WAIT TIME TO BE FOUND FOR THE TRUCK
10 ITRUCK(NUMTK,3) = TIME
C
C INCREMENT NUMBER OF TRUCKS IN QUEUE
  IASP(NASP,IND) = IASP(NASP,IND) + 1
C * * * INCREMENT 'DEMAND'
  DO 20 IAM = 1,LPPAR(1)
    IASP(NASP,IAM*3+19) = IASP(NASP,IAM*3+19) + IMIX(MIX,IAM)
  20 CONTINUE
C * * * IF EQ 9, THIS IS SECOND ASP AND NO AMMO -- PUT IN QUEUE
  IF(ITRUCK(NUMTK,2) .EQ. 9) GO TO 5
C * * * IF EQ 777, NOT ENOUGH AMMO AT ANY ASPs -- PUT IN QUEUE
  IF(IPARM(4) .EQ. 777) GO TO 5
C * * * FIND AMMO TYPE NEEDED
C **** IF MLRS SCHEDULE THE ASP EVENT
C
C IF THERE ARE LESS THAN 3 TRUCKS ALREADY LOADING
  IF(NUMAM .EQ. 10) THEN
    IF(IASP(NASP,8) .LE. 3) THEN
      IASP(NASP,8) = IASP(NASP,8) + 1
      IPARM(1) = 2
      IPARM(2) = NASP + 10
      IPARM(3) = NUMTK
      IPARM(4) = NUMTK
      CALL SCHED(7,IPARM,TIME)
      RETURN
    ELSE

```

```

        GO TO 5
    ENDIF
ENDIF
C**** LOOK FOR A SERVER AT THE ASP
NUMQ = IASP(NASP,7)
CALL GETQUE(NUMSRV,NUMQ)
IF(NUMSRV .GT. 0) THEN
C****    SCHEDULE THE ASP EVENT
        IPARM(1) = 1
        IPARM(2) = NASP + 10
        IPARM(3) = NUMTK
        IPARM(4) = NUMSRV
        ITRUCK(NUMSRV,3) = 4
        CALL SCHED(7,IPARM,TIME)           @ ASP
        RETURN
    ELSE
        IASP(NASP,20) = IASP(NASP,20) + 1   @ 'NO SERVER' CNTR
    END IF
C    PUT TRUCK INTO PROPER QUEUE
5 CALL PUTQUE(NUMTK,NASPQ)
RETURN
END

```

```

d.  SUBROUTINE ASPCK
      SUBROUTINE ASPCK (IPARM)
C**** ASPCK -- CHECKS FOR SUFFICIENT AMMO AT ALTERNATE ASPS (TO BUMP TO)
C      CALLED BY ASPARV
C      CALLS SCHED, OPERA
C
C**** L TOLIN      MAY 82
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- ASP NUMBER
C**** IPARM(4) -- MIX ON TRUCK COMING IN (RETNS 777 IF NO ASP HAS AMMO)
C
C      SCHEDULES -- ASPARV
C                  UNIT TRUCK TO SISTER OR REAR ASP IF PARENT ASP
C                  HAS INSUFFICIENT AMMO - KEEPS UNIT TRUCK AT
C                  PARENT ASP IF INSUFFICIENT AMMO AT
C                  SISTER OR REAR ASPS
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C      INTEGER ONHAND,DEMAND
C
C      LOCAL VARIABLES
C      MIX - THE INDEX OF THE AMMO TYPE FROM IMIX
C      KVAL - FLAG 1= ROUTINE, 4= MLRS QUEUE
C      NASP - ASP NUMBER
C
C      DETERMINE AMMO MIX INDEX.
C      NASP = IPARM(3) - 10
C      MIX = ITRUCK(IPARM(2),5)
C *** DETERMINE AMMO TYPE (NUMAM) FOR ASP ARRAY
C      NUMAM = MIX
C      IF(MIX .GT. LPPAR(7))NUMAM = MIX - LPPAR(7)
C      ONHAND = NUMAM*3+18
C      DEMAND = NUMAM*3+19
C      NRDSND = IMIX(MIX,NUMAM)
C      NUMTK = IPARM(2)
C ** DETERMINE 'SISTER' ASP NUMBER(NSASP)
C      J = NASP/2
C      JJ = J * 2
C      IF(NASP .EQ. JJ)THEN
C          NSASP = NASP -1
C      ELSE
C          NSASP = NASP + 1
C      END IF
C ** IF SISTER ASP HAS BEEN INTERDICTED, CHECK REAR ASP
C ** DETERMINE IF 'SISTER' ASP HAS SUFFICIENT AMMO
C      IF(IASP(NSASP,ONHAND) - IASP(NSASP,DEMAND) .GE.
C      * NRDSND .AND. NASP+10 .NE. ISERV(6))THEN
C          IASP(NSASP,DEMAND) = IASP(NSASP,DEMAND) + IMIX(MIX,NUMAM)
C          IPARM(3) = NSASP + 10
C          DDLAY = 90. - IDAY * 30.
C          CALL OPERA(NUMTK,DDLAY,TFAIL)

```

```

CALL SCHED(5,IPARM,DDTIME)          @ ASPARV
C FIND THE VALUE FOR THE QUEUE: 1 FOR ROUTINE, 4 FOR MLRS
KVAL = 1
IF(NUMAM .EQ. 10)KVAL = 4
C FIND THE QUEUE WAIT TIME AND ADD IT TO TOTAL QUEUE TIME
NTQUE = TIME - ITRUCK(NUMTK,3)
JASP(NSASP,KVAL+1) = JASP(NSASP,KVAL+1) + NTQUE
C IF THIS IS THE LONGEST QUEUE WAIT, RECORD IT
IF(NTQUE.GT.JASP(NSASP,KVAL+2))JASP(NSASP,KVAL+2)=NTQUE
ITRUCK(NUMTK,2) = 9
ITRUCK(NUMTK,3) = 9
ITRUCK(NUMTK,13) = ITRUCK(NUMTK,13) + 1 @ BUMP COUNTER TO RASP
WRITE(6,100)NUMTK,MIX,NASP + 10,NSASP + 10,TIME
100 FORMAT(' UNIT TRUCK',I4,' MIX',I3,' SENT FROM ASP',I3,
Z ' TO SISTER ASP',I3,' AT TIME',F8.1)
ELSE
C ** DETERMINE 'REAR' ASP NUMBER(NRASP)
NRASP = IASP(NASP,11) - 10
IF (IASP(NASP,2) .EQ. 0 .AND. NRASP .GT. 0) THEN
IASP(NRASP,DEMAND) = IASP(NRASP,DEMAND) + IMIX(MIX,NUMAM)
IPARM(3) = NRASP + 10
DIST = IASP(NASP,1) - IASP(NRASP,1)
ITKTYP = ITRUCK(NUMTK,1)
TVLTIM = 60. * DIST/ITYPE(ITKTYP,IDAY + 1)
CALL OPERA(NUMTK,TVLTIM,TFAIL)
TOTIM = TIME + TVLTIM + TFAIL
CALL SCHED(5,IPARM,TOTIM)          @ ASPARV
C FIND THE VALUE FOR THE QUEUE: 1 FOR ROUTINE, 4 FOR MLRS
KVAL = 1
IF(NUMAM .EQ. 10)KVAL = 4
C FIND THE QUEUE WAIT TIME AND ADD IT TO TOTAL QUEUE TIME
NTQUE = TIME - ITRUCK(NUMTK,3)
JASP(NRASP,KVAL+1) = JASP(NRASP,KVAL+1) + NTQUE
C IF THIS IS THE LONGEST QUEUE WAIT, RECORD IT
IF(NTQUE.GT.JASP(NRASP,KVAL+2))JASP(NRASP,KVAL+2)=NTQUE
ITRUCK(NUMTK,2) = 9
ITRUCK(NUMTK,3) = 9
ITRUCK(NUMTK,13) = ITRUCK(NUMTK,13) + 1 @ BUMP CNTR TO RASP
WRITE(6,400)NUMTK,MIX,NASP + 10,NRASP + 10,TIME
400 FORMAT(' UNIT TRUCK',I4,' MIX',I3,' SENT FROM ASP',I3,
Z ' TO REAR ASP',I3,' AT TIME',F8.1)
ELSE
C ** NO AMMO OF NUMAM RETURN
ITRUCK(NUMTK,3) = TIME
IPARM(4) = 777
WRITE(6,500)NUMTK,MIX,NASP + 10,TIME
500 FORMAT(' UNIT TRUCK',I4,' MIX',I3,' HELD AT ASP',I3,
Z ' AT TIME',F8.1)
END IF
END IF
RETURN
END

```

```

e.      SUBROUTINE ASPAR1
        SUBROUTINE ASPAR1 (IPARM)
C**** EVENT ASPAR1 -- ARRIVAL OF ASP-ATP S&P AT ASP (FROM ATP)
C          TRUCK IS SERVICED(MAY PICKUP FULL TRAIL) AND
C          RETURNED TO ATP.
C      EVENT TYPE 12
C      CALLED BY MAINARM
C      CALLS IQ, PUTQUE, OPERA, SCHED, FINTK, GETQUE, INTRDK
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- ATP NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- ASP NUMBER
C**** IPARM(4) -- DIST FROM REAR ASP TO ATP
C
C      SCHEDULES      ASPAR1(ITSELF)
C                      ASPAR2, EMPTY CSA S&P BACK
C                      ATPAR2, ATP S&P BACK
C                      UNTARV, SERVER RETURN
C      SCHEDULED BY ASPAR1, ATPAR2, SERVER
C**** CHECKS      -- DELAY DUE TO MTRF AND INTERDICTION
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5),IIPARM(5)
C      INTEGER ONHAND,DEMAND,ONWAY,AVAIL
C      DO 1 I = 1,5
C          IIPARM(I) = 0
1      CONTINUE
C**** LOCAL VARIABLES DEFINITION
C      MIX - AMMO ON TRUCK INDEX TO IMIX
C      NUMAM - AMMO TYPE
C      TVLTIM - TRAVEL TIME TO ATP
C      ITKTYP - TRUCK TYPE
C      TFAIL - TIME LOST DUE TO TRUCK FAILURE
C      THIND -TIME LOST DUE TO INTERDICTION
C      TMLD - TIME TO LOAD AMMO AT ASP
C      TOTIM - TIME OF ARRIVAL AT REAR ASP
C      TOTTIM - TIME OF ARRIVAL AT  ATP
C      HITI - HITCH TIME
C
C      NATP = IPARM(1)
C      NUMSP = IPARM(2)
C      NASP = IPARM(3) - 10
C      RDIST = IPARM(4)
C
C                      @ .GT. 0 IF FROM FORWARD ASP
C
C      IFLAG = 0
C      NEGSVR = 0
C      TDELAY = 0.
C      TLOAD = 0.
C      HITI = 0.          @ UNHITCH/HITCH TIME
C      FIND AMMO MIX INDEX ON THE TRUCK - MIX
C      MIX = ITRUCK(NUMSP,5)
C * * * DETERMINE AMMO TYPE(NUMAM)

```

```

NRDSND = IMIX(MIX,NUMAM)
C      SET ITRUCK(N,3) EQUAL TO THE TIME OF QUEUE ENTRY, THIS
C      ALLOW QUEUE WAIT TIME TO BE FOUND FOR THE TRUCK
      ITRUCK(NUMSP,3) = TIME
C * * * CHECK FOR SUFFICIENT AMMUNITION
      ONHAND = NUMAM * 3 + 18
      DEMAND = NUMAM * 3 + 19
      ONWAY = NUMAM * 3 + 20
      AVAIL = IASP(NASP,ONHAND) - IASP(NASP,DEMAND)
      IF(AVAIL .LT. NRDSND)THEN
        KRASP = IASP(NASP,11) - 10 @ REAR ASP +
        IF(IASP(KRASP,ONHAND) - IASP(KRASP,DEMAND) +
$ IASP(KRASP,ONWAY) .LT. NRDSND)THEN
          CALL PUTQUE(NUMSP,IQ(11,NASP))
          IASP(NASP,DEMAND) = IASP(NASP,DEMAND) + NRDSND
          WRITE(6,25)NUMSP,MIX,NASP+10,TIME
25      FORMAT(' ATP S&P',I3,' MIX',I3,' KEPT AT ASP',I3,
$          ' AT TIME',F8.1)
          RETURN
        END IF
C      * * * SCHEDULE ASPAR1 TO REAR ASP
      IPARM(3) = KRASP + 10
      ITRUCK(NUMSP,3) = 5
      DIST = IASP(NASP,1) - IASP(KRASP,1)
      ITKTYP = ITRUCK(NUMSP,1)
      TVLTIM = 60. * DIST / ITYPE(ITKTYP,IDAY+3)
      CALL OPERA(NUMSP,TVLTIM,TFAIL)
      TOTIM = TIME + TVLTIM + TFAIL
      IPARM(4) = DIST + IATP(NATP,2)
      CALL SCHED(12,IPARM,TOTIM) @ ASPAR1
      ITRUCK(NUMSP,14) = ITRUCK(NUMSP,14) + 1 @ INCREMENT + BUMPS TO RASP
      WRITE(6,35)NUMSP,MIX,NASP+10,KRASP+10,TIME
35      FORMAT(' ATP S&P',I3,' MIX',I3,' BUMPED FROM ASP',I3,
$          ' TO REAR ASP',I3,' AT TIME',F8.1)
      RETURN
    END IF
C
C * * * SEARCH FOR FULL S & P
      NASPQ = IQ(7,NASP)
      CALL FINTK(NASPQ,MIX,NCSASP,1)
      IF(NCSASP .EQ. 0)GO TO 40
C * IF .GE., HAVE FOUND FULL CSA S & P--ASSUME SWAP IN 30 MIN
      IF (ITRUCK(NCSASP,6) .GE. 10000)THEN
        ITRUCK(NCSASP,6) = 0
        IIPARM(1) = NASP + 10
        IIPARM(2) = NCSASP
        IIPARM(3) = 0
        IIPARM(4) = 555
        CALL SCHED(13,IIPARM,TIME + 30.) @ ASPAR2
C * NOW SCHED ASP-ATP S & P TO ATP
        HITI = 30.
        ITRUCK(NUMSP,11) = ITRUCK(NUMSP,11) + 1 @ INCREMENT THROUGHPUT
        NEGSUR = 1
        GO TO 80

```

```

C * IF PARTIAL OR NO LOAD ON CSA S & P, TRY TO FIND SERVER
40  NSVRQ = IASP(NASP,7)
   CALL GETQUE(NUMSVR,NSVRQ)
   IF(NUMSVR .EQ. 0) THEN                                @ NO SERVER -> QUEUE
       IF(NCSASP .NE. 0) CALL PUTQUE(NCSASP,NASPG)
       CALL PUTQUE(NUMSP,IQ(11,NASP))
       RETURN
   ELSE
       JASP(NASP,7) = JASP(NASP,7) + 1    @ INCREMENT & SERVED
       ITRUCK(NUMSVR,3) = 4
       IF(NCSASP .EQ. 0) GO TO 70
       GO TO 55
   END IF
END IF

C
C ** SERVER AVAILABLE--SEE IF ANOTHER S & P HAS AMMO TYPE NEEDED
50  CALL FINTK(NASPG,NUMAM,NCSASP,0)
C   IF NO TRUCK, GO TO GRND LOAD
   IF(NCSASP .EQ. 0) GO TO 70
C   FIND THE NUMBER OF ROUNDS ON NCSASP. IF SUFFICIENT, DECREMENT
C   AMMO, SCHEDULE UNTARV, PUT TRUCK BACK IN ASP Q.
C   IF INSUFFICIENT, EMPTY S&P TRUCK, SEND TO CSA, DECREMENT
C   THE NUMBER OF ROUNDS REQUIRED, FIND ANOTHER TRUCK WITH
C   PROPER AMMO OR COMPLETE LOADING FROM STOCK.
C   UPDATE PER CENT ROUNDS ON THE TRUCK.
35  NRONTK = (IMIX(MIX,NUMAM)*ITRUCK(NCSASP,6)+9999)/10000
   ITRUCK(NUMSVR,13) = ITRUCK(NUMSVR,13) + 1 @ LOAD FROM CSA S&P
C   IF INSUFFICIENT ROUNDS GO TO 60
   IF(NRDSND .GT. NRONTK) GO TO 60
C   SUFFICIENT AMMO ON TRUCK, DECREMENT AMMO ON TRUCK
   ITRUCK(NCSASP,6) = 10000*(NRONTK - NRDSND)/IMIX(MIX,NUMAM)
   IF(ITRUCK(NCSASP,6) .EQ. 0) GO TO 60
   IF(IFLAG .EQ. 1) THEN
       RNDS = NRDSND
       RNDV = FRNN
       TDELAY = TDELAY + IMIX(MIX,32) * (RNDS/RNDV)
   ELSE
       TDELAY = IMIX(MIX,32)
   END IF
C   SCHEDULE RELEASE OF S & P
   IIPARM(1) = NASP + 10
   IIPARM(2) = NCSASP
   IIPARM(3) = 0
   IIPARM(4) = 999
   CALL SCHED(13,IIPARM,TIME + TDELAY)    @ ASPAR2
   GO TO 90
C   INSUFFICIENT AMMO OR EXACTLY ENOUGH AMMO ON S&P
C   TIME TO SHIFT PARTIAL LOAD
60  FRNN = NRDSND
   TLOAD = IMIX(MIX,32) * NRONTK / FRNN
   IF(IDAY .EQ. 0) TLOAD = 1.54*TLOAD
   TDELAY = TDELAY + TLOAD + 5.0
   NRDSND = NRDSND - NRONTK
   ITRUCK(NCSASP,6) = 0

```

```

IIPARM(2) = NCSASP
IIPARM(3) = 0
IIPARM(4) = 555
CALL SCHED(13,IIPARM,TIME + TDELAY)      @ ASPAR2
IFLAG = 1
C   IF EXACTLY ENOUGH ROUNDS ON TRUCK, SEND BACK TO UNIT
   IF(NRDSND .EQ. 0) GO TO 90              @ RTN, SERVING S&P EMPTY
C   GO GET ANOTHER S&P TRUCK TO COMPLETE THE LOAD
C   OR COMPLETE FROM ON THE GROUND STOCK
   GO TO 50
70 NRST = NRDSND
   TTLOAD = TLOAD + (IMIX(MIX,32))*(NRST / IMIX(MIX,NUMAM))
   IF (IDAY .EQ. 0) TTLOAD = 1.54*TTLOAD
   TDELAY = TDELAY + TTLOAD
C * * * DECREMENT ASP AMMO
90 IASP(NASP,ONHAND) = IASP(NASP,ONHAND) - NRDSND
C   INCREMENT AMMO USED FROM ASP
   IASPAM(NASP,MIX) = IASPAM(NASP,MIX) + 1
C   INCREMENT ROUNDS ON-THE-WAY TO ATP
   IATP(NATP,ONHAND+2) = IATP(NATP,ONHAND+2) + IMIX(MIX,NUMAM)
C   SCHEDULE ATPAR2, COMPUTE NECESSARY PARAMETERS
   ITKTYP = ITRUCK(NUMSP,1)
   TULTIM = 60. * IATP(NATP,2) / ITYPE(ITKTYP,IDAY+3)
   IF(RDIST .GT. 0.)TULTIM = 60. * RDIST / ITYPE(ITKTYP,IDAY+3)
   IPARM(4) = MIX
   NTIQ = TIME - ITRUCK(NUMSP,3)
   ITRUCK(NUMSP,12) = ITRUCK(NUMSP,12) + NTIQ @ TRUCK TIME IN QUEUE
   ITRUCK(NUMSP,13) = ITRUCK(NUMSP,13) + 1      @ * * @ ASP
   JASP(NASP,8) = JASP(NASP,8) + NTIQ
   IF(NTIQ .GT. JASP(NASP,9))JASP(NASP,9) = NTIQ
   ITRUCK(NUMSP,3) = 4
C   COMPUTE INTERDICTION TIME LOST
   CALL INTRDK(NUMSP,THIND)
   IF(THIND .LE. 0.) THEN
C       COMPUTE TIME LOST DUE TO TRUCK FAILURE
       CALL OPERA(NUMSP,TULTIM,TFAIL)
   ELSE
       TFAIL = 0.
   END IF
C   CONSIDER LOAD TIME AT ASP WHICH MIGHT BE ZERO
   TMLD = IMIX(MIX,32)
C**** IF NO INTERDICTION, BYPASS
   IF(THIND .GT. 0.) THEN
C****   DECREMENT AMMO AGAIN SINCE LOST A TRUCK LOAD
       IASP(NASP,ONHAND) = IASP(NASP,ONHAND) - NRDSND
       IASPAM(NASP,MIX) = IASPAM(NASP,MIX) + 1
C****   ADD ANOTHER LOAD TIME
       THIND = THIND + TMLD
   END IF
C   SCHEDULE ARRIVAL AT ATP AT TIME TOTTIM
   TOTTIM = TIME + TULTIM + THIND + TFAIL + TMLD + HITI
   ITRUCK(NUMSP,6) = 10000
   CALL SCHED(11,IPARM,TOTTIM)            @ ATPAR2
C

```

```

IF(NEGSUR .EQ. 1)RETURN
CALL OPERA(NUMSUR,TMLD,TFAIL)
SVRTI = TIME + TMLD + TFAIL
IPARM(1) = NASP + 10
IPARM(2) = NUMSUR
IPARM(3) = 1
IPARM(4) = 0
CALL SCHED(8,IPARM,SVRTI)          @ UNTARV
ITRUCK(NUMSUR,10) = ITRUCK(NUMSUR,10) + 1 @ ATP-ASP S&P LOADUP
ITRUCK(NUMSUR,6) = ITRUCK(NUMSUR,6) + INT(TMLD)
C
RETURN
END

```

```

f.      SUBROUTINE ASPAR2
        SUBROUTINE ASPAR2 (IPARM)
C***** EVENT ASPAR2 -- ARRIVAL OF S&P TRUCK AT ASP FROM CSA
C          MAY SWITCH TRAIL WITH ATP S&P
C      EVENT TYPE 13
C      CALLED BY MAINARM
C      CALLS      PUTQUE, GETQUE, OPERA, INTRDK, SCHED, IQ, SERVER
C***** D. REMEN SEP 80
C***** IPARM(1) -- ASP NUMBER
C***** IPARM(2) -- TRUCK NUMBER
C***** IPARM(3) --
C***** IPARM(4) -- 555: S & P IS EMPTY  999: PARTIAL LOAD RETURN
C          444: ARRIVAL FROM DAO
C***** SCHEDULES -- CSAARV, ARRIVAL OF S&P TRUCK AT CSA
C          ASPAR2, RETURN OF EMPTY ASP S&P
C          ATPAR2, RETURN OF EMPTY ATP S&P
C      SCHEDULED BY ASP, ASPAR1, ASPAR2, ATPAR1, CSADEP, SERVER, ATPAR2
C***** CHECKS -- DELAY DUE TO MTBF AND INTERDICTION
C      NOTE:      EACH TRACTOR HAS TWO TRAILERS!  ASP S&P TRACTORS STAY
C      -----    WITH THE EMPTY TRAILER.  (ATP TRACTORS STAY WITH THE FULL
C      TRAILER).
C
        INCLUDE LOG,LIST
        DIMENSION IPARM(5),ISPQ(10),IIPARM(5)
        INTEGER ONHAND
C
C***** LOCAL VARIABLES:
C***** MIX -- AMMO MIX NUMBER ON TRUCK
C***** DIST -- DIST TO ASP OR CSA
C***** TULTIM -- TRAVEL TIME
C***** IKTYP -- TRUCK TYPE
C***** TFAIL -- DELAY ENROUTE DUE TO FAILURE
C***** TOTIM -- TIME OF ARRIVAL OF TRUCK AT ASP OR CSA
C***** THIND -- INTERDICTION TIME DELAY
C***** INDEX -- NUMBER OF QUEUE FOR CSA - ASP TRUCK
C***** IND -- INDEX FOR AMMO INVENTORY CONTROL IN IASP
C***** ISPQ() -- S&P NUMBERS FROM QUEUE TO FORM A CONVOY
C
        NASP=IPARM(1)-10
        NUMSP = IPARM(2)
        ISTAT = IPARM(4)
C
C***** FIND THE MIX ON THE TRUCK
        MIX = ITRUCK(NUMSP,5)
        IF(MIX.LE. 0) THEN
            WRITE(6,1) NUMSP
1          FORMAT(' ASPAR2 -- ZERO MIX ON TRUCK ',I4)
            RETURN
        END IF
C
        IQUT=0
C*****LOOK FOR OFFLOADED S&P RETURNING
        IF(ISTAT.EQ.555)THEN

```

```

IASP(NASP,6)=IASP(NASP,6)+1
CALL PUTQUE(NUMSP,IASP(NASP,4))
ITRUCK(NUMSP,3) = TIME
END IF
IF(ISTAT .EQ. 999)THEN
    CALL PUTQUE(NUMSP,IASP(NASP,4))
    RETURN
END IF

C
C*****CHECK FOR EMPTY S3P TO RETURN TO CSA IN CONVOY
IF(IASP(NASP,6).GE.7)THEN
    DO 3 I = 1,10
        ISPQ(I) = 0
3    CONTINUE
    CALL GETQUE(NUMSP,IASP(NASP,4))
    KOUNT=1
    IF(NUMSP.EQ.0)RETURN
    NCHKSP=NUMSP
    CALL PUTQUE(NCHKSP,IASP(NASP,4))
13    CALL GETQUE(NUMSP,IASP(NASP,4))
    IF(ITRUCK(NUMSP,6).EQ.0)THEN
        ISPQ(KOUNT)=NUMSP
        IF(NUMSP.EQ.NCHKSP)GO TO 11
        IF(KOUNT .EQ. 7)GO TO 11
        KOUNT=KOUNT + 1
        GO TO 13
    ELSE
        CALL PUTQUE(NUMSP,IASP(NASP,4))
        IF(NUMSP.EQ.NCHKSP)GO TO 11
        GO TO 13
    END IF
END IF

C
C*****HAVE ALL THE EMPTY TRAILERS,FIND DISTANCE AND SCHEDULE
11    DIST=IASP(NASP,1)
    ITKTYP=ITRUCK(NUMSP,1)
    TVLTIM=60.*DIST/ITYPE(ITKTYP,IDAY+3)
    TS=1.
    DO 12 JJ=1,KOUNT
        NUMSP = ISPQ(JJ)
        NQTI = TIME - ITRUCK(NUMSP,3)
        ITRUCK(NUMSP,13) = ITRUCK(NUMSP,13) + NQTI
        ITRUCK(NUMSP,14) = ITRUCK(NUMSP,14) + 1    @ # TIMES AT ASP
        ITRUCK(NUMSP,3) = 4
        CALL INTRDK(NUMSP,TMIND)
        IF(TMIND .LE. 0.) THEN
            CALL OPERA(NUMSP,TVLTIM,TFAIL)
        ELSE
            TFAIL = 0.
        END IF
        TOTIM=TVLTIM+TIME+TFAIL+TMIND+TS
        IIPARM(1)=NASP+10
        IIPARM(2)=NUMSP
        IIPARM(3)=0
        IIPARM(4) = 0

```

```

        TS=TS+1.
        IASP(NASP,6)=IASP(NASP,6)-1
12      CONTINUE
      END IF
      IF(IOUT.EQ.555)RETURN
C
C * * * INCREMENT AMMO ON HAND AT THIS ASP/ DECR ON- -WAY
      NUMAM = MIX - LPPAR(8)
      ONHAND = NUMAM * 3 + 19
      IASP(NASP,ONHAND) = IASP(NASP,ONHAND) + (IMIX(MIX,NUMAM)
Z * ITRUCK(NUMSP,6) + 9999) / 10000
      IASP(NASP,ONHAND+2) = IASP(NASP,ONHAND+2) - (IMIX(MIX,NUMAM)
Z * ITRUCK(NUMSP,6) + 9999) / 10000
      ICSTYP = NUMAM
C
C * * * * ARRIVAL COUNTERS
      IASPSP(NASP,NUMAM) = IASPSP(NASP,NUMAM) + 1
      IF (ISTAT .EQ. 444) THEN
        ITRUCK(NUMSP,11) = ITRUCK(NUMSP,11) + 1 @ # ARRIVALS FROM DAO
        IASPAM(NASP,70+ICSTYP) = IASPAM(NASP,70+ICSTYP) + 1
      ELSE
        ITRUCK(NUMSP,10) = ITRUCK(NUMSP,10) + 1 @ # ARRIVALS FROM CSA
        IASPAM(NASP,90+ICSTYP) = IASPAM(NASP,90+ICSTYP) + 1
      END IF
C**** UPDATE TRUCK STATUS TO THE CSA-ASP QUEUE
      ITRUCK(NUMSP,3)= 3
C** LOOK FOR ASP-ATP S&P OF SAME MIX
      IF(ICSTYP .GT. 10) GO TO 30
      NATPQ = IQ(11,NASP)
      CALL GETQUE(NATPSP,NATPQ)
      IF(NATPSP .EQ. 0)GO TO 30
      IATPCK = NATPSP
      CALL PUTQUE(NATPSP,NATPQ)
40    CALL GETQUE(NATPSP,NATPQ)
      MIXX = ITRUCK(NATPSP,5)
      IATPTY = MIXX - LPPAR(8)
      IF (ICSTYP .EQ. IATPTY) GO TO 70
      CALL PUTQUE(NATPSP,NATPQ)
      IF(IATPCK .EQ. NATPSP) GO TO 30
      GO TO 40
C** HAVE FOUND ASP-ATP S&P WITH SAME AMMO TYPE:
C** SWAP LOAD FROM CSA TO ATP S&P
C** SCHEDULE RELEASE OF CSA S&P
70    ITRUCK(NUMSP,6) = 0
      IPARM(4) = 555
      CALL SCHED(13,IPARM,TIME+30.) @ ASPAR2
C** SCHEDULE ASP-ATP S&P TO ATP
      NATP = ITRUCK(NATPSP,4) - 75
      IPARM(1) = NATP
      IPARM(2) = NATPSP
      IPARM(3) = NASP
      IPARM(4) = 0
      DIST = IATP(NATP,2)
      ITXTYP = ITRUCK(NATPSP,1)

```

```

CALL INTRDK(NATPSP,THIND)
IF(THIND .LE. 0.) THEN
    CALL OPERA(NATPSP,TULTIM,TFAIL)
ELSE
    TFAIL = 0.
END IF
TOTIM = TIME + TULTIM + TFAIL + THIND + 30. @ 30 MIN UNH/H T1
CALL SCHED(11,IPARM,TOTIM) @ ATPAR2
C
NTIQ = TIME - ITRUCK(NATPSP,3)
JASP(NASP,8) = JASP(NASP,8) + NTIQ
IF (NTIQ .GT. JASP(NASP,9)) JASP(NASP,9) = NTIQ
ITRUCK(NATPSP,12) = ITRUCK(NATPSP,12) + NTIQ
ITRUCK(NATPSP,11) = ITRUCK(NATPSP,11) + 1 @ THRU PUT COUNTER
C
ITRUCK(NATPSP,3) = 4
ITRUCK(NATPSP,6) = 10000
C
JASP(NASP,7) = JASP(NASP,7) + 1 @ INCREMENT ASP-ATP TRUCKS SERVED
C
IASPAM(NASP,60+ICSTYP) = IASPAM(NASP,60+ICSTYP) + 1 @ # SERVED
IASP(NASP,ONHAND) = IASP(NASP,ONHAND) - IMIX(MIXX,IATPTY)
C
DEMAND = ONHAND + 1
IASP(NASP,ONHAND+1) = IASP(NASP,ONHAND+1) - IMIX(MIXX,IATPTY)
C
RETURN
C*****PUT OTHER TRUCKS INTO THE QUE
30 INDEX = IQ(7,NASP)
CALL PUTQUE(NUMSP,INDEX)
C
C **** IF SERVER AVAILABLE, OFF-LOAD ARRIVING CSA 93P
IQSVR = IASP(NASP,7)
CALL GETQUE(NUMSVR,IQSVR)
IF(NUMSVR .GT. 0)THEN
    IPARM(2) = NUMSVR
    IPARM(3) = 1
    CALL SERVER(IPARM)
END IF
RETURN
END

```

```

9.      SUBROUTINE ATP
          SUBROUTINE ATP (IPARM)
C**** EVENT ATP -- SERVICE OF TRUCK FROM QUEUE AT ATP.
C          LOADS OFF S&P (USE FORKLIFT/DRAIN) OR OFF THE GROUND.
C      EVENT TYPE 6
C      CALLED BY MAINARM
C      CALLS      IQ, FINTK, SCHED, OPERA, LDFWDR (FOR ARTY), INTROK
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- 1 = ARTILLERY QUEUE,      2 = MANUEVER QUEUE
C**** IPARM(2) -- ATP NUMBER
C**** IPARM(3) -- UNIT TRUCK NUMBER
C**** IPARM(4) -- SERVER NUMBER
C      SCHEDULES  ASPARV, FOR INTRODUCED UNIT TRUCKS
C                  ATPAR1, CSA S&P -> ATP
C                  ATPARV, RESCHEDULE ATP EVENT
C                  UNTARV, SERVER OR UNIT TRUCK RETURNS
C                  ATPAR2, ASP S&P -> ATP
C      SCHEDULED BY ATPARV, DEPASP, SERVER
C
C**** NOTE !!! IN THIS ROUTINE NUMTK(N,3) CONTAINS THE TIME THE
C      TRUCK ENTERED THE QUEUE !
C
C      (1) TAKES TRUCK OUT OF ITS QUEUE
C      (2) CALCULATES LOAD TIME AS FUNCTION OF LOAD MIX
C          NUMBER AND NUMBER OF SERVERS ACTIVE FOR THIS QUEUE.
C
C**** CHECKS -- DELAY IN ARRIVAL DUE TO MTBF AND INTERDICTION.
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5), IIPARM(5)
C      LOCAL VARIABLE DEFINITION
C      NUMTK - TRUCK TO BE SERVED
C      MIX - INDEX OF AMMO MIX ON TRUCK
C      NRNDSN - NUMBER OF ROUNDS NEEDED BY THE TRUCK NUMTK
C      NUMAM - TYPE OF ROUNDS NEEDED BY NUMTK
C      NATP - ATP NUMBER
C      HRND - NUMBER OF POWDER CHARGES NEEDED
C      NATPQ -- NUMBER OF THE ASP<->ATP TRUCK QUEUE
C      NUMSP - NUMBER OF ASP ATP TRUCK
C      NRONGSP - NUMBER OF ROUNDS ON SUPPLY S&P
C      MIXX - MIX INDEX OF AMMO ON SUPPLY S&P
C      DIST - ROAD DIST TO BE TRAVELED
C      TVLTIM - ROAD TRAVEL TIME
C      TFAIL - TIME DELAY DUE TO FAILURE
C      TIND - TIME DELAY DUE TO INTERDICTION
C      TOTTIM - TIME TO SCHEDULE ATP OR ASP ARRIVAL
C      TPAR - TIME REQUIRED TO SHIFT A PARTIAL LOAD
C      ETIME - DELAY TIME FOR ATPARV
C      DTIME - DELAY TIME FOR UNTARV
C      FRNN - REAL VARIABLE FOR NUMBER OF ROUNDS NEEDED
C      TLOAD - LOAD TIME

```

```

9999 TLOAD = 0.
      TDELAY = 0.
      IFLAG = 0.
      TTDEL = 0.

C
      IQTYPE = IPARM(1)
      NATP = IPARM(2)
      NUMTK = IPARM(3)
      NUMSVR = IPARM(4)
      IPARM(3) = 0
      IPARM(4) = 0

C
      DO 1 I = 1,5
          IIPARM(I) = 0
1 CONTINUE
C      FIND THE QUEUE WAIT TIME AND ADD IT TO TOTAL QUEUE WAIT TIME
      NTQUE = TIME - ITRUCK(NUMTK,3)
      ITRUCK(NUMTK,12) = ITRUCK(NUMTK,12) + NTQUE
      JATP(NATP,KVAL+1) = JATP(NATP,KVAL+1) + NTQUE
C      IF THIS IS THE LARGEST WAIT, STORE IT
      IF(NTQUE.GT.JATP(NATP,KVAL+2)) JATP(NATP,KVAL+2)=NTQUE
      IF(ITRUCK(NUMTK,6) .GT. 0) TTDEL = 15.
C      FIND AMMO MIX INDEX OF TRUCK MIX
      MIX = ITRUCK(NUMTK,5)
C      FIND AMMO TYPE WANTED. ASSUME ONLY ONE TYPE
      NUMAM = MIX
      IF(MIX .GT. LPPAR(7)) NUMAM = MIX - LPPAR(7)
C      RECORD NUMBER OF ROUNDS NEEDED - NRNDN
      NRNDN = IMIX(MIX,NUMAM)-((ITRUCK(NUMTK,6)*IMIX(MIX,NUMAM)+
      Z 9999)/10000)
C      NOW TO LOCATE S & P CONTAINING PROPER TYPE OF AMMO
C      FIRST CHECK CSA S & PS. PASS AMMO AND QUEUE TO CHECK.
      NATPQ = IQ(IATPSD(2),NATP)
130 CALL FINTK(NATPQ,NUMAM,NUMSP,0)
      IF(NUMSP .EQ. 0) THEN
C          NO CSA TRUCK SO TRY ASP-ATP TRUCK
C      **** IF HAVE LOOKED AT ASP QUEUE, THERE IS NO AMMO GO TO 142, TRUCK LOS
      IF(NATPQ.EQ.IQ(IATPSD(3),NATP)) GO TO 142
      NATPQ=IQ(IATPSD(3),NATP)
      GO TO 130
C      **** WRITE FLAG
142 WRITE(6,141) NATP,NUMAM,TIME
141 FORMAT(' ATP ',I2,' S&PS BUSY FOR AMMO ',I2,' AT TIME ',F8.2)
      WRITE(LUOUT,141) NATP,NUMAM,TIME
      IPARM(1) = ITRUCK(NUMTK,4)
      IPARM(2) = NUMTK
      IPARM(3) = NATP
      IPARM(4) = 999
      ETIME = TIME
      IF(ITRUCK(NUMTK,6) .LE. 0) ETIME = ETIME + 15.
      CALL SCHED(4,IPARM,ETIME + TDELAY + TTDEL)
      IF(NUMAM .EQ. 10) THEN
          IIPARM(1) = NATP
          IIPARM(2) = -1
          @ ATPARV

```

```

        CALL SCHED(3,IIPARM,ETIME + TDELAY)
        RETURN
    END IF
    ITRUCK(NUMSVR,3) = 4
    CALL OPERA(NUMSVR,TLOAD,TFAIL)
    DTIME = TIME + TLOAD + TFAIL
    IPARM(1) = NATP
    IPARM(2) = NUMSVR
    IPARM(3) = 0
    IPARM(4) = 0
    CALL SCHED(3,IPARM,DTIME)
    RETURN
END IF
C FIND THE NUMBER OF ROUNDS ON NUMSP. IF SUFFICIENT, DECREMENT
C AMMO, SCHEDULE UNTARV, PUT S & P BACK IN ASP Q.
C IF INSUFFICIENT EMPTY CSA S & P, SEND TO CSA, DECREMENT
C THE NUMBER OF ROUNDS REQUIRED, FIND ANOTHER S & P WITH
C THE PROPER AMMO
C UPDATE PER CENT ROUNDS ON THE S & P
MIXX = ITRUCK(NUMSP,5)
NRONSP = (IMIX(MIXX,NUMAM)*ITRUCK(NUMSP,6)+9999)/10000
WRITE(LUOUT,300)MIX,MIXX,NUMAM,NRNSDN,NUMTK,NRONSP,NUMSP,NATPG
300 FORMAT(' IATP ',8I6)
C IF INSUFFICIENT ROUNDS GO TO 150
IF(NRNSDN .GT. NRONSP)GO TO 150
C SUFFICIENT AMMO ON S & P. DECREMENT AMMO ON S & P.
C IF ARTY AMMO GO LOAD POWDER
NRND = IMIX(MIX,NUMAM)
IF(NUMAM.GT.2.AND.NUMAM.LE.7)CALL LDPWDR(NRND,IPARM,NUMAM)
ITRUCK(NUMSP,6) = 10000 * (NRONSP - NRNSDN) / IMIX(MIXX,NUMAM)
C SCHEDULE RELEASE OF S & P (SCHED ATPAR1 OR 2)
IF(IFLAG.EQ.1)THEN
    TDELAY = TDELAY + IMIX(MIX,31) * (NRNSDN/FRNN)
ELSE
    IF(NUMAM .GT. 3 .AND. NUMAM .LE. 7)THEN
        TDELAY = (IMIX(MIX,31)/2)
    ELSE
        TDELAY = IMIX(MIX,31)
    END IF
END IF
135 IIPARM(1) = NATP
IIPARM(2) = NUMSP
IIPARM(4) = 555
IF(IDAY .EQ. 0)TDELAY = 1.54 * TDELAY
TOTIM = TIME + TDELAY
IF(NUMSP .GE. IATPSD(1))THEN
    CALL SCHED(11,IIPARM,TOTIM)
ELSE
    CALL SCHED(10,IIPARM,TOTIM)
END IF
C GO TO SCHEDULE UNTARV
GO TO 200
C*** INSUFFICIENT AMMO OR EXACTLY ENOUGH AMMO ON S & P
C TIME TO SHIFT PARTIAL LOAD

```

```

TPAR = IMIX(MIX,31) * (NRONSP / FRNN) + 5.0
NRNDSN = NRNDSN - NRONSP
ITRUCK(NUMSP,6) = 0
ITRUCK(NUMTK,6) = (NRONSP/FRNN) * 10000
C SCHEDULE RELEASE OF S & P (SCHED ATPAR1 OR 2)
  IF(IFLAG.EQ. 1)THEN
    TDELAY = TDELAY + TPAR
    GO TO 151
  ELSE
    TDELAY = TPAR
  END IF
151 IIPARM(1) = NATP
    IIPARM(2) = NUMSP
    IIPARM(4) = 555
    IF(IDAY.EQ. 0)TDELAY = 1.54 * TDELAY
    TOTIM = TIME + TDELAY
    IF(NUMSP.GE. IATPSD(1))THEN
      CALL SCHED(11,IIPARM,TOTIM) @ ATPAR2
    ELSE
      CALL SCHED(10,IIPARM,TOTIM) @ ATPAR1
    END IF
C**** IF EXACTLY ENOUGH ROUNDS ON TRUCK,SEND TRUCK BACK TO UNIT
  IF(NRNDSN.EQ.0) GO TO 200
C GO GET ANOTHER ASP-ATP TRUCK TO COMPLETE THE LOAD
  IFLAG = 1
  GO TO 130
C HAVE SUFFICIENT AMMO, SCHEDULE UNTARV AND NEXT ATP DECREMENT
200 IATP(NATP,IQTYPE+13) = IATP(NATP,IQTYPE+13) - 1
  IF(IATP(NATP,IQTYPE+13).LE. 0)IATP(NATP,IQTYPE+13)=0
C DECREMENT AMMO AND AMMO DEMAND
  IATP(NATP,NUMAM*3+19) = IATP(NATP,NUMAM*3+19)
Z - IMIX(MIX,NUMAM)
  IATP(NATP,NUMAM*3+19) = IATP(NATP,NUMAM*3+19) -
Z IMIX(MIX,NUMAM)
  IF(NUMAM.GT. 2 .AND. NUMAM.LE. 5)THEN
    IATP(NATP,27) = IATP(NATP,27) - IMIX(MIX,NUMAM)
    IATP(NATP,28) = IATP(NATP,28) - IMIX(MIX,NUMAM)
    IATP(NATP,51) = IATP(NATP,51) - IMIX(MIX,NUMAM)
  ELSE
    IF(NUMAM.GT. 5 .AND. NUMAM.LE. 9)THEN
      IATP(NATP,42) = IATP(NATP,42) - IMIX(MIX,NUMAM)
      IATP(NATP,43) = IATP(NATP,43) - IMIX(MIX,NUMAM)
      IATP(NATP,51) = IATP(NATP,51) - IMIX(MIX,NUMAM)
    END IF
  END IF
C**** CHARGE AMMO ISSUED
  MIXIND = MIX @ 10TON LOAD
  IF(ITRUCK(NUMTK,1).NE. 1)MIXIND = MIX - 20 @ 5TON OR 10TON MLRS
  IF(ITRUCK(NUMTK,5).EQ. 10)MIXIND = MIX @ MLRS 10TON W/12 TLR
  IATPAM(NATP,MIXIND)=IATPAM(NATP,MIXIND)+1
  IF(IFLAG.EQ. 1)THEN
    TLOAD = TDELAY
  ELSE
    TLOAD=IMIX(MIX,31)

```

```

      IF (IDAY.EQ.0) TLOAD = 1.54*TLOAD
C**** CHECK IF MLRS AND SCHEDULE DEPARTURE
      IF (NUMAM .EQ. 10) THEN
        DTIME = TIME + TLOAD
        IIPARM(1) = NATP
        IIPARM(2) = -1
        IIPARM(4) = 0
        CALL SCHED(8,IIPARM,DTIME)
        CONTINUE
      END IF
C      SET THE VALUE FOR ARTY OR MANUEVER QUEUE
      KVAL = 4
      IF(IQTYPE .EQ. 2) KVAL = 1
C      ADD ONE TO THE NUMBER SERVED BY THIS QUEUE
      JATP(NATP,KVAL) = JATP(NATP,KVAL) + 1
      IF(NUMAM .EQ. 10)GO TO 158
C**** SCHEDULE THE AVAILABILITY OF THE SERVER
      ITRUCK(NUMSVR,3) = 4
      CALL OPERA(NUMSVR,TLOAD,TFAIL)
      DTIME = TIME + TLOAD + TFAIL
      IPARM(1) = NATP
      IPARM(2) = NUMSVR
      CALL SCHED(8,IPARM,DTIME)
      ITRUCK(NUMSVR,6) = ITRUCK(NUMSVR,6) + TLOAD
      IF(ITRUCK(NUMTK,1) .EQ.1) THEN
        ITRUCK(NUMSVR,12) = ITRUCK(NUMSVR,12) + 1 @ 10 TON CNTR
      ELSE
        ITRUCK(NUMSVR,11) = ITRUCK(NUMSVR,11) + 1 @ 5 TON CNTR
      END IF
C
C      SCHEDULE UNTARV OF SUPPLY TRUCK
158 ITRUCK(NUMTK,3) = 4
      CALL INTRDK(NUMTK,THIND)
      NUNIT = ITRUCK(NUMTK,4)
      IPARM(1) = NUNIT
      IPARM(2) = NUMTK
      NTYP = IUNIT(NUNIT,1)
C **** IF SUPPLY TRUCK INTERDICTED, SCHEDULE ASPARV
      IF(THIND .GT. 0)THEN
        ITRUCK(NUMTK,6) = 0
        IPARM(3) = IUNIT(NUNIT,3)
        IPARM(4) = ITRUCK(NUMTK,5)
        TOTTIM = TIME + THIND + TLOAD
        CALL SCHED(5,IPARM,TOTTIM)
        JUNIT(NTYP,14) = JUNIT(NTYP,14) + 1
        RETURN
      END IF
      DIST = IUNIT(NUNIT,4)
      ITKTYP = ITRUCK(NUMTK,1)
      TVLTIM = 60. * DIST / ITYPE(ITKTYP,IDAY+1)
      CALL OPERA(NUMTK,TVLTIM,TFAIL)
      JUNIT(NTYP,13) = JUNIT(NTYP,13) + 1
C      ADD TO THE NUMBER KILLED AND FAILED THIS TRIP
      IF(TFAIL .GT. 0) JUNIT(NTYP,15) = JUNIT(NTYP,15) + 1

```

```
JUNIT(NTYP,16) = JUNIT(NTYP,16) + NTQUE + TLOAD + TULTIM  
TOTTIM = TIME + TULTIM + TFAIL + TLOAD  
CALL SCHED(9,IPARM,TOTTIM)  
ITRUCK(NUMTK,6) = 10000
```

@ UNTARV

C

```
RETURN  
END
```

```

h.      SUBROUTINE ATPARV
        SUBROUTINE ATPARV (IPARM)
C**** EVENT ATPARV -- ARRIVAL OF UNIT TRUCK AT ATP
C                      FIND FORKLIFT OR CRAN, AND A S&P (SCHED ATP)
C      EVENT TYPE 4
C      CALLED BY MAINARM
C      CALLS      SCHED, IQ, FINTK, PUTQUE, GETQUE, OPERA, INTRDK
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- ATP NUMBER
C**** IPARM(4) -- MIX OR 999 = RETURN OF PARTIALLY LOADED UNIT TRUCK
C
C**** SCHEDULES   -- ASPARV, ARRIVAL OF UNIT TRUCK AT ASP
C                      (IF AMMO IS NOT CURRENTLY ON HAND FOR ALL
C                      TRUCKS IN QUEUE)
C                      -- ATP, SERVICE OF UNIT TRUCK FROM QUEUE AT ATP
C                      -- ATPARV, RESCHEDULE 30 MIN LATER
C**** SCHEDULED BY ATP, ATPARV, UNTDEP
C                      (IF ATP SERVICE WAS IDLE FOR THIS QUEUE)
C
C**** DATA REQUIRED -- AMMO REQUIRED BY TRUCKS IN QUEUE.
C
        INCLUDE LOG,LIST
        DIMENSION IPARM(5)
C      LOCAL VARIABLES DEFINED
C      NUMQ - ATP QUEUE FOR ARTY OR ROUTINE SERVICE
C      NUMQS - SERVER QUEUE
C      MIX - INDEX OF AMMO MIX USED TO ACCESS IMIX.
C      NEEDTK - NUMBER OF ROUNDS NEEDED TYPE I BY UNIT TRUCK.
C      INDEX - INDEX COMPUTED FOR AMMO TYPE I TO ACCESS
C              ONHAND AND WANTED BY TRUCK IN QUEUE.
C      JONHAND - AMOUNT OF AMMO TYPE I PRESENTLY ON HAND AT ATP
C      NUMOLD - CHECK TRK FOR SERVER
C      NEEDOT - AMOUNT OF AMMO I NEEDED BY OTHER TRUCKS IN QUEUE.
C      MANART - FLAG SET TO 2 IF MANEUVER AMMO, 1 IF ARTY AMMO
C      DIST - DIST FROM ASP TO ATP.
C      TVLTIM - UNOPPOSED TRAVEL TIME.
C      TFAIL - TRAVEL TIME INCREMENT DUE TO MECHANICAL FAILURE
C      TMIND - TRAVEL TIME INCREMENT DUE TO INTERDICTION
C      NFAOMD - TOTAL RDS NEEDED BY ALL ARTY TRKS
C      TOTIM - TIME OF TRUCK ARRIVAL AT ASP
C
C      NFLAG = 0                      @ S&P FLAG
C      NUNIT = IPARM(1)
C      NUMTK = IPARM(2)
C      NATP  = IPARM(3)
C      IPART = IPARM(4)
C
C      DETERMINE AMMO MIX WANTED BY THE TRUCK.
C      MIX = ITRUCK(NUMTK,5)
C *** IF ATP ASSIGNMENT HAS CHANGED WHILE UNIT TRUCK IS ENROUTE.

```

```

LATP = IUNIT(NUNIT,2)
IF(NATP .NE. LATP)THEN
  IPARM(3) = LATP
  CALL SCHED(4,IPARM,TIME + 30.)
  RETURN
END IF
IF(MIX.LE.0) THEN
  WRITE(6,10) NUMTK
10  FORMAT(' ATPARV -- ZERO MIX ON TRUCK ',I4)
  RETURN
END IF
C  SET ITRUCK(N,3) EQUAL TO THE TIME OF QUEUE ENTRY. THIS
C  ALLOW QUEUE WAIT TIME TO BE FOUND FOR THE TRUCK
ITRUCK(NUMTK,3) = TIME
C  FIND THE UNIT TYPE
JTYPE = IUNIT(NUNIT,1)
IF(IUNIT(NUNIT,2).EQ.0)GO TO 400
NUMAM = MIX
IF(MIX .GT. LPPAR(7))NUMAM = MIX - LPPAR(7)
C  DETERMINE QUANTITY NEEDED ON THIS TRUCK (NEEDTK)
NEEDTK = IMIX(MIX,NUMAM)-(ITRUCK(NUMTK,6)*IMIX(MIX,NUMAM)+
Z  9999/10000)
C  ASSUME MANEUVER AMMO.
MANART = 2
C  IF ARTY RESET MANART
IF(NUMAM .GT. LPPAR(3) .AND. NUMAM .NE. 9)MANART = 1
IF(IPART .EQ. 999)GO TO 25
C  HOW MANY ROUNDS ARE NEEDED BY THE OTHER TRUCKS IN THE QUEUE
INDEX = NUMAM * 3 + 19
NEEDOT = IATP(NATP,INDEX)
C  HOW MANY ROUNDS I ARE AT ATP - JONHND
JONHND = IATP(NATP,INDEX - 1)
C  IF INSUFFICIENT ON HAND GO TO 400
IF(JONHND .LT. NEEDOT + NEEDTK)GO TO 400
C**** CHECK IF A G & P IS AVAILABLE FOR RELOADING
25 NATPQ = IQ(IATPSD(2),NATP)
30  CALL FINTK(NATPQ,NUMAM,NUMSP,0)
IF(NUMSP .NE. 0)GO TO 40
IF(NATPQ .NE. IQ(IATPSD(3),NATP))THEN
  NATPQ = IQ(IATPSD(3),NATP)
  GO TO 30
ELSE
  IF(IPART .EQ. 999)GO TO 200
  IF(NUMAM .GE. 4 .AND. NUMAM .LE.7)THEN
    GO TO 50
  ELSE
    GO TO 60
  END IF
END IF
C  HAVE FOUND A TRUCK, PUT BACK IN QUEUE
40  CALL PUTQUE(NUMSP,NATPQ)
NFLAG = 1
IF(IPART .EQ. 999)GO TO 80
C  IF NOT ARTY GO TO 60

```

```

      IF(MANART.EQ. 1 .AND. NUMAM.EQ. 10) GO TO 60
C     HAVE ARTY IS THERE SUFFICIENT POWDER
C     HOW MANY RDS ARE NEEDED BY ALL ARTY TRKS IN QUEUE(SUFFICIENT PDS?)
50  IF (NUMAM.LE. 5) THEN
      NFADMD = IATP(NATP,31) + IATP(NATP,34)
      IF(IATP(NATP,27) - IATP(NATP,28) .GE. NFADMD + NEEDTK) THEN
          GO TO 60
      ELSE
          GO TO 400
      END IF
    ELSE
      NFADMD = IATP(NATP,37) + IATP(NATP,40)
      IF(IATP(NATP,42) - IATP(NATP,43) .GE. NFADMD + NEEDTK) THEN
          GO TO 60
      ELSE
          GO TO 400
      END IF
    END IF
60  CONTINUE
C     ADD TO QUEUE DEMAND FOR AMMO TYPE
      IF(IPART.EQ. 999)GO TO 80
      INDEX = NUMAM * 3 + 19
      IATP(NATP,INDEX) = IATP(NATP,INDEX) + IMIX(MIX,NUMAM)
C**** IF ARTY ADD TO POWDER , IF NOT GO TO 70
      IF(MANART.EQ. 1 .AND. NUMAM.GT. 9) GO TO 70
      IF(MANART.EQ.2) GO TO 70
      IF(NUMAM.GT. LPPAR(3) .AND. NUMAM.LE. 5)THEN
          IATP(NATP,28)=IATP(NATP,28)+IMIX(MIX,NUMAM)
      ELSE IF(NUMAM.GT. 5 .AND. NUMAM.LT. 9)THEN
          IATP(NATP,43)=IATP(NATP,43)+IMIX(MIX,NUMAM)
      END IF
70  CONTINUE
C     INCREMENT NUMBER OF TRUCKS IN THE QUEUE
      IATP(NATP,MANART+13) = IATP(NATP,MANART+13) + 1
      IF(NFLAG.EQ. 0)GO TO 200 @ NO SIF
C**** IF MLRS TRUCK SCHED ATP EVENT NOW
C     IF THERE ARE LESS THAN 3 TRUCKS ALREADY LOADING
80  IF (NUMAM.EQ. 10) THEN
      IF(IATP(NATP,8) .LT. 3) THEN
          IATP(NATP,8) = IATP(NATP,8) + 1
          IPARM(1) = 1
          IPARM(2) = NATP
          IPARM(3) = NUMTK
          IPARM(4) = NUMTK
          CALL SCHED(6,IPARM,TIME) @ ATP
          RETURN
      ELSE
          ITRUCK(NUMTK,15) = ITRUCK(NUMTK,15) + 1 @ MLRS WAITS FOR 3
          GO TO 200
      END IF
    END IF
C**** FIND THE QUEUE FOR SERVERS
      NUMQS = IATP(NATP,10)
C     GET A SERVER FROM THIS QUEUE

```

```

NUMOLD = NUMSRV
C IF NO SERVER PUT THE TRUCK IN ITS PROPER QUEUE
IF (NUMSRV .EQ. 0) THEN
    IATP(NATP,20) = IATP(NATP,20) + 1 @ 'NO SERVER' CNTR
    GO TO 200
END IF
C HAVE A SERVER PROCESS THE MANUEVER OR ARTY QUEUE AS DIRECTED
IF (MANART .EQ. 2) THEN
100 IF (ITRUCK(NUMSRV,1) .NE. 9) GO TO 110
C**** HAVE A FORKLIFT FOR MANUEVER TRUCK, SCHEDULE ATP EVENT
    IPARM(1) = 2
    IPARM(2) = NATP
    IPARM(3) = NUMTK
    IPARM(4) = NUMSRV -
    CALL SCHED(6,IPARM,TIME) 3 ATP
    RETURN
C
C**** NOT A FORKLIFT, TRY AND FIND ONE
110 CALL PUTQUE(NUMSRV,NUMQS)
C GET A NEW SERVER
    CALL GETQUE(NUMSRV,NUMQS)
    IF (NUMSRV .NE. NUMOLD) GO TO 100
ELSE
C**** PROCESSING AN ARTY TRUCK
120 IF (ITRUCK(NUMSRV,1) .NE. 9) GO TO 130
C**** HAVE A CRANE WITH THE ARTY TRUCK SCHEDULE ATP EVENT
    IPARM(1) = 1
    IPARM(2) = NATP
    IPARM(3) = NUMTK
    IPARM(4) = NUMSRV
    CALL SCHED(6,IPARM,TIME) 3 ATP
    RETURN
C NOT A CRANE, TRY AND FIND ONE
130 CALL PUTQUE(NUMSRV,NUMQS)
C GET A NEW SERVER
    CALL GETQUE(NUMSRV,NUMQS)
    IF (NUMOLD .NE. NUMSRV) GO TO 120
C CANNOT FIND A CRANE, USE THE FORKLIFT
    IPARM(1) = 1
    IPARM(2) = NATP
    IPARM(3) = NUMTK
    IPARM(4) = NUMSRV
    CALL SCHED(6,IPARM,TIME) 3 ATP
    RETURN
C
END IF
C FIND QUEUE NUMBER - NUMQUE
200 NUMQ = IATP(NATP,MANART +10)
    CALL PUTQUE(NUMTK,NUMQ)
    RETURN
C INSUFFICIENT AMMO SEND TO ASP
C FIND DIST TO ASP
400 DIST = IUNIT(NUNIT,5)-IUNIT(NUNIT,4)
    ITXTYP = ITRUCK(NUMTK,1)

```

```

C   CHANGE TRUCK STATUS CODE
    ITRUCK(NUMTK,3) = 5
C   COMPUTE INTERDICTION DELAY - TMIND
    CALL INTRDK(NUMTK,TMIND)
    IF(TMIND .LE. 0.) THEN
C       COMPUTE DELAY DUE TO FAILURE - TFAIL
        CALL OPERA(NUMTK,TULTIM,TFAIL)
    ELSE
        TFAIL = 0.
    END IF
C   COMPUTE ASP ARRIVAL TIME - TOTIM
    TOTIM = TIME +TULTIM + TFAIL + TMIND
C**** RECORD NO. OF TRUCKS BUMPED TO ASP
    INDEX = NUMAM + 20 @ 10TON BUMPED
    IF(ITRUCK(NUMTK,1) .NE. 1) INDEX = NUMAM + 30 @ 5T OR 10T MLRS
    IF(ITRUCK(NUMTK,5).EQ.10)INDEX = NUMAM+20 @ MLRS 10TON W/ 12 TLR
    IATPAH(NATP,INDEX)= IATPAH(NATP,INDEX) + 1
    IPARM(3) = IUNIT(NUNIT,3)
C   TRUCK IS BEING BUMPED TO ASP, ADD ONE TO THE NUMBER SENT
    JUNIT(JTYP,9) = JUNIT(JTYP,9) + 1
C   ADD TO THE INTERDICTION AND FAILURE COUNTERS FOR THIS MOVE
    IF(TMIND .GT. 0)JUNIT(JTYP,10) = JUNIT(JTYP,10) + 1
    IF(TFAIL .GT. 0) JUNIT(JTYP,11) = JUNIT(JTYP,11) + 1
C   ADD THE TRAVEL TIME TO THE CUMULATIVE TRAVEL TIME FOR THIS MOVE
    JUNIT(JTYP,12) = JUNIT(JTYP,12) + TULTIM
    IPARM(4)=ITRUCK(NUMTK,5)
    CALL SCHED(5,IPARM,TOTIM)
    ITRUCK(NUMTK,14) = ITRUCK(NUMTK,14) + 1 @ ASPARU @ BUMP TO ASP CNTR
C
    RETURN
    END

```

```

i.      SUBROUTINE ATPAR1
        SUBROUTINE ATPAR1 (IPARM)
C**** EVENT ATPAR1 -- ARRIVAL OF S & P AT ATP FROM CSA
C              LOOKS FOR EMPTY TRAIL TO TAKE BACK TO CSA
C      EVENT TYPE 10
C      CALLED BY MAINARM
C      CALLS      OPERA, SCHED, IQ, PUTQUE, GETQUE, INTRNK
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- ATP NUMBER
C**** IPARM(2) -- S&P NUMBER
C**** IPARM(3) -- ASSOCIATED ASP NUMBER (MIX IF CSA TO CFA-ATP LINK)
C**** IPARM(4) -- 555 : RETURN FROM RELOAD  333: ARRIVAL AT DAD
C              MIX IF FROM CSADEP
C
C      SCHEDULES      ASPAR2, BUMP -> ASP IF ATP IS OVERSTOCKED
C              ATPAR1, RESCHEDULE IN 30 MIN OR DIVERT TO 2ND ATP
C              CSAARV, EMPTY TRUCK TO CSA
C      SCHEDULED BY  ATP, ATPAR1, CSADEP
C**** PUTS TRUCK IN CSA-ATP QUEUE
C
C**** CHANGES      -- ATP AMMO SUPPLY.
C      NOTE:      EACH TRACTOR HAS TWO TRAILERS!  FOR ATP S&PS THE TRACTOR
C      -----      IS ASSOCIATED WITH A FULL TRAILER (AT THE ASP THE TRACTOR
C              GOES WITH THE EMPTY TRAILER).
C
C      INCLUDE LOG,LIST
C      DIMENSIONN IPARM(5) ,IIPARM(5)
C      INTEGER AVAIL
C
C**** LOCAL VARIABLES
C**** MIX -- MIX OF AMMO INDEX CARRIED ON THE TRUCK FROM THE CSA
C**** NATPQ -- NUMBER OF QUEUE FOR CSA - ATP TRUCKS
C**** IND -- INDEX FOR CURRENT AMMO SUPPLY BASE ON IATP DEFINITION
C**** LATP -- ATP WITH LOWEST STOCK
C**** LEAST -- STOCK AT LATP
C
C      NATP = IPARM(1)
C      NUMSP = IPARM(2)
C      NASP = IATP(NATP,6) - 10
C      ISTAT = IPARM(4)
C
C      IF(NUMSP .EQ. 0)GO TO 10
C      IF(ISTAT .EQ. 555)GO TO 3
C**** FIND THE MIX ON THE TRUCK
C      MIX = ITRUCK(NUMSP, 5)
C      DETERMINE AMMO TYPE
C * * * NUMAM IS FOR IATP ARRAY == IAM IS FOR IMIX ARRAY
C * * * * THE ABOVE DIFFER ONLY FOR FUZES (MIX 80)
C      NUMAM = MIX - LPPAR(9)
C      IAM = NUMAM
C      IF(MIX .EQ. 80)NUMAM = 11
C      IF(MIX.LE. 0) THEN

```

```

2   FORMAT(' ATPAR1 -- ZERO MIX ON TRUCK ',I4)
   RETURN
END IF

C
   LATP = NATP
C   IATPSD(4) = CFA ATP +
   IF(ISTAT.NE. 333 .OR. NATP.EQ. IATPSD(4))GO TO 3
C*** DETERMINE ATP DESTINATION (FROM DAO)
70  IND = NUMAM * 3 + 18
C* * * AVAILABLE = ONHAND - DEMAND + ON-THE-WAY
   AVAIL = IATP(NATP,IND) - IATP(NATP,IND+1) + IATP(NATP,IND+2)
   LEAST = AVAIL
   IF (NUMAM.EQ. 9) GO TO 95
C *** IF AMMO AT PARENT IS LOW, SCHEDULE ARRIVAL TO IT
   IAMCK = IAMLVL(1,NUMAM) * .75
   IF(AVAIL.LT. IAMCK)GO TO 90
C *** FIND ATP WITH LEAST AMMO ON HAND
   IF(NATP.LE. 3)THEN
      JLOW = 1
      JHIGH = 3
   ELSE
      JLOW = 4
      JHIGH = 5
   END IF
   DO 90 J = JLOW,JHIGH
      AMAV = IATP(J,IND) - IATP(J,IND+1) + IATP(J,IND+2)
      IF(AMAV.GE. LEAST)GO TO 90
      LEAST = AMAV
      LATP = J
90  CONTINUE
   IF(.7 * AVAIL.LT. LEAST)LATP = NATP

C
C * * * IF .GT., ATPS OVERSTOCKED, SEND S&P TO ASSC ASP
95  IF(LEAST * 100.GT. IAMLVL(1,29) * IAMLVL(1,NUMAM))THEN
      IF(IASP(NASP,2).LT. 1) NASP = IASP(NASP,1) - 10
      DIST = ABS(IATP(NATP,3) - IASP(NASP,1))
      ITYP = ITRUCK(NUMSP,1)
      TRTM = 60. * DIST / ITYPE(ITYP,IDAY+3)
      CALL OPERA(NUMSP,TRTM,TFAIL)
      IIPARM(1) = NASP + 10
      IIPARM(2) = NUMSP
      IIPARM(3) = 0
      IIPARM(4) = 444
      CALL SCHED(13,IIPARM,TIME + TRTM + TFAIL)
      WRITE(6,24)NUMSP,MIX,NASP+10,TIME
      9 ASPARC
24  FORMAT(' ATP S&P',I5,' MIX',I3,' DIVERTED TO ASP',I3,
Z   ' AT TIME',F9.1)
      IASP(NASP,IND+2) = IASP(NASP,IND+2) + IMIX(MIX,IAM)
      RETURN
   END IF

C
   IF(NATP.NE. LATP)WRITE(6,25)NATP,NUMSP,MIX,LATP,TIME
26  FORMAT(' ATP',I3,' S&P',I5,' MIX',I3,' DIVERTED TO ATP',I3,
Z   ' AT TIME',F9.1)

```

```

    IPARM(1) = LATP
90 IPARM(4) = 0
    DIST = IATP(LATP,1) - IATP(LATP,3)
    ITYP = ITRUCK(NUMSP,1)
    TRTM = 60. * DIST / ITYPE(ITYP, IDAY+3)
    CALL OPERA(NUMSP, TRTM, TFAIL)
    TOTIM = TIME + TRTM + TFAIL
    CALL SCHED(10, IPARM, TOTIM)
    IATP(LATP, NUMAM*3+20) = IATP(LATP, NUMAM*3+20) + IMIX(MIX, IAM)
    RETURN
C**** PUT TRUCK IN CSA-ATP AMMO QUEUE
3 IF (TIME .GT. 1440. .AND. NATP .EQ. IATPSD(4) ) THEN
    NATP = 1
    LATP = 1
    GO TO 70
END IF
    NATPQ = IQ(2, NATP)
    MTSP = NUMSP
    IF(ITRUCK(NUMSP,6) .LE. 0 .AND. IATP(NATP,5) .GT. 0)
ZGO TO 30
    CALL PUTQUE(NUMSP, NATPQ)
    IF(ISTAT .EQ. 555) THEN
        ITRUCK(NUMSP,3) = TIME
        IF(ITRUCK(NUMSP,6) .GT. 0) ITRUCK(NUMSP,3) = 2
        RETURN
    END IF
C
C**** ADD AMMO TO THAT AVAILABLE - DECR ON-THE-WAY
    IND = NUMAM* 3 + 18
    IATP(NATP, IND) = IATP(NATP, IND) + (IMIX(MIX, IAM)
Z * ITRUCK(NUMSP,6) + 9999)/10000
    IATP(NATP, IND+2) = IATP(NATP, IND+2) - (IMIX(MIX, IAM)
Z * ITRUCK(NUMSP,6) + 9999)/10000
    IF(MIX. EQ. 80) GO TO 7
    IATPSP(NATP, NUMAM) = IATPSP(NATP, NUMAM) + 1
C
C**** UPDATE TRUCK STATUS TO THE CSA-ATP QUEUE
7 IF(ITRUCK(NUMSP,3) .GE. 4) IATP(NATP,5) = IATP(NATP,5) + 1
    ITRUCK(NUMSP, 3) = 2
    ITRUCK(NUMSP,10) = ITRUCK(NUMSP,10) + 1 @ # ARRIVALS FROM CSA
C**** CHECK QUE FOR EMPTY S & PS
C**** BRING FIRST S & P FROM QUEUE(MTSP)
10 IF(IATP(NATP,5) .LE. 0) GO TO 50
    NATPQ = IQ(2, NATP)
    CALL GETQUE(MTSP, NATPQ)
C
C**** SEARCH FOR EMPTY S & P, STORE NCHTK AND PUT IT BACK
C**** IN QUEUE.
    NCHTK = MTSP
    IF(NCHTK .EQ. 0) GO TO 50
    CALL PUTQUE(MTSP, NATPQ)
15 CALL GETQUE(MTSP, NATPQ)
C
C**** DETERMINE TRUCK LOAD

```

```

C      WRONG TRUCK, PUT BACK IN QUFUE
      CALL PUTQUE(MTSP,NATPQ)
C      IF LAST TRUCK,GO TO 50
      IF(MTSP.EQ.NCHTK)GO TO 50
      GO TO 15
C****      FOUND EMPTY TRUCK - SCHED CSA ARRIVAL
C      DETERMINE DISTANCE TO BE TRAVELED
      30 DIST = IATP(NATP,1)
      ITKTYP = ITRUCK(MTSP,1)
      TRTM = 60. * DIST/ITYPE(ITKTYP,IDAY+3)
      IF(ITRUCK(MTSP,3) .GT. 10)THEN
        NTIQ = TIME - ITRUCK(MTSP,3)
        ITRUCK(MTSP,13) = ITRUCK(MTSP,13) + NTIQ
      END IF
      ITRUCK(MTSP,3) = 4
C      INTERDICTION DELAY - TMIND
      CALL INTRDK(MTSP,TMIND)
      IF (TMIND .LE. 0.) THEN
C      COMPUTE DELAY DUE TO FAILURE - TFAIL
        CALL OPERA(MTSP,TRTM,TFAIL)
      ELSE
        TFAIL = 0.
      END IF
      TOTTIM = TRTM + TIME + TFAIL + TMIND + 20.
      IIPARM(1) = NATP
      IIPARM(2) = MTSP
      IIPARM(3) = IATP(NATP,6)
      IIPARM(4) = 0
C      ASSUME CSA - ATP TRUCK
      CALL SCHED(9,IIPARM,TOTTIM)
      IATP(NATP,5) = IATP(NATP,5) - 1
      RETURN
C      NO EMPTY TRUCK FOUND SCHED FALSE EVENT
      50 IPARM(2) = 0
      CALL SCHED(10,IPARM,TIME + 30.)
C
      RETURN
      END

```

3 CSAARV

3 ATPAR1

```

j.      SUBROUTINE ATPAR2
        SUBROUTINE ATPAR2 (IPARM)
C****  EVENT ATPAR2  --  ARRIVAL OF S&P AT ATP FROM ASP (OR EMPTY FROM
C                               RELOADING IN ATP).
C                               FULL, PUT IN QUEUE: EMPTY, SEND TO ASP
C      EVENT TYPE 11
C      CALLED BY MAINARM
C      CALLS      IQ, PUTQUE, OPERA, INTRDK, SCHED
C
C****  J. FOX      JAN 79
C
C****  IPARM(1) -- ATP NUMBER
C****  IPARM(2) -- TRUCK NUMBER
C****  IPARM(3) -- ASSOCIATED ASP NUMBER
C****  IPARM(4) -- 555 : MEANS RETURN FROM RELOAD
C                               OR MIX : (FROM ASPAR1) SERVICE OF ATP TO ASP S&P
C
C      SCHEDULES      ASPAR1, EMPTY S&P -> ASP
C      SCHEDULED BY  ASPAR1, ASPAR2, ATP
C
C****  PUTS TRUCK IN ASP-ATP QUEUE
C
C****  CHANGES      --  ATP AMMO SUPPLY.
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5),IIPARM(5)
C
C****  LOCAL VARIABLES
C****  MIX  -- MIX NUMBER OF AMMO CARRIED ON THE TRUCK
C****  NATPQ  -- NUMBER OF ATP QUEUE FOR LOADED AMMO TRUCKS
C****  IND  -- INDEX FOR CURRENT AMMO SUPPLY IN IATP.
C****  ISTAT  -- IPARM(4) MIX OR 555(RTN RELOAD)
C****  MTSP  -- TRUCK NUMBER OF AN EMPTY S&P -> ASPAR1
C
C      NATP = IPARM(1)
C      NUMSP = IPARM(2)
C      ISTAT = IPARM(4)
C
C      IF(ISTAT .EQ. 555)GO TO 3
C****  FIND MIX NUMBER ON TRUCK
C      MIX = ITRUCK(NUMSP, 5)
C      IF(MIX .LE. 0) THEN
C          WRITE(6,7) NUMSP
C      2   FORMAT(' ATPAR2 -- ZERO MIX ON TRUCK ',I4)
C          RETURN
C      END IF
C
C****  PUT TRUCK IN ATP AMMO TRUCK QUEUE
C      3   IF(ITRUCK(NUMSP,6) .LE. 0)THEN
C          MTSP = NUMSP
C          GO TO 30
C      END IF
C      NATPQ = IQ(3,NATP)
C      CALL PUTQUE (NUMSP, NATPQ)

```

```

C
C**** ADD AMMO TO AMMO AVIALABLE - DECR ON-THE-WAY
C * * * NUMAM IS FOR IATP ARRAY == IAM IS FOR IMIX ARRAY
C * * * * THE ABOVE DIFFER ONLY FOR FUZES (MIX 80)
    NUMAM = MIX - LPPAR(3)
    IAM = NUMAM
    IF(MIX .EQ. 80) NUMAM = 11
    IND = NUMAM * 3 + 18                @ ON HAND
    IATP(NATP,IND) = IATP(NATP,IND) + (IMIX(MIX,IAM) * ITRUCK
Z    (NUMSP,6) + 9999) / 10000
    IATP(NATP,IND+2) = IATP(NATP,IND+2) - (IMIX(MIX,IAM) * ITRUCK
Z    (NUMSP,6) + 9999) / 10000        @ ON-THE-WAY
    IATPSP(NATP,NUMAM+11) = IATPSP(NATP,NUMAM+11) + 1
C
C**** UPDATE TRUCK STATUS TO BEING IN THE ATP QUEUE
C    IF(ITRUCK(NUMSP,3).EQ.4) IATP(NATP,5)=IATP(NATP,5)+1
    ITRUCK(NUMSP, 3) = 2
    ITRUCK(NUMSP,10) = ITRUCK(NUMSP,10) + 1 @ # ARRIVALS FROM ASP
C
    RETURN
C
C****    HAVE FOUND EMPTY TRUCK - SCHED ASP ARRIVAL
C        DETERMINE DIST TO BE TRAVELED
    30 DIST = IATP(NATP,2)
    ITKTYP = ITRUCK(MTSP,1)
    TRTM = 60. * DIST / ITYPE(ITKTYP,IDAY+3)
    ITRUCK(MTSP,3) = 4
C        COMPUTE INTERDICTION DELAY - THIND
    CALL INTRDK(MTSP,THIND)
    IF(THIND .LE. 0.) THEN
C        COMPUTE DELAY DUE TO FAILURE - TFAIL
        CALL OPERA(MTSP, TRTM, TFAIL)
    ELSE
        TFAIL = 0.
    END IF
    TOTTIM = TRTM + TIME + TFAIL + THIND + 20.
    IIPARM(1) = NATP
    IIPARM(2) = MTSP
    IIPARM(3) = IATP(NATP,6)
C        ASSUME ASP - ATP TRUCK
    CALL SCHED(12,IIPARM,TOTTIM)                @ ASPAR1
C
    RETURN
END

```

```

k.      SUBROUTINE CONTRL
        SUBROUTINE CONTRL (TIME)
C      EVENT TYPE 18
C      CALLED BY MAINARM, INIT, SCHED
C      CALLS      EDITD, REPORT, SCHED, CREEVT
C
C***** ALLOWS INTERACTIVE CONTROL FOR DATA EDITING AND REPORTS
C***** ALLOWS SCHEDULING OF NEXT CONTRL TIME.
C***** H. JONES      FEB 79
C
C      SCHEDULES  CONTRL, ENDSIM
C      SCHEDULED BY CONTRL
        DIMENSION IPARM(5)
10      WRITE(6,20) TIME
15      FORMAT(' (1) - EDIT DATA ',/,
Z      ' (2) - WRITE REPORT ',/,
Z      ' (3) - SCHEDULE CONTROL ',/,
Z      ' (4) - RETURN ',/,
Z      ' (5) - STOP SIMULATION NOW ',/,
Z      ' (6) - EDIT TRUCK QUEUES ',/,
Z      ' (7) - CREATE EVENTS')
20      FORMAT(' TIME = ',F9.2,/, ' ?')
        READ(5,*,ERR=25) IOPT
        IF(IOPT .LT. 1 .OR. IOPT .GT. 7) GO TO 25
        GO TO (30, 40, 50, 70, 60, 65, 68), IOPT
25      WRITE(6,15)
        GO TO 10
C***** EDIT DATA
30      CALL EDITD
        GO TO 10
C
C***** WRITE REPORT
40      CALL REPORT (9)
        GO TO 10
C
C***** SCHEDULE CONTROL
50      WRITE(6,55)
55      FORMAT(' ENTER TIME FOR NEXT CONTROL  ')
        READ(5,*) TNEXT
        CALL SCHED (19, IPARM, TNEXT)      9 CONTRL
        GO TO 10
C
C***** EDIT TRUCK QUEUES
65      CALL TRKPUT
        GO TO 10
C
C***** CREATE EVENTS
68      CALL CREEVT
        GO TO 10
C
C***** STOP SIMULATION.
60      IPARM(1) = 9999
        IPARM(2) = 9999
        IPARM(3) = 9999

```

CALL SCHED (19, IPARM, TIME - .1)
70 RETURN
END

9 ENDSIM

```

1.  SUBROUTINE CREEVT
    SUBROUTINE CREEVT
C**** ENABLES INTERACTIVE CREATION OF EVENTS SUCH AS TRUCKS
C**** TO ARRIVE AT AN ATP FROM THE CSA IN MID-CI.
C
C      CALLED FROM ARM MENU(CTRL)
C      CALLS READF TO GET VALUES FROM THE KEYBOARD
C      SCHED TO SCHEDULE THE CREATED EVENT
C**** JAMES FOX ESQ. DDT. TNT. MARCH NINETEEN HUNDRED AND SEVENTY NINE
C
C**** LOCAL VARIABLE DEFINITION
C**** IPARM - CONTAINS THE 5 PARAMETERS OF THE EVENT
C**** INTGR - ' ' UP TO 6 INTEGER VALUES FROM THE CONSOLE
C**** IWORD - ' ' UP TO 6 ALPHA VALUES FROM THE CONSOLE
C**** REAL - ' ' UP TO 6 REAL VALUES FROM THE CONSOLE
C**** IEND - ' ' END OF INPUT CHECK
C**** TOTIM - TIME OF SCHEDULED EVENT
C**** ITYP - EVENT TYPE
C**** IPARM(1) - UNIT/ATP/ASP NO.
C**** IPARM(2) - TRUCK NO.
C
C      CHARACTER*10 IWORD,IHELP,IEND
C      DIMENSION IPARM(5),INTGR(6),IWORD(6),REAL(6)
C      DATA IHELP //'HELP'/
C      DATA IEND //'END'/
C
C      5 WRITE(6,100)
C      LU1 = 5
C      10 WRITE(6,150)
C      150 FORMAT(/,1X,' ? ')
C      CALL READF(LU1,6,INTGR,REAL,IWORD)
C**** IF END OF INPUT RETURN (200)
C      IF(IWORD(1).EQ.IEND) GO TO 200
C      IF(IWORD(1).EQ.IHELP) GO TO 5
C**** LOAD EVENT TYPE, PARAMETERS, AND TIME
C      ITYP = INTGR(1)
C      IF(ITYP.LE.0.OR.ITYP.GT.17) GO TO 10
C      DO 20 I = 1,5
C          IPARM(I) = INTGR(I+1)
C      20 CONTINUE
C      TOTIM = REAL(1)
C      CALL SCHED(ITYP,IPARM,TOTIM)
C      GO TO 10
C
C      100 FORMAT(' TO CREATE AN EVENT, INPUT AS A GROUP SEPARATED BY',/,
C      Z      ' COMMAS OR SPACES THE FOLLOWING 7 VALUES ',/,
C      Z      ' EVENT TYPE (INTEGER VALUES BETWEEN 1 AND 17)',/,
C      Z      ' 5 PARAMETERS FOR EACH EVENT(INTEGER,DEPEND ON EVENT TYPE',/,
C      Z      ' AND TIME (DECIMAL MINUTES, REAL)',/,
C      Z      ' EXAMPLE: 10,1,512,0,0,0,305.',/,
C      Z      ' CSA-TO-ATP TRUCK 512 WILL ARRIVE AT ATP AT TIME = 305 MIN')
C      200 RETURN
C      END

```

m. SUBROUTINE CSAARV

```

SUBROUTINE CSAARV (IPARM)
C**** EVENT CSAARV -- ARRIVAL OF S&P TRUCK AT CSA
C                      TRUCK IS SERVICED FROM CSA STOCK & SENT FWD
C          EVENT TYPE 9
C
C**** LARRY TOLIN      AUG 82
C
C**** IPARM(1) -- ATP NUMBER OR ASP NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- ASSOCIATED ASP NUMBER
C**** IPARM(4) -- '1' FLAG FOR INITIAL REPLENISHMENT TRIP
C                      (OTHERWISE 30 MIN. HITCH/UNHITCH TIME)
C                      -- '-1' FLAG TO PUT S&P INTO CSA QUEUE
C
C      CALLED BY MAINARM
C      CALLS      PUTQUE, SCHED, GETQUE
C
C**** SCHEDULES      -- CSADEP -- UNLESS STOCKAGE OBJECTIVE IS MET AT ASFS
C      SCHEDULED BY ASPAR2, ATPAR1
C
C**** CHANGES      -- CSA AMMO SUPPLY.
C      LOCAL VARIABLES
C          HITCH      -- TIME TO HITCH UP THE S&P
C          INITL      -- IPARM(4), INITIAL REPLEN TRIP, 1=YES
C      THESE VARS ARE USED TO FIND MIX TO GET:
C          AMCODE      -- LOOP TO FIND WORST NEEDED TYPE
C          AMPCT      -- LEAST % SO FAR
C          ASPPCT      -- % AT NUMASP
C          NLASP      -- ASP WITH ASPPCT
C          NUMASP      -- ASP THAT THE S&P WILL GO TO
C          PCT      -- PERCENT OF AMCODE AT ASP
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C      INTEGER AMCODE,ASP
C
C      INITL = IPARM(4)
C      IF(INITL.EQ. 1)THEN
C          HITCH = 0.
C      ELSE
C          HITCH = 30.
C      END IF
C
C      NUMSP = IPARM(2)
C      ITRUCK(NUMSP,3) = 8
C * * * * TO MAKE THE 'THO' DECISION PROCESS INTERACTIVE FOR
C      SCHEDULING CONVOYS TO EITHER ASFS AND/OR ATPS. CHANGE THE
C      SETTING OF 'IPARM(4)' TO '-1' IN ASPAR2 AND/OR ATPAR1,
C      RESPECTIVELY
C
C      THEN SCHEDULE S&PS (USING SSG ADDEVTARM) WITH 'IPARM(4)'
C      EQUAL TO '1'

```

```

      IF(INITL .EQ. -1)THEN
        CALL PUTQUE(NUMSP,176)
        RETURN
      END IF
C
      IF(ITRUCK(NUMSP,4) .GT. 95)GO TO 10
C
C * * * ATP S&P, SET CORRECT ATP # AND SCHEDULE TO CSADEP
      IF(ITRUCK(NUMSP,4) .NE. IATPSD(4) + 75)THEN @ (4) IS CFA ATP #
3      NATP = ITRUCK(NUMSP,4) - 75
        IF(NATP .NE. IPARM(1))THEN
          WRITE(6,5)NUMSP,NATP,IPARM(1),TIME
5          FORMAT(' ATP S&P',I5,' REASSIGNED TO PARENT ATP',I2,
$            ' FROM ASP/ATP',I3,' TIME',F8.1)
          END IF
          NASP = IATP(NATP,6)
          IPARM(1) = NATP
          IPARM(3) = MIX
          IPARM(4) = 0
          CALL SCHED(16,IPARM,TIME + HITCH)      @CSADEP
          MIXAM = ITRUCK(NUMSP,5) - LPPAR(8)
          ICSA(1,MIXAM) = ICSA(1,MIXAM) + 1 @ INCR CSA-ATP AMMO ISSUED
          RETURN
        END IF
C
C * * * CFA ATP S&PS -- PUT IN QUE
      IF(INITL .EQ. 1)GO TO 3
      CALL PUTQUE(NUMSP,176)
      ITRUCK(NUMSP,3) = TIME
      RETURN
C
C * * * ASP S&P, CHECK RECEIVING ASPS FOR GREATEST AMMO TYPE NEEDED
10 IF(INITL .EQ. 1)GO TO 35 @ INITIAL REPLENISHMENT FOR THIS S&P
C * * * LOOP THROUGH ALL RECEIVING ASPS (STATUS = 1)
C
      ASPPCT = 100.
      DO 20 AMCODE = 1,LPPAR(1)
        IF(IAMLVL(2,AMCODE) .EQ. 0)GO TO 20
        AMPCT = 100.
        DO 30 ASP = 1,10
          IF(IASP(ASP,2) .NE. 1)GO TO 30
C * *
          ASPPCT = (ONHAND - DEMAND + ON-THE-WAY)/STOCKAGE OBJECTIVE
          PCT = (IASP(ASP,AMCODE*3+18) - IASP(ASP,AMCODE*3+19) +
$            IASP(ASP,AMCODE*3+20))/IAMLVL(2,AMCODE)
          IF(PCT .LT. AMPCT)THEN
            AMPCT = PCT
            NLASP = ASP
          END IF
30      CONTINUE
          IF(AMPCT .LT. ASPPCT)THEN
            ASPPCT = AMPCT
            NUMASP = NLASP
            MIXAM = AMCODE
          END IF

```

```

C * * * IF ASPPCT .GT. ASP S 0, PUT S&P IN CSA QUEUE
    IASPCT = ASPPCT
    IF(IASPCT .GT. IAMLVL(1,30))THEN
        CALL PUTQUE(NUMSP,176)
        ITRUCK(NUMSP,3) = TIME
        ICSA(2,31) = ICSA(2,31) + 1      @ INCR CSA EMPTY S&P CNTR
        RETURN
    END IF

C
C * * * PREPARE S & P FOR CSADEP
    ITRUCK(NUMSP,4) = NUMASP + 125
    MIX = MIXAM + 60
    ITRUCK(NUMSP,5) = MIX
    ITRUCK(NUMSP,6) = 10000
    IPARM(1) = NUMASP + 10
    IPARM(3) = MIX
35 IPARM(4) = 0
    CALL SCHED(16,IPARM,TIME + HITCH)      @ CSADEP

C
C * * * INCR AMMO ISSUED AT CSA BY ONE S&P LOAD
    IF(INITL .EQ. 1)THEN
        MIXAM = ITRUCK(NUMSP,5) - LPPAR(8)
        NUMASP = IPARM(1) - 10
        MIX = ITRUCK(NUMSP,5)
    END IF
    ICSA(2,MIXAM) = ICSA(2,MIXAM) + 1

C * * * INCR ROUNDS ON-THE-WAY AT ASP
    IASP(NUMASP,MIXAM * 3 + 20) = IASP(NUMASP,MIXAM * 3 + 20) +
    $      IMIX(MIX,MIXAM)

C * * * CHECK QUEUE FOR EMPTY S&PS
    IF(ICSA(2,31) .LE. 0)RETURN
    CALL GETQUE(NUMSP,176)
    ICKSP = NUMSP
    CALL PUTQUE(NUMSP,176)
    CALL GETQUE(NUMSP,176)
    IF(ITRUCK(NUMSP,6) .EQ. 0)THEN
        ICSA(2,31) = ICSA(2,31) - 1      @ DECR CSA EMPTY S&P COUNTER
        GO TO 10
    ELSE
        CALL PUTQUE(NUMSP,176)      @ CSA S&P QUEUE
        IF(ICKSP .EQ. NUMSP)THEN
            ICSA(2,31) = 0
            RETURN
        END IF
    END IF
    STOP ' CSAARV '
    END

```

n. SUBROUTINE CSADEP

```

SUBROUTINE CSADEP (IPARM)
C      EVENT TYPE 16
C**** EVENT CSADEP -- DEPARTURE OF TRUCK FROM CSA
C      ATP USES CONVOYS OF 3 TRUCKS EACH, ASP USES CONVOYS OF 7 TRUCKS
C      EACH. TRUCKS ARE SCHEDULED 1 MINUTE APART.
C
C      CALLED BY MAINARM
C      CALLS      OPERA, INTRDK, SCHED, PUTQUE, GETQUE
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- ATP NUMBER OR ASP NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C**** IPARM(3) -- MIX
C**** IPARM(4) --
C
C**** SCHEDULES      -- ATPAR1, ARRIVAL OF TRUCK AT ATP
C                      -- ASPAR2, ARRIVAL OF TRUCK AT ASP
C
C      SCHEDULED BY CSAARV
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5),ISPQ(10)
C      INTEGER OWNER
C
C**** LOCAL VARIABLES :
C**** LDTIM -- TIME TO LOAD TRUCK
C**** DIST  -- DIST BACK TO ASP OR ATP
C**** TULTIM -- TRAVEL TIME
C**** ITKTYP -- TRUCK TYPE
C**** TS      -- (TIME) SPACE BETWEEN TRUCKS IN A CONVOY
C**** INCONV -- NUMBER OF TRUCKS IN A CONVOY
C**** NSPCK  -- CHECK TRUCK FOR S&P S
C**** TFAIL  -- DELAY ENROUTE DUE TO FAILURE
C**** TOTTIM -- TIME OF ARRIVAL OF TRUCK BACK TO ATP
C**** ISPQ() -- S&P S AWAITING CONVOY
C**** TMIND  -- INTERDICTION TIME DELAY
C
C      NUMSP = IPARM(2)
C      ITRUCK(NUMSP,3) = TIME
C      IF(IPARM(1) .GT.10)GO TO 100
C ***** ATP S&P *****
C
C      NATP = IPARM(1)
C
C      IF(NATP .EQ. IATPSD(4))THEN @ CFA ATP
C          MIX = ITRUCK(NUMSP,5)
C          NUMAM = MIX - LPPAR(3)
C          IATP(NATP,NUMAM * 3 + 20) = IATP(NATP,NUMAM * 3 + 20)
C          + IMIX(MIX,NUMAM) @ ON-THE-WAY TO CFA ATP
C          DIST = IATP(NATP,1)
C          ITKTYP = ITRUCK(NUMSP,1)

```

```

        ITRUCK(NUMSP,3) = 4
        CALL INTRDK(NUMSP,TMIND)
        IF(TMIND.EQ.0.)CALL OPERA(NUMSP,TULTIM,TFAIL)
        TOTIM = TIME + TULTIM + TFAIL + TMIND
        IPARM(4) = ITRUCK(NUMSP,5)
        CALL SCHED(10,IPARM,TOTIM)          @ ATPARI
        RETURN
    END IF

C
C * * * INCREMENT ATP S&P COUNTER
C
        CALL PUTQUE(NUMSP,176)
        IATP(NATP,13) = IATP(NATP,13) + 1
        IF(IATP(NATP,13) .LT. 3)RETURN

C
C * * * THREE S&PS AVAILABLE FOR CONVOY
C
        IATP(NATP,13) = IATP(NATP,13) - 3 @ DCRMT ATP S&P CONVOY COUNTER
        IATP(NATP,17) = IATP(NATP,17) + 1 @ INCR # OF CONVOYS CNTR

C
        DIST = IATP(NATP,3)
        INCONV = 3
        IEVTYP = 10
        IPARM(1) = NATP
        IPARM(4) = 333
        IF(NATP.EQ.IATPSD(4))IPARM(4) = 0
        OWNER = NATP + 75
        GO TO 20

C
C ***** ASP S&P *****
C
C * * * INCR ASP S&P CONVOY CNTR
C
        100 NASP = IPARM(1) - 10
        CALL PUTQUE(NUMSP,176)
        IASP(NASP,14) = IASP(NASP,14) + 1
        IF(IASP(NASP,14) .LT. 7)RETURN
C * * * SEVEN S&PS AVAILABLE FOR CONVOY
C
        IASP(NASP,14) = IASP(NASP,14) - 7
        IASP(NASP,17) = IASP(NASP,17) + 1
        DIST = IASP(NASP,1)
        INCONV = 7
        IEVTYP = 13
        IPARM(1) = NASP + 10
        IPARM(4) = 0
        OWNER = NASP + 125

C
C * * MOVE S&PS FROM CSA QUEUE TO CONVOY (ISPQ) QUEUE
        20 DO 3 JJ = 1,10
            ISPQ(JJ) = 0
        3   CONTINUE
        CALL GETQUE(NUMSP,176)
        KOUNT=1

```

```

      NSPCK=NUMSP
      CALL PUTQUE(NSPCK,176)
13  CALL GETQUE(NUMSP,176)
      IF(ITRUCK(NUMSP,4).EQ.OWNER)THEN
          ISPO(KOUNT)=NUMSP
          IF(NUMSP.EQ.NSPCK)GO TO 11
          IF(KOUNT.EQ.INCONV)GO TO 11
          KOUNT=KOUNT + 1
          GO TO 13
      ELSE
          CALL PUTQUE(NUMSP,176)
          IF(NUMSP.EQ.NSPCK)GO TO 11
          GO TO 13
      END IF
C
C*****HAVE ALL TRAILERS,FIND DISTANCE AND SCHEDULE
11  ITKTYP = ITRUCK(NUMSP,1)
      TULTIM = 60.*DIST/ITYPE(ITKTYP,IDAY+3)
      TS = 1.
      DO 12 JJ = 1,KOUNT
          NUMSP = ISPO(JJ)
          NTIQ = TIME - ITRUCK(NUMSP,3)
          ITRUCK(NUMSP,12) = ITRUCK(NUMSP,12) + NTIQ
          ITRUCK(NUMSP,11) = ITRUCK(NUMSP,11) + 1      @ # TIMES AT CSA
          ITRUCK(NUMSP,3) = 4
          TFAIL = 0.
          CALL INTRDK(NUMSP,TMIND)
          IF(TMIND.EQ.0.)CALL OPERA(NUMSP,TULTIM,TFAIL)
          TOTIM = TULTIM + TIME + TFAIL + TMIND + TS
          IPARM(2) = NUMSP
          IPARM(3) = ITRUCK(NUMSP,5)
          CALL SCHED(IEVTYP,IPARM,TOTIM)              @ ATPAR1/ASPAR2
          TS=TS+1.
12  CONTINUE
      RETURN
      END

```

```

O.      SUBROUTINE DEMAND
        SUBROUTINE DEMAND (IPARM)
C**** EVENT DEMAND -- CHECKS AMMO DEMAND OF UNITS.  RFLOAD IF NECESSARY.
C      EVENT TYPE 1
C      CALLED BY MAINARM
C      CALLS      RDIEXO(SCHEDULES NEXT DEMAND), SCHED, OPERA
C
C**** D. HILLIS  JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C
C**** SCHEDULES -- RELOAD, RESUPPLY OF UNITS.
C                  HELARV, ARRIVAL OF HELICOPTER AT UNIT
C                  HASPAR, HELI RETURN (FAILED IN ROUTE)
C      SCHEDULED BY RDIEXO, RDJIFF(READS DEMAND FILE)
C
C**** LOCAL VARIABLE DEFINITIONS
C**** K - UNIT AMMO INDEX
C**** NFLAG - 0 RELOAD NOT SCHEDULED YET. 1 RELOAD ALREADY SCHEDULED
C**** IFLAG - 0 NORMAL MODE. 1 - 155 HE OR ICM AMMO BELOW ORL
C**** I - UNIT NUMBER
C**** IA - LOOP INDEX
C**** II - LOOP INDEX
C**** TVLTIM - ROAD TRAVEL TIME
C**** TFAIL - TIME LOST DUE TO REMEDIAL MAINTENANCE
C**** TOTIM - TIME TO SCHEDULE THE EVENT
C**** IRRL - ROUTINE RESUPPLY LEVEL FOR LIVE WPNS
C**** IBAM - BASIC AMMO LEVEL FOR LIVE WPNS
C**** IRGND - NO. RND5 ON GROUND AT FARP
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C
C      NUNIT = IPARM(1)
C      NASP = IUNIT(NUNIT,3) - 10
C
C      CALL RDIEXO(NUNIT)
C      INITIALIZE FLAGS AND COUNTERS
C      IFLAG = 0
C      NFLAG = 0
C**** SELECT AN AMMO TYPE
C
C      DO 100 KK = 1,LPPAR(6)
C      K = KK * 13 - 5
C      IF(IUNIT(NUNIT,K).EQ.0) GO TO 100
C      IBAM=IUNIT(NUNIT,K+1)*IUNIT(NUNIT,K+7)
C**** CHECK FOR A FARP
C      IF(IUNIT(NUNIT,1).EQ.8) GO TO 50
C      IF(IBAM-IUNIT(NUNIT,K+4).EQ.0) GO TO 100
C**** CHECK FOR ROUTINE RESUPPLY
C      IRRL=IUNIT(NUNIT,K+1)*IUNIT(NUNIT,K+5)
C      IF(IUNIT(NUNIT,K+4).GT.IRRL) GO TO 100
C      L = IUNIT(NUNIT,K)
C

```

```

      IF(IUNIT(NUNIT,1).EQ.4) THEN
C**** CHECK FOR AMMO TYPES 4 AND 5
      IF(IUNIT(NUNIT,K) .EQ. 4 .OR. IUNIT(NUNIT,K) .EQ. 5) THEN
C**** CHECK TO SEE IF CURRENT AMMO SUPPLY GT CRITICAL RESUP LEVEL
      IF(IUNIT(NUNIT,K+4).GT.IUNIT(NUNIT,K+6)*IUNIT(NUNIT,K+1))GOTO35
      IFLAG = 1
      END IF
      END IF
C
C**** IS THERE AMMO OF THIS TYPE ON TRUCKS
      35 IF(IUNIT(NUNIT,K+8) .GT. 0) THEN
C      THERE IS AMMO ON A TRUCK
      IF(IFLAG .EQ. 1)GO TO 110
      IF(NFLAG .EQ. 1)GO TO 100
C      SCHEDULE RELOAD IMMEDIATELY
      CALL SCHED(2,IPARM,TIME)          @ RELOAD
      NFLAG = 1
      ELSE
      IF(IFLAG .EQ. 1)GO TO 150
      END IF
      GO TO 100
C**** DETERMINE AMMO REQUIREMENT AT FARP
      50 IRGND=(IUNIT(NUNIT,K+4)-IBAM) + IUNIT(NUNIT,K+3) @ CUR CAP + SHRT
      IF(IRGND .LE. 0)IRGND = 0
      IF(IUNIT(NUNIT,K+3).GT.IRGND) THEN
      IUNIT(NUNIT,K+3)=IUNIT(NUNIT,K+3)-IRGND
      IRGND = 0
      IUNIT(NUNIT,K+4)=IBAM-IUNIT(NUNIT,K+3)
      WRITE(LUOUT,210) IUNIT(NUNIT,K+4),IRGND
      GO TO 35
      END IF
      IRGND=IRGND-IUNIT(NUNIT,K+3)
      IUNIT(NUNIT,K+4)=IRGND+IBAM
      IUNIT(NUNIT,K+3)=0
      IUNIT(NUNIT,K+2) = 0
      WRITE(LUOUT,210) IUNIT(NUNIT,K+4),IRGND
      210 FORMAT(' DMD - FARP O/H= ',I5,' ON GRND= ',I5)
      GO TO 100
C**** COMPARE AVAILABLE AMMO AGAINST CRL
      110 IF(IUNIT(NUNIT,K+8)+IUNIT(NUNIT,K+4).GT.IUNIT(NUNIT,K+6)*
      Z IUNIT(NUNIT,K+1)) THEN
      IF(NFLAG .NE. 1) THEN
      CALL SCHED(2,IPARM,TIME)          @ RELOAD
      NFLAG = 1
      END IF
      IFLAG = 0
      GO TO 100
      END IF
      IF(NFLAG .EQ. 1)GO TO 150
      CALL SCHED(2,IPARM,TIME)          @ RELOAD
      NFLAG = 1
C**** HELICOPTER RESUPPLY LOGIC
C**** DOES UNIT ALREADY HAVE MAX NUMBER OF HELICOPTERS ASSIGNED
      150 IF(IUNIT(NUNIT,139) .EQ. 2)GO TO 170

```

```

        IF(IUNIT(NUNIT,138) .EQ. 1)GO TO 160
        WRITE(LUOUT,155)TIME
155  FORMAT(' AT ',F8.2,' MIN. NO HELICOPTERS AVAILABLE ')
C
        GO TO 170
160  WRITE(LUOUT,165)TIME
165  FORMAT(' AT ',F8.2,' MIN. HELI SCHEDULED, NO OTHERS AVAIL. ')
170  IF(NFLAG .EQ. 1)GO TO 200
        IFLAG = 0
        GO TO 100
180  LPPAR(5) = LPPAR(5) - 1
C      FIND AVAILABLE HELI(MISSION = 5, STATUS = 3)
        DO 185 II = 1,LPPAR(4)
            IF(ITRUCK(II,2) .NE. 5)GO TO 195
            IF(ITRUCK(II,3) .EQ. 6)GO TO 185
            IF(ITRUCK(II,3) .EQ. 3)GO TO 175
185  CONTINUE
        WRITE(LUOUT,186)
186  FORMAT(' CANNOT FIND THE AVAIL HELICOPTER-DEMAND ')
        GO TO 200
C      HAVE HELICOPTER II UPDATE STATUS
175  ITRUCK(II,3) = 4
        ITRUCK(II,9) = ITRUCK(II,9) + 1 @ HELI TRIP COUNTER
C      SCHEDULE ARRIVAL AT UNIT
        IPARM(2) = II
C      FIND TRAVEL TIME TVLTIM
        TVLTIM = 60. * IUNIT(NUNIT,5) / ITYPE(6,IDAY+1)
        CALL OPERA(II,TVLTIM,TFAIL)
        MIX=ITRUCK(II,5)
        TOTIM = TIME + TVLTIM + TFAIL + IMIX(MIX,32)
C *** INCREMENT ASP AMMO ISSUED BY ONE HELI LOAD
        IASPAM(NASP,90) = IASPAM(NASP,90) + 1
C *** DECREMENT AMMO ON HAND AT ASP
        DO 187 IA = 1,LPPAR(1)
            IASP(NASP,IA * 3 + 18) = IASP(NASP,IA * 3 + 18) - IMIX(MIX,IA)
187  CONTINUE
C**** IF HELICOPTER FAILS IN ROUTE TO UNIT
C      SEND ANOTHER HELICOPTER, IF AVAILABLE
C      SCHED HELASP
C      SET STATUS AS DOWN
        IF(TFAIL .LE. 0)GO TO 198
        ITRUCK(II,3)=6
        CALL SCHED(15,IPARM,TOTIM) @ HASPAR
        GO TO 190
198  IUNIT(NUNIT,138) = IUNIT(NUNIT,138) + 1
C***
        CALL SCHED(14,IPARM,TOTIM) @ HELARV
        IASP(NASP,15) = IASP(NASP,15) + 1
        IASP(NASP,3) = IASP(NASP,3) + 1
        IF(IUNIT(NUNIT,138) .EQ. 2)GO TO 170
C**** MIX 91 IS FOR THE CH47 HELICOPTER
        IF(IMIX(91,L)+IUNIT(NUNIT,K+4).GT.IUNIT(NUNIT,K+6)*IUNIT(NUNIT,
$      K+1)) GO TO 170
        GO TO 190

```

200 RETURN
END

p.

```

SUBROUTINE DEPASP
  SUBROUTINE DEPASP(IPARM)
C**** ROUTINE DEPASP REMOVES A MLRS TRUCK FROM SERVICE AT THE ASP. IT
C**** CHECKS FOR A MLRS TRUCK IN THE WAIT QUEUE AND SCHEDULES AN ASP
C**** EVENT IMMEDIATELY IF ONE IS FOUND.
C
C   CALLED BY UNTARV
C   CALLS      GETQUE, SCHED, PUTQUE
C   SCHEDULES ASP, ATP
C
C**** L. IVERSON DEC 81
C
C**** LOCAL VARIABLES
C   NUMQ  -- THE QUEUE NUMBER FOR MLRS TRUCKS WAITING SERVICE
C   NUMTK -- THE TRUCK NUMBER OF THE TRUCK FOUND IN NUMQ
C   NCHTK -- CHECK TRUCK FOR FIRST TRUCK IN QUEUE
C
C**** IPARM(1) = ASP/ATP NUMBER
C**** IPARM(2) = 0  -- FLAG THAT IT IS A MLRS TRUCK LEAVING THE ASP
C****           = -1 -- FLAG THAT IT IS A MLRS TRUCK LEAVING THE ATP
C
C   INCLUDE LOG,LIST
C   DIMENSION IPARM(5)
C
C**** REMOVE THE TRUCK FROM ASP SERVICE
C   IF(IPARM(2) .EQ. -1) GO TO 10
C   NASP = IPARM(1) - 10
C   IASP(NASP,8) = IASP(NASP,8) - 1
C
C * * * * IF ASP INTERDICTED, DO NOT SCHEDULE MLRS SERVICE
C   IF((NASP + 10) .EQ. ISERV(6))RETURN
C
C**** CHECK FOR ANOTHER MLRS TRUCK TO LOAD
C   NUMQ = IASP(NASP,10)
C   CALL GETQUE(NUMTK,NUMQ)
C**** IF A TRUCK IS FOUND SCHEDULE EVENT ASP
C   IF(NUMTK .EQ. 0) RETURN
C   IPARM(1) = 2
C   IPARM(2) = NASP + 10
C   IPARM(3) = NUMTK
C   IPARM(4) = NUMTK
C   CALL SCHED(7,IPARM,TIME)
C**** ADD ONE TO THE NUMBER OF MLRS TRUCKS BEING SERVED
C   IASP(NASP,3) = IASP(NASP,3) + 1
C   RETURN
C
C
C   ELSE MUST BE AN ATP
C
C   10 NATP = IPARM(1)
C   IATP(NATP,8) = IATP(NATP,8) - 1
C   IF(IATP(NATP,8) .LT. 0)PRINT*, 'DEPASP ',NATP,IATP(NATP,8),TIME
C
C * * * * IF ATP INTERDICTED, DO NOT SCHEDULE MLRS SERVICE

```

```

C      NUMQ = IATP(NATP,11)
      CALL GETQUE(NUMTK,NUMQ)
C**** SEARCH FOR RIGHT TRUCK
C      IF QUEUE IS EMPTY RETURN
      IF (NUMTK .EQ. 0) RETURN
      NCHTK = NUMTK
      CALL PUTQUE(NUMTK,NUMQ)
12     CALL GETQUE(NUMTK,NUMQ)
C      DETERMINE MIX ON TRUCK
      MIX = ITRUCK(NUMTK,5)
      IF (MIX .EQ. 10 .OR. MIX .EQ. 40) GO TO 15
C      WRONG TRUCK PUT BACK IN QUEUE
      CALL PUTQUE(NUMTK,NUMQ)
C      IF LAST TRUCK, RETURN
      IF (NUMTK .EQ. NCHTK) RETURN
      GO TO 12
C
15     IPARM(1) = 1
      IPARM(2) = NATP
      IPARM(3) = NUMTK
      IPARM(4) = NUMTK
      CALL SCHED(6,IPARM,TIME)           @ ATP
C**** ADD ONE TO THE NUMBER OF MLRS TRUCKS BEING SERVED
      IATP(NATP,8) = IATP(NATP,8) + 1
C
20     RETURN
      END

```

```

q.  SUBROUTINE DUALMX
    SUBROUTINE DUALMX(I,K,J)

C
C    RELOADS MORTARS AND BUSHMASTERS FROM THE SAME TRUCK UNTIL BOTH
C    ARE FILLED OR THE TRUCKS RUN OUT OF AMMO. THE SAME PERCENTAGE
C    OF ROUNDS WILL BE GIVEN TO EACH SYSTEM DURING THE RELOAD.
C    CALLED BY RELOAD
C    CALLS      IQ, FINTK, INTRDK, SCHED
C
C    SCHEDULES ASPARV, DEAD TRUCK -> ASP
C            UNTDEP, RETURNS UNIT TRUCK
C            UNTARV,      ''
C
C**** L. IVERSON NOV 81
C
C**** LOCAL VARIABLES
C      I      -- UNIT NUMBER
C      K      -- UNIT ATTRIBUTE FOR MORTAR AMMUNITION
C      J      -- THE UNIT ATTRIBUTE FOR BUSHMASTER AMMUNITION
C      NDMORT -- THE NUMBER OF ROUNDS SHORT OF MORTAR AMMO
C      NDBUSH -- THE NUMBER OF ROUNDS SHORT OF BUSHMASTER AMMO
C      RPMORT -- THE PERCENT OF A TRUCK LOAD OF MORTAR AMMO NEEDED
C      RPBUSH -- THE PERCENT OF A TRUCK LOAD OF BUSHMASTER AMMO NEEDED
C      NUM1   -- THE UNIT ATTRIBUTE OF THE MOST NEEDED AMMO (J OR K)
C      NUM2   -- THE UNIT ATTRIBUTE OF THE OTHER AMMO
C      ND1    -- THE NUMBER OF ROUNDS NEEDED TO FILL THE GREATEST DEMAND
C      ND2    -- THE NUMBER OF ROUNDS TO BE GIVEN TO THE OTHER WEAPON
C      NUMQ   -- THE QUEUE NUMBER FOR UNIT TRUCKS
C      NUMTK  -- THE NUMBER OF THE TRUCK TAKEN FROM NUMQ TO FILL NEED
C      THIND  -- THE DELAY TIME ASSESSED FOR A TRUCK INTERDICTION
C      LOAD1  -- THE AMOUNT OF THE MOST NEEDED AMMO ON THE TRUCK NUMTK
C      LOAD2  -- THE AMOUNT OF THE OTHER AMMO ON THE TRUCK
C      NEWLD1 -- THE ROUNDS FROM TRUCK NUMTK TO LOAD ON THE WEAPON SYSTEM
C      NEWLD2 -- THE ROUNDS TO LOAD ON THE OTHER WEAPON
C      KIND   -- THE EVENT NUMBER: 3 = UNTDEP, 8 = UNTARV
C      NRPW1  -- THE ROUNDS LOADED ON EACH WEAPON OF TYPE NUM1
C      NRPW2  -- THE ROUNDS LOADED ON EACH WEAPON OF TYPE NUM2
C      NW1    -- THE NUMBER OF WEAPONS LOADED, TYPE NUM1
C      NW2    -- THE NUMBER OF WEAPONS LOADED, TYPE NUM2
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C
C**** FIND THE AMMO TYPE WITH THE LARGEST DEMAND - NUM1
10  NDMORT = IUNIT(I,K+7) * IUNIT(I,K+1) - IUNIT(I,K+4)
    NDBUSH = IUNIT(I,J+7) * IUNIT(I,J+1) - IUNIT(I,J+4)
    RMIX16 = IMIX(16,16)
    RMIX17 = IMIX(16,17)
    RPMORT = NDMORT/RMIX16
    RPBUSH = NDBUSH/RMIX17
    IF(RPMORT .GT. RPBUSH) THEN
        NUM1 = K
        NUM2 = J
        ND1  = NDMORT

```

```

ELSE
  NUM1 = J
  NUM2 = K
  ND1 = NDBUSH
  ND2 = RMIX16 * RPBUSH
END IF

C
WRITE(LUOUT,20)IUNIT(I,NUM1),ND1,IUNIT(I,NUM2),ND2
20 FORMAT(' RELOAD AMMO TYPES ',4I5,' ROUNDS')
C**** PULL TRUCK FROM QUEUE
NUMQ = IQ(1,I)
30 CALL FINTK(NUMQ,16,NUMTK,0)
WRITE(LUOUT,40)NUMTK
40 FORMAT(' RELOAD AFTER FINTK ',I5)
IF(NUMTK.EQ. 0) RETURN
C**** CHECK FOR INTERDICTION
CALL INTRDK(NUMTK,THIND)
IF(THIND.NE. 0) THEN
C **** ADD ONE TO THE NUMBER OF TRUCKS KILLED DURING RELOAD
JUNIT(IUNIT(I,1),23) = JUNIT(IUNIT(I,1),23) + 1
C DECREMENT UNIT AMMO ON TRUCKS
IUNIT(I,J+8)=IUNIT(I,J+8)-(IMIX(16,17)*ITRUCK(NUMTK,6)+9999)/10000
IUNIT(I,K+8)=IUNIT(I,K+8)-(IMIX(16,16)*ITRUCK(NUMTK,6)+9999)/10000
IPARM(1) = I
IPARM(2) = NUMTK
IPARM(4) = 16
C SCHEDULE ASPARV FOR KILLED TRUCK
CALL SCHED(5,IPARM,TIME + THIND) @ ASPARV
ITRUCK(NUMTK,6) = 0
GO TO 30
END IF

C
C**** CALCULATE THE TRUCK AMMO LOAD FOR AMMO TYPE NUM1 AND NUM2
LOAD1 = (IMIX(16,IUNIT(I,NUM1))*ITRUCK(NUMTK,6)+9999)/10000
LOAD2 = (IMIX(16,IUNIT(I,NUM2))*ITRUCK(NUMTK,6)+9999)/10000
C**** CHECK DEMAND AGAINST LOAD
IF(ND1.GE. LOAD1) THEN
  NEWLD1 = LOAD1
  NEWLD2 = LOAD2
  ITRUCK(NUMTK,6) = 0
  KIND = 3
ELSE
  NEWLD1 = ND1
  NEWLD2 = ND2
  ITRUCK(NUMTK,6) = 10000 * (LOAD1-ND1)/IMIX(16,IUNIT(I,NUM1))
  KIND = 8
END IF
IF(ITRUCK(NUMTK,6) .LE. 1000)THEN
  NEWLD1 = LOAD1
  NEWLD2 = LOAD2
  ITRUCK(NUMTK,6) = 0
  KIND = 3
END IF
C**** CALCULATE THE UNLOAD TIME, FIND ROUNDS PER WPN AND NUM WPNS

```

```

      NRPW1 = 1
    ELSE
      NRPW1 = ND1 / IUNIT(I,NUM1+2)
    END IF
    IF(IUNIT(I,NUM2+2) .EQ. 0) THEN
      NRPW2 = 1
    ELSE
      NRPW2 = ND2 / IUNIT(I,NUM2+2)
    END IF
    IF(NRPW1 .LE. 0) NRPW1 = 1
    IF(NRPW2 .LE. 0) NRPW2 = 1
    NW1 = MINO(LOAD1/NRPW1,IUNIT(I,NUM1+2))
    NW2 = MINO(LOAD2/NRPW2,IUNIT(I,NUM2+2))
    DELAY = 2 * IRSTME(16,3) + (NW1 + NW2) * (IRSTME(16,1))
    IF(NUM1 .EQ. J) THEN
      DELAY = DELAY + NEWLD2 * IRSTME(16,2)/100
    ELSE
      DELAY = DELAY + NEWLD1 * IRSTME(16,2)/100
    END IF
C**** SEND THE TRUCK OUT
    IPARM(1) = I
    IPARM(2) = NUMTK
    CALL SCHED(KIND,IPARM,TIME + DELAY)      @ UNTDEP/UNTARV
    ITRUCK(NUMTK,11) = ITRUCK(NUMTK,11) + 1 @ RELOAD COUNTER
    ITRUCK(NUMTK,3) = 0
C**** ADJUST AMMO ON TRUCKS AND CURRENT AMMO SUPPLY
    IUNIT(I,NUM1+8) = IUNIT(I,NUM1+8) - NEWLD1
    IUNIT(I,NUM2+8) = IUNIT(I,NUM2+8) - NEWLD2
    IUNIT(I,NUM1+4) = IUNIT(I,NUM1+4) + NEWLD1
    IUNIT(I,NUM2+4) = IUNIT(I,NUM2+4) + NEWLD2
C**** DECREMENT THE NUMBER OF ROUNDS SHORT
    IUNIT(I,NUM1+3) = IUNIT(I,NUM1+3) - NEWLD1
    IF(IUNIT(I,NUM1+3) .LT. 0) IUNIT(I,NUM1+3) = 0
    IUNIT(I,NUM2+3) = IUNIT(I,NUM2+3) - NEWLD2
    IF(IUNIT(I,NUM2+3) .LT. 0) IUNIT(I,NUM2+3) = 0
    IUNIT(I,NUM1+2) = IUNIT(I,NUM1+2) - NW1
    IF(IUNIT(I,NUM1+2) .LT. 0) IUNIT(I,NUM1+2) = 0
    IUNIT(I,NUM2+2) = IUNIT(I,NUM2+2) - NW2
    IF(IUNIT(I,NUM2+2) .LT. 0) IUNIT(I,NUM2+2) = 0
C**** ADD THE DELAY TIME TO THE TOTAL UNIT RELOAD TIME
    JUNIT(IUNIT(I,1),21) = JUNIT(IUNIT(I,1),21) + DELAY
    IF(IUNIT(I,NUM1+7)*IUNIT(I,NUM1+1)-IUNIT(I,NUM1+4).GT.0) GO TO 10
C
    RETURN
  END

```

```

r.      SUBROUTINE EDITD
        SUBROUTINE EDITD
C****  ALLOWS EDITING OF DATA IN COMMON LOG
C      CALLED BY EDIT, CONTRL(THE EDIT OR ARM MENUS)
C      CALLS      READF
C****  H. JONES      FEB 79
C****  NOTE ALL VARIABLES IN COMMON LOG ARE 2 DIMENSIONAL
        CHARACTER*10 AUNIT, NAME, IWORD, IEND
        COMMON /LOG/ IATP(10,53), IASP(10,110), IUNIT(75,142),
$   ITRUCK(1400,15), ITYPE(9,6), IMIX(91,32), INTER(1,10),
$  IRSTME(23,3), IATPSD(1,5),
$   IDAY(1,1), TIME(1,1), IATPAM(10,40), ICSA(3,32), LPPAR(1,10),
$   IASPAM(10,120), LUOUT(1,1), TCIST(1,1), TCILNG(1,1), LOOK(1,19),
$   ,JUNIT(8,24), JATP(10,6), JASP(10,9)
$   ,IATPSP(10,22), IASPSP(10,30), IAMLVL(2,30), ISERV(1,10)
        COMMON / AUNIT / AUNIT(75,2)
        DIMENSION  INTGR(10), REAL(10), IWORD(10)
        INTEGER*4 SKIP
        DIMENSION NAME(27), LIMIT1(27), LIMIT2(27)

C
        DATA NAME /'IATP', 'IASP', 'IUNIT', 'ITRUCK', 'ITYPE',
$   'IMIX', 'INTER', 'IRSTME', 'IATPSD', 'IDAY',
$   'TIME', 'IATPAM', 'ICSA', 'LPPAR', 'IASPAM', 'LUOUT', 'TCIST',
$   'TCILNG', 'LOOK', 'AUNIT', 'JUNIT', 'JATP', 'JASP',
$   'IATPSP', 'IASPSP', 'IAMLVL', 'ISERV'/

C
        DATA LIMIT1 /10, 10, 75, 1400, 9,
$   91, 1, 23, 1, 1,
$   1, 10, 3, 1, 10, 1, 1, 1, 19, 75, 8, 10,10,10,10,2,1/

C
        DATA LIMIT2 /53, 110, 142, 15, 6,
$   32, 10, 3, 5, 1,
$   1, 40, 32, 10, 120, 1, 1, 1, 1, 2, 24, 6, 9,22,30,30,10/
        DATA IEND/'END'/
        NNames = 27
        LU1 = 5
        LEAP = 1
        SKIP = 1

C
        10 WRITE(6,120)
           CALL READF (LU1, 10, INTGR, REAL, IWORD)

C
C****  BRANCH ON DATA TYPE
        20 IF(IWORD(1) .EQ. IEND) GO TO 110
           DO 30 KTYPE = 1,NNames
              IF(IWORD(1) .EQ. NAME(KTYPE)) GO TO 40
        30 CONTINUE
           GO TO 10

C
C****  SET LIMITS FOR DATA TYPE
        40 ILOW = INTGR(1)
           IHIGH = INTGR(2)
           IFLG = 0
           IF(ILOW .EQ. 0 .AND. IHIGH .EQ. 0) IFLG = 1

```

```

IF(IFLG .EQ. 1) IHIGH = LIMIT1(KTYPE)
IF(IHIGH .EQ. 0) IHIGH = ILOW
IF(IHIGH .GT. LIMIT1(KTYPE)) IHIGH = LIMIT1(KTYPE)
IF(ILOW .GT. LIMIT1(KTYPE)) GO TO 10

C
C**** BACKGROUND HAS BEEN SET, READ CHANGE OR LIST COMMAND
50 WRITE(6,140)
CALL READF (LU1, 10, INTGR, REAL, IWORD)
IF(IWORD(1) .EQ. 'LIST' .OR. IWORD(1) .EQ. 'L') GO TO 60
IF(IWORD(1) .EQ. 'CHANGE' .OR. IWORD(1) .EQ. 'C') GO TO 90
IF(IWORD(1) .EQ. 'LEAP') THEN
    LEAP = INTGR(1)
    GOTO 50
ELSE
    IF(IWORD(1) .EQ. 'SKIP') THEN
        SKIP = INTGR(1)
        GOTO 50
    END IF
END IF
GO TO 20

C
C**** LIST COMMAND
60 IATT1 = INTGR(1)
IATT2 = INTGR(2)
IFLG = 0
IF(IATT1 .EQ. 0 .AND. IATT2 .EQ. 0) IFLG = 1
IF(IFLG .EQ. 1) IATT1 = 1
IF(IFLG .EQ. 1) IATT2 = LIMIT2(KTYPE)
IF(IATT2 .EQ. 0) IATT2 = IATT1
IF(IATT2 .GT. LIMIT2(KTYPE)) IATT2 = LIMIT2(KTYPE)

C
IF(IATT1 .GT. LIMIT2(KTYPE)) GO TO 50
DO 80 INDEX = ILOW, IHIGH, LEAP
WRITE(6,150) NAME(KTYPE), INDEX
DO 90 IATT = IATT1, IATT2, SKIP
IF(KTYPE .EQ. 1) IVALUE = IATP(INDEX, IATT)
IF(KTYPE .EQ. 2) IVALUE = IASP(INDEX, IATT)
IF(KTYPE .EQ. 3) IVALUE = IUNIT(INDEX, IATT)
IF(KTYPE .EQ. 4) IVALUE = ITRUCK(INDEX, IATT)
IF(KTYPE .EQ. 5) IVALUE = ITYPE(INDEX, IATT)
IF(KTYPE .EQ. 6) IVALUE = IMIX(INDEX, IATT)
IF(KTYPE .EQ. 7) IVALUE = INTER(INDEX, IATT)
IF(KTYPE .EQ. 8) IVALUE = IRSTME(INDEX, IATT)
IF(KTYPE .EQ. 9) IVALUE = IATPSD(INDEX, IATT)
IF(KTYPE .EQ. 10) IVALUE = IDAY(INDEX, IATT)
IF(KTYPE .EQ. 11) IVALUE = TIME(INDEX, IATT)
IF(KTYPE .EQ. 12) IVALUE = IATPAM(INDEX, IATT)
IF(KTYPE .EQ. 13) IVALUE = ICSA(INDEX, IATT)
IF(KTYPE .EQ. 14) IVALUE = LPPAR(INDEX, IATT)
IF(KTYPE .EQ. 15) IVALUE = IASPAM(INDEX, IATT)
IF(KTYPE .EQ. 16) IVALUE = LUOUT(INDEX, IATT)
IF(KTYPE .EQ. 17) IVALUE = TCIST(INDEX, IATT)
IF(KTYPE .EQ. 18) IVALUE = TCILNG(INDEX, IATT)
IF(KTYPE .EQ. 19) IVALUE = LOOK(INDEX, IATT)

```

```

        WRITE(6,210) IATT, AUNIT(INDEX,IATT)
        GO TO 50
    END IF
    IF(KTYPE .EQ. 21) IVALUE = JUNIT(INDEX,IATT)
    IF(KTYPE .EQ. 22) IVALUE = JATP(INDEX,IATT)
    IF(KTYPE .EQ. 23) IVALUE = JASP(INDEX,IATT)
    IF(KTYPE .EQ. 24) IVALUE = IATPSP(INDEX,IATT)
    IF(KTYPE .EQ. 25) IVALUE = IASPSP(INDEX,IATT)
    IF(KTYPE .EQ. 26) IVALUE = IAMLVL(INDEX,IATT)
    IF(KTYPE .EQ. 27) IVALUE = ISERV(INDEX,IATT)
    IF(KTYPE .LE. 0 .OR. KTYPE .GT. 27) GO TO 40
C
    70 WRITE(6,160) IATT, IVALUE
    30 CONTINUE
    GO TO 50
    55 WRITE(6,210) IATT, AUNIT(INDEX,IATT)
    GO TO 50
C
C
C**** CHANGE COMMAND
    90 IATT = INTGR(1)
    VALUE = INTGR(2) + REAL(1)
    IF(IATT .GT. LIMIT2(KTYPE)) GO TO 50
C
    DO 100 INDEX = ILOW, IHIGH, LEAF
C
    INSERT VALUE IN PROPER ARRAY
    IF(KTYPE .EQ. 1) IATP(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 2) IASP(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 3) IUNIT(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 4) ITRUCK(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 5) ITYPE(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 6) IMIX(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 7) INTER(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 8) IRSTME(INDEX, IATT) = VALUE
    IF(KTYPE .EQ. 9) IATPSD(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 10) IDAY(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 11) TIME(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 12) IATPAM(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 13) ICSA(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 14) LPPAR(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 15) IASPAM(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 16) LUOUT(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 17) TCIST(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 18) TCILNG(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 19) LOOK(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 20) THEN
        WRITE(6,200)
        READ(5,130) IWORD(2)
        AUNIT(INDEX,IATT) = IWORD(2)
    END IF
    IF(KTYPE .EQ. 21) JUNIT(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 22) JATP(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 23) JASP(INDEX,IATT) = VALUE
    IF(KTYPE .EQ. 24) IATPSP(INDEX,IATT) = VALUE

```

```
IF(KTYPE .EQ. 26) IAMLVL(INDEX,IATT) = VALUE  
IF(KTYPE .EQ. 27) ISERV(INDEX,IATT) = VALUE
```

C

```
100 CONTINUE  
GO TO 50
```

C

```
110 RETURN  
120 FORMAT(1X,'VARIABLE NAME = ')  
130 FORMAT(A10)  
140 FORMAT(1X,'..')  
150 FORMAT(/,1X,A10,I5)  
160 FORMAT(1X,'ATTRIBUTE ',I4,' = ',I7)  
200 FORMAT(1X,' UNIT NAME = ')  
210 FORMAT(1X,' ATTRIBUTE ',I4,' = ',A10)  
END
```

```

S.      SUBROUTINE ENDSIM
        SUBROUTINE ENDSIM(IPARM)
C      EVENT TYPE 19
C**** SIMULATION END
C
C      WRITES FILE 4 (THE TDATABASE FOR THE NEXT CI).
C
C      CALLED BY MAINARM
C      SCHEDULED BY CONTRL, INIT
C
C**** H. JONES      FEB 79
C
C      CHARACTER*10 AUNIT
C      INCLUDE LOG,LIST
C      INCLUDE QUENUM,LIST
C      INCLUDE QUEPNT,LIST
C      INCLUDE AUNIT,LIST
C      DIMENSION IPARM(5)
C
C      WRITE(4) IATP,IASP,IUNIT,ITRUCK,ITYPE,IMIX,INTER,IRSTME,
Z      IATPSD,IDAY,TIME,IATPAM,ICSA,LPPAR,IASPAM,LUOUT,TCIST,
Z      TCILNG,LOOK,IHEAD,ITEMS,AUNIT,JUNIT,JATP,JASP,
Z      IATPSP,IASPSP,IAMLVL,ISERV
C
C**** WRITES TERMINATION MESSAGE
        IF(IPARM(1).EQ.9999) WRITE(6,10) TIME @ NORMAL
        IF(IPARM(1).EQ.8888) WRITE(6,11) TIME @ ABNORMAL
10  FORMAT(1X,'SCHEDULED STOP, TIME = ',F8.1,/,1X,'SSG ARMPL.FREE;
Z      THEN SSG ARMPL.EDITYES')
11  FORMAT(1X,'STOP SIIMULATION FROM CONTROL, TIME = ',F8.1)
C
C      RETURN
C      END

```

```

t.      SUBROUTINE EVSTOP
        SUBROUTINE EVSTOP
C        CALLED BY MAINARM
C***** WRITES UNUSED EVENTS ON THE EVENTS FILE FOR THE NEXT CI.
C***** EVTCT--.          (FILE 3)
C***** H. JONES    FEB 79
C
        INCLUDE EVENTS,LIST
        WRITE(3) JSTAT, JEVDS, IEVS
C
        RETURN
        END

```

AD-A131 218

AMMUNITION RESUPPLY MODEL VOLUME 2 PROGRAMMER'S MANUAL

2/3

(U) COMBINED ARMS OPERATIONS RESEARCH ACTIVITY FORT

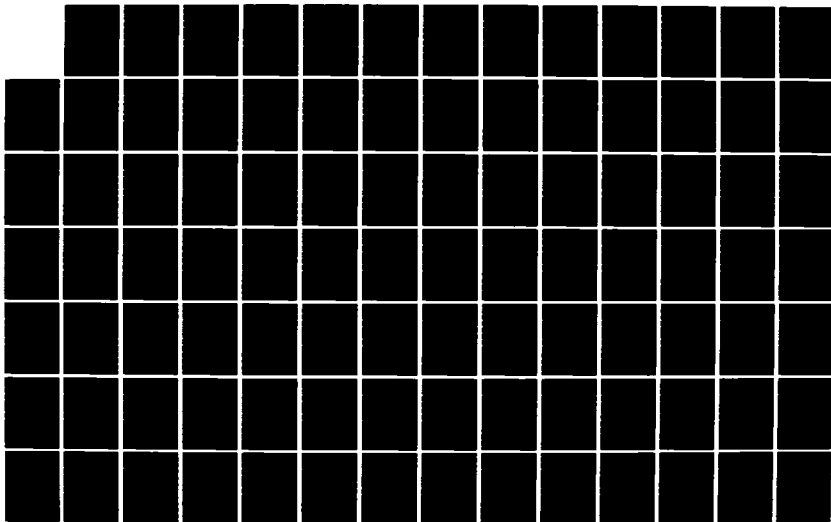
LEAVENWORTH KS A M RESNICK ET AL. MAY 83

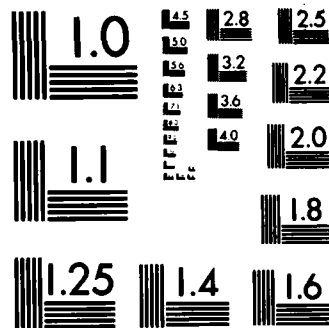
UNCLASSIFIED

CAORA/TR-6/83-VOL-2

F/G 19/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

u.      SUBROUTINE FINTK
        SUBROUTINE FINTK (NQUE, NUMAM, NUMTK, MINMAX)
C**** DETERMINES NUMBER OF TRUCK (NUMTK) IN QUEUE (NQUE) HAS
C**** THE SMALLEST PERCENTAGE LOAD OF ROUNDS OF TYPE (NUMAM)
C**** JIM FOX          JAN 79
C
C      CALLED BY ASP, ASPAR1, ATP, ATPARV, DUALMX, LDPWDR, RELOAD, SERVER
C      CALLS      GETQUE, PUTQUE
C
C**** LOCAL VARIABLES:
C**** ITRCK  -- SAVES TRUCK NUMBER WITH THE SMALLEST LOAD
C**** NPERSV -- SMALLEST PERCENT FOUND
C**** NCHTK  -- END OF QUEUE CHECK
C**** MIX    -- AMMO MIX INDEX
C**** MINMAX -- 0, FIND TRUCK WITH SMALLEST LOAD
C              1, FIND TRUCK WITH LARGEST LOAD
C
C      INCLUDE LOG,LIST
C**** INITIALIZE AMMO PERCENT
C      IF(MINMAX .EQ. 0) THEN
C          NPERSV = 10010
C      ELSE
C          NPERSV = 0
C      END IF
C
C**** ASSUME NO TRUCK WITH PROPER AMMO
C      NUMTK = 0
C
C**** BRING FIRST TRUCK FROM QUEUE (ITRCK)
C      CALL GETQUE (ITRCK, NQUE)
C
C**** IF QUEUE IS EMPTY RETURN
C      IF(ITRCK .EQ. 0) RETURN
C
C**** THERE ARE SOME TRUCKS IN QUEUE
C**** SEARCH FOR RIGHT TRUCK, STORE NCHTK AND PUT IT BACK IN QUEUE
C      NCHTK = ITRCK
C      CALL PUTQUE(ITRCK, NQUE)
C
C**** PULL TRUCK FROM QUEUE
C      25 CALL GETQUE(ITRCK,NQUE)
C
C**** DETERMINE AMMO MIX TYPE
C      MIX = ITRUCK(ITRCK, 5)
C      LOAD = ITRUCK(ITRCK,6)
C
C      IF THIS TRUCK HAS THE DESIRED LOAD, THEN KEEP TRACK IN NUMTK.
C**** IF RIGHT AMMO COMPARE LOAD SIZE; IF NOT GO TO CHECK END QUEUE
C      IF(MIX(MIX, NUMAM) .GT. 0) THEN
C          IF(MINMAX .EQ. 0 .AND. LOAD .LE. NPERSV .OR.
C          *   MINMAX .GE. 1 .AND. LOAD .GE. NPERSV) THEN
C              IF(ITRUCK(ITRCK,6) .NE. 0) THEN
C                  IF(NUMTK .NE. 0) CALL PUTQUE(NUMTK, NQUE)
C                  NPERSV = ITRUCK(ITRCK,6)
C                  NUMTK = ITRCK

```

```

        RETURN
      END IF
    END IF
  END IF
C**** WRONG TRUCK, PUT BACK IN QUEUE
  CALL PUTQUE(ITRCK,NQUE)
C
C**** IF LAST TRUCK, RETURN
  IF(ITRCK .NE. NCHTK) GO TO 25
C
C**** HAVE FOUND CORRECT AMMO CHECK TO SEE IF SHOULD BE SAVED
  RETURN
END

```

```

V.  SUBROUTINE GETEVT
    SUBROUTINE GETEVT(IEVT,ITH,ITS)
    GETS NEXT EVENT (FROM EVENT QUEUE)

C
C
C    CALLED BY NEXTEV, WHICH RETURNS THE EVENT TO MAINARM.
C    GETEVT DOES NOT CALL ANYTHING.
C
C    RETURNS IEVT(5) -- THE PARM5 FOR THE NEXT EVENT IN QUEUE.
C           ITH      -- EVENT TIME, IN WHOLE MINUTES.
C           ITS      -- THE FRACTIONAL PART OF EVENT TIME(MIN * 3600).
C
C    BOB DAVIDSON
C
C    INCLUDE EVENTS,LIST
C    DIMENSION IEVT(5),JFORE(2048),JBACK(2048)
C    EQUIVALENCE (JFORE(1),JEVDS(1,1)),(JBAC(1),JEVDS(1,2)),
Z  (JFIRST,JSTAT(1)),(JLAST,JSTAT(2)),(JEMPTY,JSTAT(3)),
Z  (NUMEVT,JSTAT(4)),(NEMPTY,JSTAT(5)),(MAXEVT,JSTAT(6))
C
C    CHECK IF AN EVENTS IN QUEUE ... IF NONE, RETURN
C
C    DO 20 IN = 1,5
        IEVT(IN) = IEVS(IN,JFIRST)           @ PARM5 FOR THE NEXT EVT
20  CONTINUE
        ITH = JEVDS (JFIRST,3)
        ITS = JEVDS (JFIRST,4)
        IF(NUMEVT.LE.0) RETURN
C
        NEXT = JFORE(JFIRST)
        JFORE (JFIRST) = JEMPTY              @ ADD THIS NODE TO
                                           @ THE EMPTY EVENT LIST
        JEMPTY=JFIRST
        IF(NEXT.LE.0) NEXT = 1
        JBACK(NEXT) = 0
        JFIRST = NEXT
        NEMPTY = NEMPTY + 1
        NUMEVT = NUMEVT - 1
C
    RETURN
    END

```

```

W.      SUBROUTINE GETQUE
        SUBROUTINE GETQUE (ITEM, NUMQUE)
C      GETS 'FIRST ITEM IN QUEUE' FROM NUMQUE
C      TO GET TRUCK FROM QUEUE 4 -- CALL GETQUE (N,4)
C      VEHICLE NUMBER IS RETRUNED IN ITEM
C
C      CALLED BY ASPAR1, ASPAR2, ASPARV, ATPARV, CSAARV, CSADEP,
C      DEPASP, FINTK, SERVER (EDIT PROG: EDIT, PRINT, TRKPUT)
C      CALLS      NONE
C**** H. JONES      DEC 78
        INCLUDE QUENUM,LIST
        INCLUDE QUEPNT,LIST
        ITEM = 0
        LITEM = 0
C
C      IF(NUMQUE .LE. 0 .OR. NUMQUE .GT. 176) THEN
C          PRINT*, 'BAD QUEUE NUMBER IN GETQUE...'
C          RETURN
C      END IF
C      IPOINT = IHEAD(NUMQUE)
C
C
C          SEARCH QUEUE FOR FRONT
10 IF(IPOINT .EQ. 0) GO TO 20
    LITEM = ITEM
    ITEM = IPOINT
    IPOINT = ITEMS(ITEM)
    GO TO 10
C
20 IF(LITEM .GT. 0) ITEMS(LITEM) = 0
    IF(LITEM .EQ. 0) IHEAD(NUMQUE) = 0
C
    RETURN
    END

```

```

X.      SUBROUTINE HASPAR
        SUBROUTINE HASPAR (IPARM)
C**** EVENT HASPAR -- ARRIVAL(RETURN) OF HELICOPTER BACK AT ASF.
C      EVENT TYPE IS
C      CALLED BY MAINARM
C      CALLS      NOTHING
C
C**** J. FOX      JAN 79
C
C**** IPARM(1)  -- UNIT *(IF SCHED BY DEMAND), ASF*(IF SCHED BY HELARV)
C**** IPARM(2)  -- HELICOPTER NUMBER
C
C**** SCHEDULES -- NOTHING
C**** SCHEDULED BY DEMAND, HELARV
C
C**** CHANGES NUMBER OF HELOS IN USE(LPPAR(5)).
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C
C      LOCAL VARIABLES
C      NHELI -- HELICOPTER NUMBER FROM ITRUCK ARRAY
C
C***** INCREMENT NUMBER OF HELICOPTER AVAILABLE FOR USE
C      LPPAR(5) = LPPAR(5) + 1
C
C      NHELI = IPARM(2)
C
C***** CHANGE STATUS TYPE
C      ITRUCK (NHELI,3) = 3
C
C      ITRUCK (NHELI,6) = 10000
C      RETURN
C      END
C
C      HELI KEPT FULL

```

```

y.      SUBROUTINE HELARV
        SUBROUTINE HELARV (IPARM)
C**** EVENT HELARV  --  ARRIVAL OF HELICOPTER AT UNIT
C                                FOR EMERGENCY RESUPPLY OF ARTY.
C      EVENT TYPE 14
C      CALLED BY MAINARM
C      CALLS      OPERA, SCHED
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- HELICOPTER NUMBER
C
C**** SCHEDULES  -- HASPAR, ARRIVAL BACK AT ASP.
C**** SCHEDULED BY- DEMAND
C
C**** CHANGES NUMBER OF HELOS IN USE TO SUPPORT THIS UNIT.
C              AMMO AT UNIT.
C
C      LOCAL VARIABLE DEFINITION
C      MIX - AMMO MIX INDEX TO COMMON IMIX-
C      IND - INDEX OF CURRENT AMMO SUPPLIES IN IUNIT
C      IN - INDEX OF TYPE AMMO IN IUNIT
C      IAMTYP - UNIT IN THE AMMO TYPE
C      TULTIM - TRAVEL TIME
C      TFAIL - TIME LOST DUE TO MAINTENANCE FAILURE
C      TMIND - TIME LOST DUE TO INTERDICTION
C      TOTIM - TIME OF SCHEDULED EVENT
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C
C      NUNIT = IPARM(1)
C      NHELI = IPARM(2)
C
C      FIND THE MIX INDEX CARRIED BY HELICOPTER
C      MIX = ITRUCK(NHELI,5)
C      IF(MIX .LE. 0)RETURN
C      INCREMENT THE AMMO ON HAND AT THE UNIT
C      DO 5 I = 1,LPPAR(6)
C          IND = 13 * I - 5
C          IN = IND + 4
C      FIND UNIT I TH AMMO TYPE
C      IAMTYP = IUNIT(NUNIT,IND)
C      IUNIT(NUNIT,IN) = IUNIT(NUNIT,IN) + IMIX(MIX,IAMTYP)
C      IUNIT(NUNIT,IN-1) = IUNIT(NUNIT,IN-1) - IMIX(MIX,IAMTYP)
C      5 CONTINUE
C      FIND TRAVEL TIME BACK TO ASP
C      TULTIM = 60. * IUNIT(NUNIT,5) / ITYPE(6,IDAY+1)
C      FIND THE DELAY ASSOCIATED WITH MECHANICAL FAILURE
C      CALL OPERA(NHELI,TULTIM,TFAIL)
C**** DECREMENT NUMBER OF HELO SERVING THIS UNIT
C      IUNIT(NUNIT,138) = IUNIT(NUNIT,138) - 1
C**** ADD NO. OF HELO THAT HAVE SERVED UNIT

```

```

C   SCHEDULE HELARV AT ASP
    IPARM(1) = IUNIT(NUNIT,3)
    TOTIM = TIME + TULTIM + TFAIL
    CALL SCHED(15,IPARM,TOTIM)           @ HASPAR
C   UPDATE ITRUCK STATUS
    ITRUCK(NHELI,3) = 4
    ITRUCK(NHELI,6) = 0                 @ LOAD
C
    RETURN
    END

```

```

Z.      SUBROUTINE INIT
        SUBROUTINE INIT
C**** INITIALIZES SIMULATION
C      CALLED FROM MAINARM AT START OF THE CI.
C      CALLS RDJIFF TO INITIALIZE DEMAND FOR THE WHOLE CI.
C      ALSO:   QINIT, SEVENT, SCHED, TRKTIM, CONTRL
C
C      SCHEDULES:   ENDSIM
C**** H. JONES   JAN 79
C
        CHARACTER*3 IANSW
        CHARACTER*10 AUNIT
        INCLUDE LOG
        INCLUDE QUENUM,LIST
        INCLUDE QUEPNT,LIST
        INCLUDE AUNIT,LIST
        INCLUDE EVENTS,LIST
        DIMENSION IPARM(5)
C
C**** READ FILES WITH ALL COMMON DATA
        READ(3) IATP,IASP,IUNIT,ITRUCK,ITYPE,IMIX,INTER,IRSTME,
Z      IATPSD,IDAY,TIME,IATPAM,ICSA,LPPAR,IASPAM,LUOUT,TCIST,
Z      TCILNG, LOOK, IHEAD, ITEMS, AUNIT, JUNIT, JATP, JASP,
Z      IATPSP,IASPSP,IAMLV,ISERV
C
        READ(7)JSTAT,JEVDS,IEVS
C
        IF(TIME .LT. 10.0)CALL QINIT
21  FORMAT(A3)
        PRINT*, 'DO YOU WISH TO ADD EVENTS?'
        READ(5,21)IANSW
        IF(IANSW .EQ. 'YES' .OR. IANSW .EQ. 'Y ' )THEN
            CALL SEVENT          @ READ ADDTNL EVENTS FILE(BUILT BY ADDEVT)
        END IF
C
        WRITE(6,22)
22  FORMAT(' ENTER TIME TO STOP SIMULATION  ')
        READ(5,*) TSTOP
        IPARM(1) = 9999
        IPARM(2) = 9999
        IPARM(3) = 9999
        IPARM(4) = 9999
        CALL SCHED (19, IPARM, TSTOP)          @ ENDSIM
C
        IF(TIME .LE. 10.)CALL TRKTIM
        CALL CONTRL (TIME)
        TIME = TCIST
C
C**** READ FILE FOR DEMANDS
        CALL RDJIFF
        RETURN
        END

```

```

aa.  SUBROUTINE INTRDK
      SUBROUTINE INTRDK(NUMTK,TLOST)

C
C   DETERMINES IF A TRUCK ABOUT TO TRAVEL A ROUTE
C   WILL BE INTERDICTED ALONG THAT ROUTE AND ASSESSES
C   TIME DELAY FOR A REPLACEMENT TRUCK.
C
C   REPLACEMENT TRUCKS SHOULD BE SCHEDULED INTO THE ASP
C   AFTER A DELAY TIME OF 'TLOST' MINUTES.
C   TWO DEPTH ZONES ARE CONSIDERED.
C
C   CALLED BY ASP, ASPAR1, ASPAR2, ATP, ATPAR1, ATPAR2, ATPARV,
C   CSADEP, DUALMX, RELOAD, UNTDEP
C   CALLS      NOTHING
C
C**** J. FOX      JAN 79
C
C****  ZONE 1  ALL UNIT TRUCKS SAVE THOSE DIVERTED FROM ATP TO ASP
C****  ALL OTHER REPLENISHMENT TRUCKS.
C
C   NUMTK -- THE NUMBER OF THE TRUCK BEING CONSIDERED.
C   TLOST -- 0 IF TRUCK IS NOT KILLED
C           -- REPLACEMENT TIME IF THE TRUCK IS KILLED.
C   MODCK -- USED TO FIGURE MODULAR ARITH
C
C**** SETS LOAD OF REPLACEMENT TRUCK TO 100 PER CENT
C   INCLUDE LOG,LIST
C   TLOST = 0.          @ ASSUME TK MADE IT
C**** DETERMINE IF THE TRUCK IS IN ZONE 2. MISSION > 1
C   IF(ITRUCK(NUMTK,2) .GT. 1)GO TO 15
C**** TRUCK IS TRAVELING THROUGH ZONE 1
C**** INCREMENT COUNTER OF TRUCKS IN ZONE 1
C   INTER(9) = INTER(9) + 1
C**** IF SUFFICIENT NUMBER OF KILLS THIS CI RETURN
C   IF(INTER(1) .GE. INTER(3))RETURN
C
C**** IF NOT EQUAL 0 MODULO INTER(7) DO NOT KILL,GO TO RETURN
C   MODCK =(INTER(9) / INTER(7)) * INTER(7)
C   IF(MODCK .NE. INTER(9))RETURN
C****  CHECK IF UNIT HAS HAD A TRUCK INTERDICTED THIS CI
C   L = ITRUCK(NUMTK,4)
C
C                                     ONLY ONE TRUCK INTRDKED PER CI
C   IF(IUNIT(L,6) .GE. TCIST) RETURN
C   INTER(1) = INTER(1) + 1
C   TLOST = INTER(5)
C
C   IF TRUCK WAS ON RESUPPLY RUN, SUBTRACT AHMO DUE IN
C   IF(ITRUCK(NUMTK,3) .GE. 4 .AND. ITRUCK(NUMTK,3) .LE. 6)THEN
C     L = ITRUCK(NUMTK,4)
C     MIX = ITRUCK(NUMTK,5)
C     DO 10 I = 1,LPPAR(6)
C       K = I * 13 - 5
C       IF(IUNIT(L,K) .EQ. 0)GO TO 10
C       IUNIT(L,K+12) = IUNIT(L,K+12) - IMIX(MIX,IUNIT(L,K))

```

```

END IF
IF(ITRUCK(NUMTK,5) .EQ. 10) ITRUCK(NUMTK,5) = 10 + LPPAR(7)
ITRUCK(NUMTK,3) = 7
ITRUCK(NUMTK,9) = ITRUCK(NUMTK,9) + 1 @ TRUCK INTRDK COUNTER
IF(ITRUCK(NUMTK,1) .EQ. 1 ) THEN
    ITRUCK(NUMTK,1) = 3
    MIX = ITRUCK(NUMTK,5) + 30
    ITRUCK(NUMTK,5) = MIX
END IF
IUNIT(L,6) = TIME
WRITE(LUOUT,30) NUMTK,INTER(5)
30 FORMAT(' HAVE KILLED ZONE 1 TRUCK ',IS,' TIME LOST = ',I6)
RETURN
C
C
C**** ZONE 2 TRUCK. INCREMENT NUMBER OF ZONE 2 TRIPS.
15 INTER(10) = INTER(10) + 1
C**** IF SUFFICIENT ZONE TWO TRUCKS ALREADY KILLED GO TO RETURN
IF(INTER(2) .GE. INTER(4)) RETURN
C**** IF NOT ZERO MNDE INTER(9), DO NOT KILL
MODCK = (INTER(10) / INTER(8)) * INTER(8)
IF(MODCK .NE. INTER(10)) RETURN
C
C**** HAVE KILLED THIS TRUCK. INCREMENT NUMBER KILLED
ITRUCK(NUMTK,3)=7
ITRUCK(NUMTK,9) = ITRUCK(NUMTK,9) + 1 @ TRUCK INTRDK COUNTER
INTER(2) = INTER(2) +1
C**** SET TIME LOST. ASSUME NOT A UNIT TRUCK.
TLOST = INTER(6)
WRITE(LUOUT,20) NUMTK,TLOST
20 FORMAT(' HAVE KILLED ZONE 2 TRUCK',IS,' TIME LOST = ',F6.1)
C
RETURN
END

```

```

bb.      SUBROUTINE IQ
          FUNCTION IQ(ITYPE, NUM)
C**** RETURNS QUEUE NUMBER ASSOCIATED WITH NUM.
C      CALLED BY ASP, ASPAR1, ASPAR2, ASPARV, ATP, ATPAR1, ATPAR2,
C      ATPARV, DUALMX, LDPWDR, RELOAD, SERVER, UNTARV
C**** JIM FOX      JAN 79
          IF(ITYPE .LT. 0 .OR. ITYPE .GT. 12) THEN
              PRINT*, ' BAD QUEUE NUMBER IN FUNCTION IQ '
              STOP ' IQ '
          END IF
          GO TO (10,20,30,40,50,60,70,80,90,100,110,120), ITYPE
C**** UNIT QUEUE
          10 IQ = NUM
             GO TO 200
C
C**** ATP QUEUE FOR CSA-ATP S & P'S
          20 IQ = 75 + NUM
             GO TO 200
C
C**** ATP QUEUE FOR ASP-ATP S & P'S
          30 IQ = 85 + NUM
             GO TO 200
C
C**** ARTILLERY SERVER QUEUE AT THE ATP
          40 IQ = 95 + NUM
             GO TO 200
C
C**** MANEUVER SERVER QUEUE AT THE ATP
          50 IQ = 105 + NUM
             GO TO 200
C
C**** QUEUE TO HOLD SERVERS AT ATP
          60 IQ = 115 + NUM
             GO TO 200
C
C**** ASP QUEUE FOR CSA-ASP S & P'S
          70 IQ = 125 + NUM
             GO TO 200
C
C**** ROUTINE SERVER QUEUE AT THE ASP
          80 IQ = 135 + NUM
             GO TO 200
C
C**** MLRS SERVER QUEUE AT THE ASP
          90 IQ = 145 + NUM
             GO TO 200
C
C**** SERVER QUEUE AT ASP
          100 IQ = 155 + NUM
              GO TO 200
C
C**** S & P QUEUE AT CSA
          110 IQ = 165 + NUM
              GO TO 200

```

120 IQ = 176
200 RETURN
END

```

CC.      SUBROUTINE LDPWDR
          SUBROUTINE LDPWDR(NRND,IPARM,ITYP)
C**** UNLOADS POWDER TRUCK WHEN ARTY AMMO TAKEN FROM ATP
C
C      CALLED BY ATP
C      CALLS      IQ, FINTK, PUTQUE
C
C**** J. FOX      JAN 79
C
C**** IPARM IS IDENTICAL TO ATP
C**** NOTHING IS RETURNED
C
C**** IPARM(2) -- ATP NUMBER
C
C**** LOCAL VARIABLE DEFINITION
C**** MIX      - NUMBER OF AMMO MIX ON TRUCK FOR COMMON IMIX
C**** NRNTK    - NUMBER OF POWDER CHARGES ON THE TRUCK
C**** NR       - AMMO TYPE FOR POWDER
C**** NPCND    - NUMBER OF CANISTERS STILL NEEDED
C**** NQUE     - ASP-ATP QUEUE NUMBER
C**** NPDRSP   - POWDER AMMO TRUCK
C**** NRND     - NUMBER OF POWDER CANISTERS NEEDED
C**** NRNTK    - NUMBER OF CANISTERS ON TRUCK
C**** NFZSP    - TRUCK NUMBER OF FUZE S&P
C**** NFZQ     - QUEUE WHERE FUZE TRUCKS ARE FOUND
C**** ITYP     - AMMO TYPE
C
          INCLUDE LOG,LIST
          DIMENSION IPARM(5)
C
          NATP = IPARM(2)
C
C**** SET AMMO TYPE AND NUMBER OF ROUNDS NEEDED
C      **HARDWIRE FOR ARTY TYPE
          NR = 3
          IF(ITYP .GT. 5)NR = 8
          NPCND = NRND
C**** FIND S & P, SAVE QUEUE WE ARE WORKING IN
          S NQUE = IQ(IATPSD(2),NATP)
          CALL FINTK(NQUE,NR,NPDRSP,0)
C**** IF HAVE TRUCK GO TO 10, ELSE CHECK CSA QUEUE
          IF(NPDRSP .GT. 0)GO TO 10
C
          NQUE = IQ(IATPSD(3),NATP)
          CALL FINTK(NQUE,NR,NPDRSP,0)
C**** IF HAVE S & P GO TO 10, ELSE WRITE ERROR
          IF(NPDRSP .GT. 0)GO TO 10
          WRITE(6,15)NR,NATP
          15 FORMAT(' NO POWDER OF TYPE',I6,' AT ATP ', I2 )
          WRITE(LUOUT,15)NR,NATP
          RETURN
C
C**** HAVE S & P. IF INSUFFICIENT AMMO,GO TO 20
          10 MIX = ITRUCK(NPDRSP,5)

```

```

      IF(NRNTK .LT. NPCND)THEN
C**** INSUFFICIENT AMMO
      ITRUCK(NPDRSP,6) = 0
      CALL PUTQUE(NPDRSP,NQUE)
C      DECREMENT ROUNDS NEEDED
      NPCND = NPCND - NRNTK
C      GO GET ANOTHER TRUCK
      GO TO 5
    END IF
C**** SUFFICIENT AMMO, OFFLOAD AND PUT BACK IN QUEUE
    ITRUCK(NPDRSP,6) = (NRNTK - NPCND) * 10000 / IMIX(MIX,NR)
    CALL PUTQUE (NPDRSP, NQUE)
C**** DECREMENT AMMO ON HAND AND DEMAND IN SUBROUTINE ATP
C**** DECREMENT FUZES FROM S & P
    NFZQ = IQ(2,NATP)
    CALL FINTK(NFZQ,20,NFZSP,0)
    IF(NFZSP .EQ. 0)THEN
      WRITE(6,200)NATP,TIME
200   FORMAT(' NO S & P WITH FUZES, ATP #',I2,', TIME',F8.2)
      WRITE(LUOUT,200)NATP,TIME
      RETURN
    END IF
    NPRCNT = NRNDS * 10000/9000.
    ITRUCK(NFZSP,6) = ITRUCK(NFZSP,6) - NPRCNT
    IF(ITRUCK(NFZSP,6) .LE. 0)ITRUCK(NFZSP,6) = 0
    CALL PUTQUE(NFZSP,NFZQ)
C
    RETURN
  END

```

```

dd.  SUBROUTINE LOOKEV
      SUBROUTINE LOOKEV (KIND, IPARM, EVTIME, IGET)
C     LOOKEV WRITES EVENTS ON THE AUDIT TRAIL.
C     DISPLAY OF AN EVENT TYPE MAY BE SUPPRESSED BY SETTING
C       LOOK(EVENT TYPE) = 0
C
C       IF IGET = 0   THIS MEANS THE EVENT IS BEING SCHEDULED NOW
C       IF IGET = 1   THE EVENT WILL BE EXECUTED NOW
C
C     CALLED BY MAINARM
C     CALLS      NOTHING
C
C**** H. JONES      MAR 79
C
C     DIMENSION IPARM(5)
C     CHARACTER*10 NAME(19),N
C
C     INCLUDE LOG,LIST
C     DATA NAME /'DEMAND', 'RELOAD', 'UNTDPE', 'ATPARV', 'ASPARV',
C     $ 'ATP   ', 'ASP   ', 'UNTARV', 'CSAARV', 'ATPAR1', 'ATPAR2',
C     $ 'ASPAR1', 'ASPAR2', 'HELARV', 'HASPAR', 'CSADEP', 'REPORT',
C     $ 'CONTRL', 'ENDSIM'/
C
C     IF(LOOK(KIND) .EQ. 0) RETURN
C     IF((KIND.EQ.10.OR.KIND.EQ.11).AND.IPARM(2).EQ.0) RETURN
C     N = NAME(KIND)
C     IF(IGET .EQ. 1) WRITE(LUOUT,10) N, IPARM, EVTIME
10  FORMAT(1X,A10,',', PARS = ',5I6,',', TIME= ',F8.1)
C
C     IF(IGET .EQ. 0) WRITE(LUOUT,30) N, IPARM, EVTIME
30  FORMAT(1X,A10,',', PARS = ',5I6,',', SCHED TIME= ',F8.1)
C     IF(KIND .EQ. 14 .AND. IGET .EQ. 1)WRITE(19,40)N,IPARM,EVTIME
40  FORMAT(35X,A10,',', PARS = ',5I6,',', TIME= ',F8.1)
C
C     RETURN
C     END

```

```

ee.      SUBROUTINE NEXTEV
          SUBROUTINE NEXTEV (ITYPE, IPARM, EVTIME)
C**** INTERFACE ROUTINE TO GET NEXT EVENT
C        CALLED FROM MAINARM
C        CALLS GETEVT
C**** H. JONES    DEC 78
C
          DIMENSION IPARM(5)
          CALL GETEVT (IPARM, ITH, ITS)
C
C          ITH -- EVENT TIME IN WHOLE MINUTES
C          ITS -- EVENT TIME (FRACTIONAL) * 3600.
          EVTIME = ITH + ITS / 3600.
          ITYPE = IPARM(5)
C
          RETURN
          END

```

```

ff.      SUBROUTINE NXTQUE
          SUBROUTINE NXTQUE (ITEM,NUMQUE)
C**** SHOWS NEXT ITEM IN VEHICLE QUEUE(LEAVES IT IN)
C
C      CALLED(IN EDIT PROGRAM) BY  PRINT, TRKPUT
C
C**** H.JONES   FEB 79
          INCLUDE QUENUM,LIST
          INCLUDE QUEPNT,LIST
          ITEM = 0
          LITEM = 0
          IPOINT = IHEAD(NUMQUE)

C
C      SEARCH DOWN QUEUE TO FRONT
10 IF(IPOINT .EQ. 0) GO TO 20
          LITEM = ITEM
          ITEM = IPOINT
          IPOINT = ITEMS(ITEM)
          GO TO 10

C
20 RETURN
      END

```

```

99.      SUBROUTINE OPERA
          SUBROUTINE OPERA (NUMTK, TVLTIM, DELAY)
C**** CALCULATES DELAY DUE TO RELIABILITY FAILURE
C      OPERA RE-INITIALIZES THE TIME TO NEXT FAIL (ITRUCK(N,7))
C      TO A NUMBER EXPONENTIALLY DISTRIBUTED ABOUT THE MEAN TIME
C      TO FAIL(ITYPE(TYPE,5)) IF THE TRUCK FAILS.
C
C      IF THE TRUCK DOES NOT FAIL, THE RUN TIME FOR THIS TRIP
C      IS DECREMENTED FROM THE TIME TO NEXT FAIL(ITRUCK(N,7)).
C      EACH TRUCK HAS CLOCK OF TIME SINCE LAST FAILURE.
C
C      CALLED BY ASP, ASPAR1, ASPAR2, ASPCK, ATP, ATPAR1, ATPAR2, ATPARV,
C      CSADEP, DEMAND, HELARV, RELOAD, SERVER, UNTOEP
C
C**** H. JONES      JAN 79
C
C**** LOCAL VARIABLES
C      DELAY - TIME LOST DUE TO REMEDIAL MAINTENANCE
C      MTBF - MEAN TIME BETWEEN FAILURES FROM ITYPE
C      TLEFT - TIME LEFT UNTIL NEXT FAILURE AFTER THIS MOVE
C      TVLTIM - TIME LENGTH OF THIS MOVE
C
C      INCLUDE LOG
C      DELAY = 0.
C
C**** COMPARE TRUCKS REMAINING TIME BEFORE FAILURE WITH TRANSIT TIME.
C      ITKTYP = ITRUCK(NUMTK, 1)
C      TLEFT = ITRUCK(NUMTK,7) - TVLTIM
C      IF(TLEFT .GT. 0.) THEN
C          ITRUCK(NUMTK,7) = TLEFT
C          RETURN
C      END IF
C
C**** FAILURE OCCURS THIS TRANSIT
C      TIME TO REPAIR IS LOG NORMALLY DISTRIBUTED
C      XMTTR = ITYPE(ITKTYP, 6)
C      XMU = .5 * ALOG((XMTTR * XMTTR) / 2.)
C      SIGMA = SQRT(ALOG((XMTTR * XMTTR) / (XMTTR * XMTTR) + 1.))
C      U1 = RANF(DUM)
C      U2 = RANF(DUM)
C      X1 = ((-2 * ALOG(U1)) **.5) * COS(2 * 3.14159 * U2)
C      DELAY = EXP(SIGMA * X1 + XMU)
C      IF(DELAY .LT. 60.) DELAY = 60.
C      IF(ITRUCK(NUMTK,3) .EQ. 0) DELAY = 60. @ RELOAD FAILURE
C**** COMPUTE TIME TO NEXT FAILURE(EXPONENTIAL DISTRIBUTION)
C      XMTBF = ITYPE(ITKTYP,5)
C      UNRN = RANF(DUM)
C      ITRUCK(NUMTK,7) = (-XMTBF*ALOG(1-UNRN))
C
C**** WRITE MESSG FOR LOST TRUCK
C      WRITE(LHOUT,5) NUMTK,TIME,INT(DELAY)
5      FORMAT(' TRUCK NUMBER',IS,' FAILED AT ',F8.1,' FOR',IS,' MINUTES')
C      ITRUCK(NUMTK,3) = 6
C      ITRUCK(NUMTK,8) = ITRUCK(NUMTK,8) + 1      @ FAILURE COUNTER

```

RETURN
END

```

hh.      SUBROUTINE PUTEVT
          SUBROUTINE PUTEVT(IEVT, ITH, ITS, ICHECK)
C
C PUTEVT PLACES AN EVENT RECORD IN THE QUEUE IN CHRONOLOGICAL
C ORDER AND UPDATES THE QUEUE DIRECTORY. ICHECK FLAG SET
C IF INSERT WAS UNSUCCESSFUL.
C
C A LINEAR SEARCH IS DONE, EITHER FROM THE BACK FORWARD
C OR FROM THE FRONT BACKWARD(WHICH EVER IS CLOSER TO THE
C EVENT TIME), TO INSERT THE EVENT.
C
C      BOB DAVISON      1978
C
C      INCLUDE EVENTS,LIST
C      DIMENSION IEVT(5),JFORE(2048),JBACK(2048),JTIME(2048,2)
C      EQUIVALENCE (JFORE(1),JEVDS(1,1)),(JBAC(1),JEVDS(1,2)),
C      Z (JTIME(1,1),JEVDS(1,3)),(JSTAT(1),JFIRST),(JSTAT(2),JLAST),
C      Z (JSTAT(3),JEMPTY),(JSTAT(4),NUMEVT),(JSTAT(5),NEMPTY),
C      Z (JSTAT(6),MAXEVT)
C CHECK IF SPACE AVAILABLE .. IF NONE, RETURN
C      ICHECK = 2048 - NEMPTY
C      IF(NEMPTY.LE.0) GOTO 400                      @ RETURN
C      ICHECK=0
C      LSAVE=JFORE(JEMPTY)
C PUT EVENT RECORD IEVT IN IEVS
C      DO 20 IN = 1,5
C          IEVS(IN,JEMPTY) = IEVT(IN)
C      20 CONTINUE
C IF NO EVENTS IN QUEUE, PERFORM THE FOLLOWING
C      IF(NUMEVT.GE.1) GOTO 200
C      JFORE(JEMPTY)=0
C      JBACK(JEMPTY)=0
C      JFIRST=JEMPTY
C      JLAST=JEMPTY
C      GOTO 380
C IF ONE EVENT IN QUEUE, PERFORM THE FOLLOWING
C      200 CONTINUE
C      ITFH=JTIME(JFIRST,1)
C      ITFS=JTIME(JFIRST,2)
C      IF(NUMEVT.GT.1) GOTO 300
C IF LOWEST TIME EVENT, PERFORM THE FOLLOWING
C      IF(ITH.GT.ITFH)GO TO 210
C      IF(ITH.EQ.ITFH.AND.ITS.GE.ITFS)GO TO 210
C      JFORE(JEMPTY)=JFIRST
C      JBACK(JEMPTY)=0
C      JBACK(JFIRST)=JEMPTY
C      JLAST=JFIRST
C      JFIRST=JEMPTY
C      GOTO 380
C
C ELSE THIS TIME IS EQUAL TO OR LATER THAN THE LAST EVFNT
C      210 CONTINUE
C      JFORE(JEMPTY)=0
C      JBACK(JEMPTY)=JFIRST

```

```

        JLAST=JEMPTY
        GOTO 380
C
C IF TWO OR MORE EVENTS IN QUEUE, PERFORM THE FOLLOWING
300 CONTINUE
C IF EVENT TIME IS LESS THAN FIRST EVENT, MAKE IEVT THE FIRST EVENT
    IF(ITH.GT.ITFH)GO TO 310
    IF(ITH.EQ.ITFH.AND.ITS.GE.ITFS)GO TO 310
    JFORE(JEMPTY)=JFIRST
    JBACK(JEMPTY)=0
    JBACK(JFIRST)=JEMPTY
    JFIRST=JEMPTY
    GOTO 390
C
C IF EVENT TIME IS GREATER THAN OR EQUAL TO LAST EVENT, MAKE IEVT LAST
310 CONTINUE
    ITLH=JTIME(JLAST,1)
    ITLS=JTIME(JLAST,2)
    IF(ITH.LT.ITLH)GO TO 320
    IF(ITH.EQ.ITLH.AND.ITS.LT.ITLS)GO TO 320
    JFORE(JEMPTY)=0
    JBACK(JEMPTY)=JLAST
    JFORE(JLAST)=JEMPTY
    JLAST=JEMPTY
    GOTO 380
C
C EVENT TIME IS BETWEEN JTIME(JFIRST) AND JTIME(JLAST)
320 CONTINUE
    NUM=NUMEVT-1
C IF EVENT TIME CLOSER TO FIRST, START SEARCH AT FIRST EVENT
    IF((ITH-ITFH)-(ITLH-ITH))326,325,350
325 IF((ITS-ITFS)-(ITLS-ITS))326,326,350
326 IND1=JFIRST
    IT1H=ITFH
    IT1S=ITFS
    IND2=JFORE(JFIRST)
    IT2H=JTIME(IND2,1)
    IT2S=JTIME(IND2,2)
    DO 330 I=1,NUM
        IF(ITH.GT.IT2H)GO TO 327
        IF(ITH.EQ.IT2H.AND.ITS.GE.IT2S)GO TO 327
        GO TO 340
327 IND1=IND2
    IT1H=IT2H
    IT1S=IT2S
    IND2=JFORE(IND2)
    IT2H=JTIME(IND2,1)
    IT2S=JTIME(IND2,2)
330 CONTINUE
    ICHECK=2
    GOTO 400
340 JFORE(IND1)=JEMPTY
    JBACK(JEMPTY)=IND1
    JFORE(JEMPTY)=IND2
    @ RETURN

```

```

      GOTO 380
C
C EVENT TIME CLOSER TO LAST, START SEARCH AT LAST EVENT
350  IND1=JLAST
      IT1H=ITLH
      IT1S=ITLS
      IND2=JBACK(JLAST)
      IT2H=JTIME(IND2,1)
      IT2S=JTIME(IND2,2)
      DO 360 I=1,NUM
          IF(ITH.LT.IT2H)GO TO 355
          IF(ITH.EQ.IT2H.AND.ITS.LT.IT2S)GO TO 355
          GO TO 370
355  IND1=IND2
      IT1H=IT2H
      IT1S=IT2S
      IND2=JBACK(IND2)
      IT2H=JTIME(IND2,1)
      IT2S=JTIME(IND2,2)
360  CONTINUE
      ICHECK=2
      GOTO 400
                                     @ RETURN
370  JFORE(IND2)=JEMPTY
      JBACK(JEMPTY)=IND2
      JFORE(JEMPTY)=IND1
      JBACK(IND1)=JEMPTY
C
C PERFORM THE FOLLOWING FOR ALL EVENTS
380  CONTINUE
      JTIME(JEMPTY,1)=ITH
      JTIME(JEMPTY,2)=ITS
      NUMEVT=NUMEVT+1
      NEMPTY=NEMPTY-1
      JEMPTY=LSAVE
C
400  RETURN
      END

```

```

ii.      SUBROUTINE PUTQUE
          SUBROUTINE PUTQUE (ITEM, NUMQUE)
C**** PUTS ITEM IN QUEUE NUMQUE
C
C      CALLED BY ASPAR1, ASPAR2, ASPARV, ATPAR1, ATPAR2, ATPARV, CSAARV,
C      DEPASP, FINTK, LDPWR, SERVER, UNTARV
C      (IN EDIT PROGRAM: EDIT, PRINT, TRKPUT)
C**** H. JONES      DEC 78
          INCLUDE LOG,LIST
          INCLUDE QUENUM,LIST
          INCLUDE QUEPNT,LIST
          IF(NUMQUE.LT.1.OR.NUMQUE.GT.176)THEN
              PRINT*, ' BAD QUEUE# IN PUTQUE...ITEM NOT PUT IN QUEUE'
              RETURN
          END IF
C
C ** LOOP FOR ALL QUEUES
DO 40 I = 1,176
    IPOINT = IHEAD(I)
C ** LOOP FOR ALL ITEMS IN QUEUE
    20 CONTINUE
    IF(IPOINT .EQ. 0)THEN
C      ** EXIT LOOP
        GO TO 30
    END IF
    IF(IPOINT .EQ. ITEM)THEN
        WRITE(6,100)ITEM,I,TIME
    100 FORMAT(' TRUCK',I5,' IS ALREADY IN QUEUE',I4,' AT TIME',F8.1)
        RETURN
    END IF
    IPOINT = ITEMS(IPOINT)
    GO TO 20
    30 CONTINUE
C ** END LOOP
    40 CONTINUE
C ** END LOOP
C
    IOLDH = IHEAD(NUMQUE)
    IHEAD(NUMQUE) = ITEM
    ITEMS(ITEM) = IOLDH
C
    RETURN
END

```

jj.

SUBROUTINE QINIT
SUBROUTINE QINIT

```
C
C  INITIALIZE THE EVENT QUEUE(EMPTY) AT THE START
C  OF FIRST CI.
C  CALLED FROM MAINARM
C
C  BOB DAVISON
C
C  INCLUDE EVENTS,LIST
C  DIMENSION JFORE(2048),JBACK(2048),JTIME(2048,2)
C  EQUIVALENCE (JFORE(1),JEVDS(1,1)),(JBAC(1),JEVDS(1,2)),
Z (JTIME(1,1),JEVDS(1,3)),(JFIRST,JSTAT(1)),(JLAST,JSTAT(2)),
Z (JEMPTY,JSTAT(3)),(NUMEVT,JSTAT(4)),(NEMPTY,JSTAT(5)),
Z (MAXEVT,JSTAT(6)),(JTIME(1,2),JEVDS(1,4))
C
C  NUMEVT=0
C  NEMPTY = 2048
C  JFIRST=0
C  JLAST=0
C  JEMPTY=1
C  DO 100 I=1,NEMPTY
C      JFORE(I)=I+1
C      JBACK(I)=I-1
C      JTIME(I,1)=0
C      JTIME(I,2)=0
100 CONTINUE
C  JFORE(NEMPTY)=0
C  JBACK(1)=0
C  PRINT*, ' QINIT'
C
C  RETURN
C  END
```

```

kk.  SUBROUTINE RDIEXO
      SUBROUTINE RDIEXO(NUNIT)

C
C      UPDATES IUNIT EACH PULSE OF A MULTI-DEMAND AND SCHED DEMAND
C
C      CALLED BY DEMAND
C      CALLS      SCHED
C      SCHEDULES DEMAND (FOR NEXT PULSE)
C
C      JIM FOX - FEB 1979
C      INCLUDE LOG,LIST
C
C      LOCAL VARIABLES
C      NUNIT - UNIT NUMBER
C      NCELLS - NUMBER OF DEMAND PULSES IN DEMAND UNIT RECORD
C      NMDEAD - NUMBER OF WEAPONS KILLED
C      NDEDRD - NUMBER OF DEAD ROUNDS, LOST WHEN WPN KILLED
C      NMRD - NUMBER OF ROUNDS
C      NUMPL - NUMBER OF THE PULSE
C      NCELLS - NUMBER OF PULSES PER CI
C
      DIMENSION IPARM(5)
      DO 5 I = 1,5
        IPARM(I) = 0
5  CONTINUE
C      SET IPARM TO CALL TO SCHEDULE DEMAND
      IPARM(1) = NUNIT
C      FIND NUMBER OF DEMAND PULSE CELLS
      NCELLS = IUNIT(NUNIT,139)
      IF(NCELLS.LE.1) GO TO 10
C      COMPUTE THE TIME OF THE NEXT DEMAND EVENT AND SCHEDULE IT.
      TOTIM = TIME + TCILNG / NCELLS
      IF(TOTIM .GT. TCIST + TCILNG)GO TO 10
      CALL SCHED(1,IPARM,TOTIM )          @ DEMAND
10  CONTINUE
      IF(NCELLS.LE.1) NCELLS=1
C      UPDATE IUNIT WITH A PART OF THE DEMAND DATA
C
      DO 100 I = 1,LPPAR(6)
        IND = I * 13 - 5
C      IF NO DATA TO UPDATE GO TO 100
        IF(IUNIT(NUNIT,IND+10) .LE. 0)GO TO 100
C      COMPUTE WHICH PULSE THAT THIS UPDATE REPRESENTS
        NUMPL = (TIME - TCIST) / (TCILNG / NCELLS) + .5
        IF(NCELLS.LE.1) NUMPL=1
C      COMPUTE NEGATIVE SURVIVOR FACTOR
C      LOWER NUMBER OF SURVIVORS FOR THIS PULSE
C      COMPUTE THE NUMBER OF DEAD TO BE ASSFSED THIS PULSE - NMDEAD
        NMDEAD = (IUNIT(NUNIT,IND+9)+NUMPL-1) / NCELLS
C
C      COMPUTE NUMBER OF RND5 LOST WITH DEAD WPN
        IF(IUNIT(NUNIT,1) .EQ. 8)THEN
          NDEDRD = IUNIT(NUNIT,IND + 7)*NMDEAD
        ELSE

```

```

      END IF
C      IF NEG. ROUNDS ON HAND - NONE LOST.
      IF(IUNIT(NUNIT,IND+4) .IE. 0) NDEDRD = 0
      IUNIT(NUNIT,IND+1) = IUNIT(NUNIT,IND+1) - NMDEAD
      IF(IUNIT(NUNIT,IND+1).LT.0) IUNIT(NUNIT,IND+1)=0
C**** ASSUME SINGLE PULSE UNIT
      IUNIT(NUNIT,IND+2)=IUNIT(NUNIT,IND+10)
C**** CHECK FOR ARTY UNITS
      IF(IUNIT(NUNIT,1).GE.4.AND.IUNIT(NUNIT,1).LE.6) IUNIT(NUNIT,IND+2)
Z = IUNIT(NUNIT,IND+1)
C      IF FARP, NUMBER OF WEAPONS SHORT AMMO = NUM IN CELL
      IF(IUNIT(NUNIT,1) .EQ. 8) IUNIT(NUNIT,IND+2) = IUNIT(NUNIT,IND+2)
Z + (IUNIT(NUNIT,IND+10) + NUMPL - 1) / NCELLS
C
C      UPDATE A PORTION OF ROUNDS SHORT
C      COMPUTE THE NUMBER OF ROUNDS SHORT TO BE ASSESSED THIS PULSE-NMRD
      NMRD = (IUNIT(NUNIT,IND+11) + NUMPL - 1) / NCELLS
      IUNIT(NUNIT,IND+3)=IUNIT(NUNIT,IND+3)+NMRD-(IUNIT(NUNIT,IND+7)
Z *NMDEAD - NDEDRD)
C      COMPUTE AMMO ON HAND
      IUNIT(NUNIT,IND+4) = IUNIT(NUNIT,IND+4) - (NDEDRD + NMRD)
100 CONTINUE
C
      RETURN
      END

```

11.

SUBROUTINE ROUTINE

READS INPUT FILE CREATED BY JIFFY,
TRANSLATES THE JIFFY LOG TO ARM NUMBERS
SCHEDULES A DEMAND EVENT FOR EACH UNIT FIRING ARM,
CREATES UNIT FOR SINGLE PULSE DEMAND LOGS.

ANN HILLS-APRIL 1983

LOCAL VARIABLE DEFINITION

18016 = JIFFY LOGGED INTERFACE REC 64 AS LONG REC REC
1 = JIFFY UNIT ID
2 = NUMBER OF AR IN BELLAR ONLY
3 = ARM AMMO CODE
4 = NUMBER OF WEAPONS ALIVE
5 = NUMBER OF WEAPONS SHORT ARM
6 = TOTAL NUMBER OF ROUNDS SHORT
7-11 NOT REPEAT OF 2-6
18017-RECORD NUMBER OF FILE 9 EQUIVALENT TO ARM UNIT NO.
IAIN = LOOP INDEX
I201 = COMPUTED INDEX TO ACCESS UNIT AMMO TYPE
IAMMO = AMMO TYPE FROM UNIT
I18 = LOOP INDEX
IAM = 18018 AMMO TYPE
I = LOOP INDEX
I11 = LOOP INDEX
I111 = LOOP INDEX
I1111 = LOOP INDEX
I11111 = LOOP INDEX
I111111 = LOOP INDEX
I1111111 = LOOP INDEX
I201 = INDEX COMPUTED FROM IAIN TO ACCESS UNIT PER AMMO TYPE
IA = LOOP INDEX
IAMM = AMMO TYPE FROM 18018
IAM = AMMO TYPE FROM 18018
I201 = COMPUTED INDEX FROM I11 TO ACCESS UNIT
IAM = AMMO TYPE FROM UNIT
NHEL1 = ACCUMULATOR FOR NUMBER OF HELICOPTERS IN BELL
N1 = INDEX TO ACCESS 18019,7,12 REC
NWE1 = ACCUMULATOR FOR NUMBER OF WEAPONS SHORT
NAPAL = ACCUMULATOR FOR NUMBER OF AR RETURNING ALIVE
NAPASH = ACCUMULATOR FOR NUMBER OF AR SHORT ROUNDS
NDELL = NUMBER OF BELL IN FAPP RECORD
DELTIM = TIME INTERVAL BETWEEN FAPP BELL RECORDS
I11 = COMPUTED INDEX TO FILE TEXT
I111 = COMPUTED INDEX TO FIND 18019 TO BUILD UNIT RECORD
TOTTIM = TIME TO SCHEDULE EVENT

include 180118
DIMENSION 18018(20),18019(5)
OPEN FILE 9(75,20,1,17)
18011 = 1
18012 = 0

10 GO TO 10

0*4*4* ZERO LAST EVENT'S DEMAND, WEAPONS KILLED, AND WEAPONS SHORT


```

        IPRM(1)=IRED
        TOTTIM=TIME+ADDTIM
        CALL SCHED(1,IPRM,TOTTIM)
    END IF
*****THIS ENDS RECORD,GET NEXT RECORD
0
500  CONTINUE
    260 FORMAT(' HAVE FINISHED SCHEDULE ')
    RETURN

END MAIN PROGRAM

SUBROUTINE SNERD(IFF,ADDTIM)

    THIS SUBROUTINE READS IN DEMAND FOR SINGLE PULSE UNITS
    IFF IS A FLAG USED IN MAIN PROGRAM TO DETERMINE WHETHER
    A DEMAND EVENT SHOULD BE SCHEDULED(IFF=0 MEANS NO.....
    ADDTIM CARRIES THE TIME FOR SCHEDULING DEMAND.

    IFF=0
    120 DO 160 IAIN = 1,LPBAR(4)  SLOOP FOR UNIT AMMO CODES
        IND1 = 12 * IAIN - 5
        IAMMO = IUNIT(IRED,IND1)
        IF(IAMMO.EQ. 0)GO TO 160
        DO 140 IIF = 1,30,5  SLOOP FOR DEMAND FILE AMMO CODES
            IJAM = IROUF(IIF)

            SCENARIO DEPENDENT CODE TO READ IN SECOND AMMO CODE 2
            IF(IJAM.EQ. 25 .AND. IAIN.EQ. 0)GO TO 160

            IF THIS AMMO CODE DOES NOT MATCH UNIT,TRY NEXT ONE
            IF(IJAM.NE. IAMMO .OR. IJAM.EQ. 0)GO TO 140
            HAVE EQUAL AMMO TYPES, UPDATE UNIT WITH NEW DATA.
            *****
            UPDATE FOR DEMAND DATA IN UNIT STATUS REPORT
            130  IUNIT(IRED,IND1+9) = IUNIT(IRED,IND1+1) - IROUF(IIF+1)
            IF(IUNIT(IRED,IND1+1).LT.IROUF(IIF+1))IUNIT(IRED,IND1+1)=
            IROUF(IIF+1)
            IF(IUNIT(IRED,IND1+9).LT. 0)IUNIT(IRED,IND1+9) = 0
            IUNIT(IRED,IND1+10) = IROUF(IIF+2)
            IUNIT(IRED,IND1+11) = IROUF(IIF+4)
            IROUF(IIF) = 0
            IFF=1

            END OF UNIT UPDATE FOR THIS AMMO TYPE
            GO TO 160
        140  CONTINUE
    1  GO UNIT AMMO MATCH
        WRITE(11,OUT,150) IRED,IJAM

```



```

      IFF=1
      RETURN
200 CONTINUE

      SUBROUTINE ARTYRD(IFF,ADDTIM)

      THIS SUBROUTINE MODIFIES IUNIT ARRAYS FOR ARTILLERY
      UNITS AND SETS FLAG FOR DEMAND EVENT, AND SETS ADDTIM
      TO SHOW TIME TO SCHEDULE THE EVENT

210 ADDTIM = TCIST + 60.
      IFF=1
      FIND AMMO TYPES TO UPDATE IUNIT HOLDING FOR ARTY
      IUNIT(IREC,139) = INT(TCILNG / 60. + .5)
      DO 240 IA = 1,30,5
      SELECT AMMO RECORD FROM IRDJF
      IAMM = IRDJF(IA)      @THIS IS AMMO TYPE FROM DEMAND FILE
      IF(IAMM.EQ.0)GO TO 240      @NO AMMO TYPE GET NEXT ONE
      FIND CORRESPONDING UNIT AMMO
      DO 220 IU = 1,LPPAR(6)
      IUA = 13 * IU - 5
      IUAM = IUNIT(IREC,IUA)      @THIS IS AMMO TYPE FROM IUNIT ARR
      IF(IUAM.EQ.0)GO TO 220      @AMMO TYPE 0 GET NEXT ONE
      IF NOT THE SAME AMMO GO TO 220
      IF(IUAM .NE. IAMM)GO TO 220
      HAVE AMMO MATCH.  SET UP FILE IUNIT.
      IUNIT(IREC,IUA+9) = IUNIT(IREC,IUA+1) - IRDJF(IA+1)
      IF(IUNIT(IREC,IUA+1).LT.IRDJF(IA+1))IUNIT(IREC,IUA+1) =
      $      IRDJF(IA+1)
      IF(IUNIT(IREC,IUA+9) .LT. 0)IUNIT(IREC,IUA+9) = 0
      IUNIT(IREC,IUA+10) = IRDJF(IA+2)
      IUNIT(IREC,IUA+11) = IRDJF(IA+3)
      GO TO 240
220 CONTINUE
      WRITE (LUCUT,230)IREC,IA
230 FORMAT(' NO IUNIT AMM MATCH - RDJIFF, UNIT ',I5,' AMMO ',I5)
      IFF=0
      RETURN
240 CONTINUE
      RETURN

      END SUBROUTINE ARTYRD
END

```

```

mm.  SUBROUTINE READF
      SUBROUTINE READF (LU, NUM, INTGR, REAL, IWORD)
C**** RETURNS UP TO NUM INTEGERS, REALS, AND STRINGS.
C**** BLANKS AND COMMAS ARE DELIMITERS.  READS FROM TERMINAL.
C**** H. JONES      1979
C
      CHARACTER*(*) IWORD
      CHARACTER*1 IBLANK,IPERD,ICOMMA,IMINUS,IQUOT,IALDIG,ICHR
      DIMENSION INTGR(NUM), REAL(NUM), IWORD(NUM)
      DIMENSION ICHR(82), IALDIG(10)
      DATA IBLANK /' ',IPERD /'.',ICOMMA /',',IMINUS /'-'/
      DATA IQUOT /'"/
      DATA IALDIG /'1','2','3','4','5','6','7','8','9','0'/
      ICHR(81) = IBLANK
      ICHR(82) = IQUOT
C
C**** READ RECORD, ZERO OUT OLD INTGR, REAL, IWORD
      READ(LU,210,END= 190) (ICHR(I), I=1,80)
      DO 10 I=1,NUM
         INTGR(I)=0
         REAL(I)=0.
      10 IWORD(I) = IBLANK
         KWORD=0
         KINTGR=0
         KREAL=0
         N=0
C
C**** CHECK NEXT CHARACTER IN RECORD
C**** SKIPPING BLANKS *****
      20 MINUS = 1
      30 N=N+1
         IF(N.EQ.81) GO TO 190
         IF(ICHR(N).EQ.IBLANK) GO TO 30
C
C**** DETERMINE IF CHAR IS NUMBER OR ALPHA
         IF(ICHR(N).EQ.IQUOT) GO TO 160
         IF(ICHR(N).NE.IMINUS) GO TO 40
         MINUS = -1
         GO TO 30
      40 ISTART = N
         NUMB=0
         IF(ICHR(N).EQ.IPERD) GO TO 90
         DO 50 I=1,10
            IF(ICHR(N).EQ.IALDIG(I)) GO TO 60
         50 CONTINUE
            GO TO 150
C
C**** BUILDING INTEGER OR INTEGER PART OF REAL
      60 N=N+1
         IF(ICHR(N).NE.IBLANK .AND. ICHR(N).NE.IPERD
           & .AND. ICHR(N).NE.ICOMMA ) GO TO 60
C
C**** CALCULATE VALUE OF INTEGER
         IEND = N-1

```

```

      DO 80 I=ISTART,IEND
        DO 70 J=1,9
          IF(ICHR(I) .EQ. IALDIG(J)) GO TO 80
70    CONTINUE
        J=0
80    NUMB = NUMB + J * 10 ** (IEND-I)
        IF(ICHR(N) .EQ. IPERD) GO TO 90
C
C**** NUMBER WAS INTEGER, STORE IT, CHECK FOR BLANKS
      KINTGR = KINTGR + 1
      INTGR(KINTGR) = NUMB * MINUS
      GO TO 20
C
C**** NUMBER WAS INTEGER PART OF REAL, NOW BUILD DECIMAL.
80    RNUMB = FLOAT(NUMB)
      ISTART = N+1
      IF(ICHR(ISTART) .EQ. IBLANK) GO TO 140
100   N=N+1
      IF(ICHR(N).NE.IBLANK .AND. ICHR(N).NE.ICOMMA ) GO TO 100
C
C**** CALCULATE VALUE OF DECIMAL
      IEND = N-1
      IDECPL = 1
      NUMB=0
      DO 130 I=ISTART,IEND
        DO 110 J=1,9
          IF(ICHR(I) .EQ. IALDIG(J)) GO TO 120
110    CONTINUE
        J=0
120    NUMB = NUMB + J * 10**(IEND-I)
130   IDECPL = IDECPL * 10
C
C**** ADD INTEGER AND DECIMAL
      DECML=FLOAT(NUMB)/IDECPL
      RNUMB = RNUMB + DECML
140   KREAL = KREAL + 1
      REAL(KREAL) = RNUMB * MINUS
      GO TO 20
C
C**** BUILDING STRING ALPHANUMERIC
150   N=N+1
      IF(ICHR(N).NE.IBLANK .AND. ICHR(N).NE.ICOMMA ) GO TO 150
      GO TO 180
160   ISTART = N+1
170   N=N+1
      IF(ICHR(N) .NE. IQUOT) GO TO 170
180   IEND = N-1
      KWORD = KWORD + 1
      LENSTR = IEND - ISTART + 1
      IF(LENSTR .GT. 10) LENSTR = 10
      ENCODE(LENSTR,200, IWORD(KWORD)) (ICHR(J), J=ISTART, IEND)
      GO TO 20
C
190   RETURN

```

210 FORMAT(80A1)
END

```

nn.      SUBROUTINE RELOAD
          SUBROUTINE RELOAD (IPARM)
C**** EVENT RELOAD -- REPLACES ROUNDS OF AMMO AT UNIT WEAPONS.
C      EVENT TYPE 2
C      CALLED BY MAINARM
C      CALLS      IQ, DUALMX, FINTK, INTRDK, SCHED, OPERA
C
C**** D. HILLIS      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- 0 IF SCHED FROM DEMAND, TRK# IF SCHED FROM UNTARV
C
C      SCHEDULES -- UNTDEP, DEPARTURE OF UNIT TRUCKS
C                  UNTARV, ARRIVAL OF TRUCKS AT UNIT.
C                  ASPARV, EMPTY TRUCKS -> ASP
C      SCHEDULED BY DEMAND OR UNTARV
C
C      RELOAD WILL OCCUR AT THE TASK FORCE LEVEL FOR MANEUVER UNITS,
C      BATTERY LEVEL FOR ARTILLERY UNITS AND ADA UNITS, AND AT
C      BATTALION FARRP'S. THE RELOAD WILL BE CALLED FROM THE DEMAND
C      EVENT.
C
C
C      AMMO WILL BE CONSOLIDATED ON TRUCKS AT UNIT. NO MORE THAN 1 TRUCK
C      PER UNIT (PER TYPE OF AMMO) WILL BE AT LESS THAN FULL LOAD WHILE
C      LOCATED AT THE UNIT. A 'SMALL LOAD' THRESHOLD MAY BE DEFINED BELOW
C      WHICH AMMO IS DUMPED TO GROUND TO ALLOW TRUCK TO RETURN TO ATP.
C
C**** LOCAL VARIABLE DEFINITION
C**** K - UNIT AMMO INDEX
C**** TOTIM - TIME OF SCHEDULED EVENT
C**** DELAY - TIME TO RELOAD WEAPONS AT THE UNIT
C**** LOAD - NUMBER OF ROUNDS ON THE TRUCK
C**** ND - AMMO DEMAND
C**** NUMTK - TRUCK NUMBER
C**** NEWLD - TRUCK LOAD ON AMMO DEMAND
C**** MX - AMMO MIX INDEX
C**** IAM - AMMO TYPE
C**** KIND - EVENT TYPE
C**** THIND - DELAY TIME DUE TO INTERDICTION
C**** NRPW - NUMBER OF ROUNDS PER WEAPON
C**** NW - NUMBER OF WEAPONS LOADED PER TRUCK
C**** ILOAD - PARTIAL LOAD
C**** IFLAG - 0 -FARP TRUCK AVAIL., 1 -NO FARP TRUCK AVAIL.
          INCLUDE LOG,LIST
          DIMENSION IPARM(5)
C
          NUNIT = IPARM(1)
          IUNITQ = IQ(1,NUNIT)
C**** SELECT AN AMMO TYPE
          DO 180 KK=1,LPPAR(6)
            K=13 * KK - 5
            IAM = IUNIT(NUNIT,K)
            IF(IAM .EQ.0) GO TO 180

```

```

      IF(IUNIT(NUNIT,K+8) .LE. 0)GO TO 180
C**** IF THE UNIT HAS MORTARS AND BUSHMASTERS USE RELOAD ROUTINE DUALMX
C**** !!!!! THE NEXT LINE ASSUMES THAT MORTARS AND BUSH. GO TOGETHER
      IF(IAM .EQ. 16) THEN
        CALL DUALMX(NUNIT,K,K+13)
        RETURN
      END IF
C
C**** CALCULATE AMMO DEMAND
10  ND = IUNIT(NUNIT,K+7) * IUNIT(NUNIT,K+1) - IUNIT(NUNIT,K+4)
    IF(IUNIT(NUNIT,1).EQ.8) ND=IUNIT(NUNIT,K+3)
    IF(IUNIT(NUNIT,1) .EQ. 8) GO TO 15
    IF(ND.LT.IUNIT(NUNIT,K+4) - IUNIT(NUNIT,K+2) * IUNIT(NUNIT,K+5))
      GO TO 180
    IF(IAM .EQ. 10)THEN
      ND = (ND / 6) * 6
      IUNIT(NUNIT,K + 3) = IUNIT(NUNIT,K + 3) - ND
    END IF
15  IF(ND .LE. 0)GO TO 180
C
      WRITE(LUOUT,20)IAM,ND
20  FORMAT(' RELOAD AMMO TYPE',2I5,' ROUNDS')
C      PULL TRUCK FROM QUEUE
30  CALL FINTK(IUNITQ,IAM,NUMTK,0)
    WRITE(LUOUT,40)NUMTK
40  FORMAT(' RELOAD AFTER FINTK ',I5)
    IF(IUNIT(NUNIT,1).EQ.8.AND.NUMTK.EQ.0) GO TO 180
    IF(IUNIT(NUNIT,1).EQ.8) GO TO 90
    IF(NUMTK .EQ. 0)GO TO 180
C**** CHECK FOR INTERDICTION
    MX=ITRUCK(NUMTK,5)
    NOFF = (IMIX(MX,IAM) * ITRUCK(NUMTK,6) + 9999)/10000
    CALL INTRDK(NUMTK,TMIND)
    IF(TMIND .EQ. 0)GO TO 90
C
C**** ADD ONE TO THE NUMBER OF TRUCKS KILLED DURING RELOAD
    JUNIT(IUNIT(NUNIT,1),23) = JUNIT(IUNIT(NUNIT,1),23) + 1
    TOTIM = TIME + TMIND
    IPARM(2) = NUMTK
    IPARM(3)=IUNIT(NUNIT,3)
    IPARM(4)=ITRUCK(NUMTK,5)
C**** SCHEDULE ASPARV FOR EMPTY TRUCK
    CALL SCHED(5,IPARM,TOTIM )
C
C**** DECREMENT UNIT AMMO ON TRUCKS
    IUNIT(NUNIT,K+8) = IUNIT(NUNIT,K+8) - NOFF
    IF (IUNIT(NUNIT,K+8).LT.0) IUNIT(NUNIT,K+8)=0
    ITRUCK(NUMTK,6) = 0
C**** THIS LOGIC IS HERE TO SOLVE THE PROBLEM OF WEAPON
C**** SYSTEMS HAVING DIFFERENT BASIC LOADS FOR THE SAME AMMO
    IF(IUNIT(NUNIT,1).NE.1 .AND. IUNIT(NUNIT,1).NE.2) GO TO 30
    IF(IUNIT(NUNIT,K).NE.2) GO TO 30
    DO 50 JJ=8,LFPAR(6) * 13 - 5,13
      IF(K.EQ.JJ) GO TO 50

```

```

50 CONTINUE
C   NO EQUAL AMMO FOUND FOR 2 IN THIS UNIT GO TO 30
    GO TO 30
C
60 IUNIT(NUNIT,JJ+8)=IUNIT(NUNIT,JJ+8)- NOFF
   IF(IUNIT(NUNIT,JJ+8).LT.0)IUNIT(NUNIT,JJ+8)=0
   GO TO 30
70 MX = ITRUCK(NUMTK,5)
   ITRUCK(NUMTK,3) = 0
   IPARM(2) = NUMTK
C**** CALCULATE THE TRUCK AMMO LOAD
   LOAD = (IMIX(MX,IAM) * ITRUCK(NUMTK,6) + 9999) / 10000
   NRPW = ND / IUNIT(NUNIT,K+2)
C**** CALCULATE THE NUMBER OF WEAPONS LOADED PER TRUCK
   NW = MIN0(LOAD / NRPW,IUNIT(NUNIT,K+2))
C**** CALCULATE THE TRAVEL TIME
   TVLTIM = 2 * IRSTME(IAM,3) + (NW + 1) * IRSTME(IAM,1)
   CALL OPERA(NUMTK,TVLTIM,TFAIL)
   IF(TFAIL .GT. 0)THEN
       IPARM(3) = 0
       IPARM(4) = 666
       CALL SCHED(8,IPARM,TIME + TFAIL)      @ UNTARV
       GO TO 30
   END IF
   DELAY = TVLTIM + IRSTME(IAM,2) * NRPW/100
C**** CHECK AMMO DEMAND AGAINST TRUCK LOAD
   IF(MX.EQ.9)GO TO 92
   IF(ND .LT. LOAD)THEN
C****   CALCULATE THE PARTIAL LOAD OF THE TRUCK
       ITRUCK(NUMTK,6) = 10000 * (LOAD - ND)/IMIX(MX,IAM)
       NEWLD = ND
       KIND = 8
       GO TO 39
   END IF
C
92 ITRUCK(NUMTK,6) = 0
   NEWLD = LOAD
   KIND = 3
   IF(MX.EQ.9)THEN
       DELAY = 20.
       NW = IUNIT(NUNIT,K+2)
   END IF
C
38 TOTIM = TIME + DELAY
C**** CHECK PARTIAL LOAD ON TRUCK
   IF (ITRUCK(NUMTK,6) .GT. 1000)GO TO 42
   IF(IUNIT(NUNIT,1) .EQ. 8 .AND. ITRUCK(NUMTK,6) .EQ. 0)GO TO 42
   IPLOAD = (IMIX(MX,IAM) * ITRUCK(NUMTK,6) + 9999)/10000
   IUNIT(NUNIT,K+4) = IUNIT(NUNIT,K+4) + IPLOAD
   IUNIT(NUNIT,K+8) = IUNIT(NUNIT,K+8) - IPLOAD
   IF(IUNIT(NUNIT,K+8).LT.0)IUNIT(NUNIT,K+8)=0
   KIND = 3
   ITRUCK(NUMTK,6) = 0
   IPARM(3) = 0

```

```

      TOTIM = TOTIM + (IRSTME(K,2) * IPLOAD/100)
C**** SCHEDULE A UNTARV OR UNTDEP DEPENDING ON VALUE OF KIND
      42 CALL SCHED(KIND,IPARM,TOTIM) @ UNTDEP/UNTARV
C
      ITRUCK(NUMTK,11) = ITRUCK(NUMTK,11) + 1 @ RELOAD COUNTER
C**** ADD THE DELAY TO THE TOTAL RELOAD TIME SPENT BY THIS TYPE UNIT
      JUNIT(IUNIT(NUNIT,1),21) = JUNIT(IUNIT(NUNIT,1),21)
      Z + (TOTIM - TIME)
C**** ADJUST AMMO ON TRUCKS AND CURRENT AMMO SUPPLY
      IUNIT(NUNIT,K+8) = IUNIT(NUNIT,K+8) - NEWLD
      IF(IUNIT(NUNIT,K+8).LT.0) IUNIT(NUNIT,K+8)=0
C**** THIS LOGIC IS HERE TO SOLVE THE PROBLEM OF WEAPON
C**** SYSTEMS HAVING DIFFERENT BASIC LOADS FOR THE SAME AMMO
      IF(IUNIT(NUNIT,1).NE.1 .AND. IUNIT(NUNIT,1).NE.2) GO TO 130
      IF(IUNIT(NUNIT,K).NE.2) GO TO 130
      DO 110 JJ=8,LPPAR(6) * 13 - 5,13
      IF(K.EQ.JJ) GO TO 110
      IF(IUNIT(NUNIT,K).EQ.IUNIT(NUNIT,JJ)) GO TO 120
110 CONTINUE
C
C      NO EQUAL AMMO 2 IN THIS UNIT GO TO 45
      GO TO 130
120 IUNIT(NUNIT,JJ+8)=IUNIT(NUNIT,JJ+8)-NEWLD
      IF (IUNIT(NUNIT,JJ+8).LT.0) IUNIT(NUNIT,JJ+8)=0
130 IUNIT(NUNIT,K+4) = IUNIT(NUNIT,K+4) + NEWLD
      IF(IUNIT(NUNIT,K+4) .LE. 0) WRITE(6,135) NUNIT, IUNIT(NUNIT,K),
      $ IUNIT(NUNIT,K+4), TIME
135 FORMAT(' UNIT',I3,' AMMO',I3,' CUR SUP',I10,' AT TIME',F8.1)
C      DECREMENT THE NUMBER OF ROUNDS SHORT
      IUNIT(NUNIT,K+3) = IUNIT(NUNIT,K+3) - NEWLD
      IF(IUNIT(NUNIT,K+3) .LE. 0) IUNIT(NUNIT,K+3) = 0
      IF(IUNIT(NUNIT,K+4).GE.IUNIT(NUNIT,K+1)*IUNIT(NUNIT,K+7))
      $ IUNIT(NUNIT,K+3) = 0
      IUNIT(NUNIT,K+2) = IUNIT(NUNIT,K+2) - NW
      GO TO 10
190 CONTINUE
C
      RETURN
      END

```

```

00.      SUBROUTINE REPORT
          subroutine report(iparm)
C**** WRITES REPORTS OF VARIOUS TYPES.
C      EVENT TYPE 17
C      CALLED BY MAINARM, CONTRL
C      CALLS      TRUCK
C**** J FOX      FEB 79
          CHARACTER*10 AUNIT,IWPN, NAME, ISAVE
          include los,LIST
          include aunit,LIST
          DIMENSION IPARM(5),IWPN(22),Jtrks(8),IRTYPE(7)
C
          DATA IWPN/'TANK','TOW','PWDR','155HE','155ICH','8inHE',
            $ '8inICH','8inPWDR','HELLFIRE','MLRS','155RAP','155CLUP','155RMK',
            $ '30mm(AAH)','8inRAP','MORTAR','BUSHMASTER','DIVAD','SMALL ARMS',
            $ 'FUZES','155GB','8INGB'/
          DATA IRTYPE /1,1,1,1,1,1,1/
15      CONTINUE
10      FORMAT (' 1)  UNIT STATUS',/,
Z        ' 2)  ATP STATUS',/,
Z        ' 3)  ASP STATUS',/,
Z        ' 4)  CSA STATUS',/,
Z        ' 5)  ATP AMMO ISSUED',/,
Z        ' 6)  ASP AMMO ISSUED',/,
Z        ' 7)  TRUCK MOVEMENT ',/,
Z        ' 8)  DEFAULT ',/,
Z        ' 9)  ALL')
          NASP = 0
          NATP = 0
          IRPT = IPARM(1)
          IF (IRPT .LT. 1 .OR. IRPT .GT. 9) THEN
              WRITE(6,10)
              READ*,IRPT
          END IF
          IF (IRPT .EQ. 0) RETURN
          IF (IRPT .EQ. 9) THEN
              DO 20 I = 1,7
20          IRTYPE(I) = 1
          END IF
          IF (IRPT .LE. 7) THEN
              DO 30 I = 1,7
30          IRTYPE (I) = 0
              IRTYPE(IRPT) = 1
              IRPT = 0
          END IF
          IF (IRTYPE(1) .GE. 1) CALL RPT01(IRTYPE(1))
          IF (IRTYPE(2) .EQ. 1) CALL RPT02(NATP)
          IF (IRTYPE(3) .EQ. 1) CALL RPT03(NASP)
          IF (IRTYPE(4) .EQ. 1) CALL RPT04
          IF (IRTYPE(5) .EQ. 1) CALL RPT05(NATP)
          IF (IRTYPE(6) .EQ. 1) CALL RPT06(NASP)
          IF (IRTYPE(7) .EQ. 1) CALL RPT07
          IF (IRPT .EQ. 0) THEN
              PRINT*, ' ENTER ZERO TO STOP '
          
```

```

        END IF
        RETURN
C
C
        SUBROUTINE RPT01(IANS)
C      REPORT TYPE ONE
C**** IUNIT REPORT
        150 FORMAT(/, ' UNIT STATUS PRINT OPTIONS: ',/,
          $ ' 1 - PRINT ALL',/,
          $ ' 2 - SINGLE UNIT',/,
          $ ' 3 - RETURN',/,
          $ ' ?')
        IF(IANS.LT.1.OR.IANS.GT.3) THEN
          WRITE(6,150)
          READ(5,*)IANS
        END IF
        GO TO (160,240,230) IANS
C**** LOOP THROUGH UNITS
        160 DO 220 I = 1,75
C**** IF NO TYPE CODE BYPASS
          IF(IUNIT(I,1).EQ. 0)GO TO 220
C**** IF NO UNIT NAME GO TO 400
          IF(AUNIT(I,2).EQ.' ') GO TO 220
C**** PRINT HEADER.
          WRITE(14,170)
        170 FORMAT(42X,///, ' UNIT STATUS',43X, 'UNIT DATA',29X, 'WPN DATA',//
          $,15X, 'AMMO-CODE  WPN-TYP  WPN-ALIVE  CUR-SHP  RNDS-ONWAY  PCBL-W ON-
          $TRKS  NO WPN SH / NO SH EA TOT-DMD ' )
          WRITE(14,180)AUNIT(I,2),I,IUNIT(I,1),IUNIT(I,2),IUNIT(I,4),
          $ IUNIT(I,3),IUNIT(I,5),IUNIT(I,7)
        180 FORMAT(1X,A10,2I3,/, ' SER ATP ',I2,I3, ' KM',/, ' SER ASP ',I2,I3,
          $KM',/, ' NO HELD ',I2)
          DO 210 J = 1,LPPAR(6)
            JJ = 13 * J - 5
            IF(IUNIT(I,JJ).EQ. 0)GO TO 210
            NMSHT=0
            IF(IUNIT(I,JJ+2).EQ.0) GO TO 190
            NMSHT = IUNIT(I,JJ+3) / IUNIT(I,JJ+2)
        190 CONTINUE
            IPCBL=10*IUNIT(I,JJ+4)/(IUNIT(I,JJ+1)*IUNIT(I,JJ+7))
            WRITE(14,200)IUNIT(I,JJ),IWPN(IUNIT(I,JJ)),IUNIT(I,JJ+1),IUNIT(
          $I,JJ+4),IUNIT(I,JJ+12),IPCBL,IUNIT(I,JJ+8),IUNIT(I,JJ+2),NMSHT,
          $ IUNIT(I,JJ+11)
        200 FORMAT(18X,I3,6X,A8,I7,I8,5X,I5,4X,I6,2X,I5,8X,I4, ' / ',I5,5X,I5)
        210 CONTINUE
C**** PRINT STATUS OF UNIT TRUCKS
          CALL TRUCK (I)
        220 CONTINUE
        230 RETURN
        240 WRITE(6,250)
        250 FORMAT(' ENTER JIFFY UNIT ID (INPUT 0 TO EXIT) ')
          READ(5,260) NAME
          IF (NAME .EQ. '0') RETURN
        260 FORMAT(A10)

```

```

        IF(IUNIT(K,1).EQ.0) GO TO 270
        IF(AUNIT(K,2).EQ.NAME) GO TO 290
270 CONTINUE
        WRITE(6,280) NAME
290 FORMAT(' UNIT ',A10,' NOT FOUND')
        GO TO 240
290 WRITE(14,300) NAME
300 FORMAT(/,1X,'UNIT',1X,A10,4X,'UNIT DATA',22X,'WPN DATA',/,10X,
$ 'WPN',10X,'RND',18X,'# WPN # RND',/,1X,'WPN-TYP',2X,
$ 'LIVE CUR-SUP ONWAY PCBL ON-TRKS',3X,'SHORT SH EA TOT-DMD')
        DO 320 KK=8,LPPAR(6) * 13 - 5,13
        IF(IUNIT(K,KK).EQ.0) GO TO 320
        NMSHT=IUNIT(K,KK+3)/IUNIT(K,KK+2)
        IPCBL=100*IUNIT(K,KK+4)/(IUNIT(K,KK+1)*IUNIT(K,KK+7))
        WRITE(14,310) IWPN(IUNIT(K,KK)),IUNIT(K,KK+1),IUNIT(K,KK+4),
$ IUNIT(K,KK+3),IPCBL,IUNIT(K,KK+8),IUNIT(K,KK+2),NMSHT,
$ IUNIT(K,KK+11)
310 FORMAT(1X,A8,1X,I3,2X,I6,2X,I5,2X,I3,2X,I6,4X,I4,2X,I5,1X,I5)
320 CONTINUE
C**** PRINT STATUS OF UNIT TRUCKS
        CALL TRUCK (K)
        GO TO 240
C
C
C
C
        SUBROUTINE RPT02(NATP)
C**** REPORT TYPE TWO
C
        ATP STATUS
        WRITE (6,*) ' ENTER NUMB OF ACTIVE ATPS (1,2,...OR10)'
        READ (5,*) NATP
        IF (NATP .LT. 1 .OR. NATP .GT. 10) NATP = 10
        DO 380 I = 1,NATP
        WRITE(14,350)I,IATP(I,14),IATP(I,15)
350 FORMAT(////,25X,' ATP STATUS ',//,3X,'ATP NO ',I3,/,
$ 10X,'QUEUE ARTY MU',/,10X,
$ 'TRUCKS IN Q',
$ 5X,I3,8X,I3,/,10X,'AMMO-CODE AMT-Q/H CUR-DMD ON-THE-WAY')
        DO 370 J = 1,11
        JJ = J * 3 + 18
        WRITE(14,360)J,IATP(I,JJ),IATP(I,JJ+1),IATP(I,JJ+2)
360 FORMAT(13X,I3,4X,I8,2X,I6,5X,I6)
370 CONTINUE
380 CONTINUE
        write(14,382)
382 format(////,37x,'atp queue information',/,
$ 5x,'atp num',5x,'trks served',5x,'avs time',5x,'max time',
$ 5x,'trks served',5x,'avs time',5x,'max time',/,
$ 16x,'manuver queue',5x,'in queue',5x,'in queue',
$ 5x,' arty queue ',5x,'in queue',5x,'in queue')
        do 384 J=1,natp
        write(14,383)J,Jatp(J,1),Jatp(J,2)/Jatp(J,1),Jatp(J,3),
$ Jatp(J,4),Jatp(J,5)/Jatp(J,4),Jatp(J,6)
383 format(7x,i2,11x,i4,11x,i4,9x,i4,9x,i4,12x,i4,9x,i4)

```

```

      RETURN
C
C
C
      SUBROUTINE RPT03(NASP)
C
C**** REPORT TYPE THREE
C**** IASP REPORT
      WRITE (6,*) ' ENTER NUMBER OF ACTIVE ASPS (1,2,...OR10)'
      READ (5,*) NASP
      IF (NASP .LE. 0 .OR. NASP .GT. 10 )NASP = 10
      DO 430 I = 1, NASP
C**** OUTPUT INFO
      WRITE(14,400)I+10,IASP(I,12),IASP(I,13)
400 FORMAT(1X,////,25X,' ASP STATUS ',////,5X,' ASP-NO ',I3,///,
$ 15X,'QUEUE          ROUTINE          MLRS',/,
$ 10X,'TRUCKS IN Q',5X,I5,8X,I5,////,' INVENTORY',6X,
$'AMMO-CODE      AMT-O/H DEMAND  ON-THE-WAY')
      DO 420 J = 1,LPPAR(1)
      JJ = J * 3 + 18
      WRITE(14,410)J,IASP(I,JJ),IASP(I,JJ+1),IASP(I,JJ+2)
410 FORMAT(19X,I3,4X,I8,2X,I7,4X,I7)
420 CONTINUE
430 CONTINUE
      RETURN
C
C
C
      SUBROUTINE RPT04
C**** REPORT TYPE FOUR
C**** ICSA REPORT
      WRITE(14,450)
450 FORMAT(1X,////,15X,' CSA-STATUS ',//,20X,' ROUNDS DRAWN FROM CSA',
$ /21X,' AMMO ',10X,'NUMBER-DRAWN ')
      DO 470 I = 1,LPPAR(1)
      WRITE(14,460)I,ICSA(3,I)
460 FORMAT(22X,I3,14X,I6)
470 CONTINUE
      RETURN
C
C
C
C
      SUBROUTINE RPT05(NATP)
C**** REPORT TYPE FIVE
C**** IATFAM REPORT - AMMO ISSUED BY ATPS
C**** LOOP THROUGH ATPS
C**** WRITE HEADERS
      WRITE (14,530)
530 FORMAT(1X,///,30X,' AMMO ISSUED BY ATP ',20X,
$ 'trucks bumped to asp ',//,5X,' atp no.      1      2      3      4
$ 5      6      7      8      9      10     11     12     13     14     15     16     17
$18     19     20',
do 550 i = 1,NATP

```

```

      $ IATPAM(I,5),IATPAM(I,6),IATPAM(I,7),IATPAM(I,8),IATPAM(I,9),
      $ IATPAM(I,10),IATPAM(I,11),IATPAM(I,12),IATPAM(I,13),IATPAM(I,
      $ 14),IATPAM(I,15),IATPAM(I,16),IATPAM(I,17),IATPAM(I,18),
      $ IATPAM(I,19),IATPAM(I,20)
540 FORMAT (7X,I2,5X,20I5)
550 CONTINUE
      RETURN
C
C
C
      SUBROUTINE RPT06(NASP)
C**** REPORT TYPE SEVEN
      IF (NASP.LE.0.OR.NASP.GT.6) NASP = 6
C**** AMMO REMOVED FROM ASPS
C**** WRITE HEADER
      WRITE(14,570)
C
      LOOP THROUGH ASPS
      DO 600 I = 1,NASP
        WRITE (14,570) I+10
570 FORMAT (///,20X,'ASP ',I2,10X,'AMMO TYPE',10X,'AMMO REMOVED')
        DO 590 J= 1,23
          WRITE (14,580) J,IASPAM(I,J)
580 FORMAT (40X,I2,15X,I7)
590 CONTINUE
600 CONTINUE
        write(14,602)
602 format(////,37X,'asp queue information',/,/,
          z 5X,'asp num',5X,'trks served',5X,'avs time',5X,'max time',
          z 5X,'trks served',5X,'avs time',5X,'max time',/,/,
          z 16X,'routine que',5X,'in queue',5X,'in queue',
          z 5X,'ssrs queue ',5X,'in queue',5X,'in queue')
          do 605 J=1,nasp
            write(14,604)J+10,Jasp(J,1),Jasp(J,2)/Jasp(J,1),Jasp(J,3),
              z Jasp(J,4),Jasp(J,5)/Jasp(J,4),Jasp(J,6)
604 format(7X,I2,11X,I4,11X,I4,9X,I4,9X,I4,12X,I4,9X,I4)
605 continue
          RETURN
C
C
C
      SUBROUTINE RPT07
C**** REPORT TYPE SEVEN
C**** TRUCKS THAT HAVE BEEN KILLED OR HAVE BROKEN
C**** LOOP THROUGH TRUCKS FOR DEAD OR BROKEN
      DO 650 I = 1,LPPAR(4)
C**** IF NOT DEAD, GO TO 630
      IF(ITRUCK(I,3) .NE. 7)GO TO 630
C**** HAVE DEAD TRUCK, PRINT NUT
      WRITE(14,620)I,AUNIT(ITRUCK(I,4),2),ITRUCK(I,1),ITRUCK(I,5)
620 FORMAT(' TRUCK NUMB',I4,' OF UNIT ',A10,' WHICH IS TYPE 'I4,' DARE
      $YING AMMN',I4,' IS DEAD')
      GO TO 650
630 CONTINUE
C**** IF NOT BEING REPAIRED GO TO 650

```



```

C
  WRITE (18,11)
11  FORMAT(//,10X,'EMPTY S&P TRAILERS AT CSA',//,
    A9X,'S&P NO',8X,'OWNER',9X,'MIX',/)
    DO 25 III=836,1211
C*****IF NOT IN CSA DROP THROUGH
C
    IF(ITRUCK(III,3).NE.8)GO TO 25
    WRITE(18,35)III,ITRUCK(III,4),ITRUCK(III,5)
35  FORMAT(10X,I4,10X,I3,10X,I2)
25  CONTINUE
C
C*****PRINT EMPTY S&P TRAILERS AT ATFS
  WRITE(18,22)
22  FORMAT(//,10X,'EMPTY S&P TRAILERS AT ATP',//,
    A9X,'S&P NO',8X,'OWNER',9X,'MIX',/)
    DO 55 JJJ=835,1211
    IF(ITRUCK(JJJ,3).NE.2)GO TO 55
    IF(ITRUCK(JJJ,6).NE.0)GO TO 55
    WRITE(18,35)JJJ,ITRUCK(JJJ,4),ITRUCK(JJJ,5)
55  CONTINUE
    return
  end

```

```

pp.      SUBROUTINE SCHED
          SUBROUTINE SCHED (ITYPE, IPARM, TIME)
C**** INTERFACE ROUTINE TO SCHEDULE EVENT
C
C      CALLED BY ASP, ASPAR1, ASPAR2, ASPARV, ASPCK, ATP, ATPAR1, ATPAR2,
C      ATPARV, CONTRL, CREEVT, CSAARV, CSADEP, DEMAND, DEASP,
C      DUALMX, HELARV, INIT, RDIEXO, RDIFF, RELOAD, SFRVFR,
C      SEVENT, UNTARV, UNTDEP
C      CALLS      LOOKEV(TO PRINT EVT), PUTEVT(TO PUT EVT IN Q), CONTRL
C
C**** H. JONES      DEC 78
C
C      LOCAL VARIABLES(NO COMMON)
C      TIME  -- TIME THAT EVENT IS TO HAPPEN, NOT CURRENT TIME!
C      ITH   -- TIME                      IN WHOLE MINUTES
C      ITS   -- TIME (FRACTIONAL PART * 3600)
C      ICHECK-- 0, PUTEVT RETURNED NORMALLY. >0, ABNORMAL.
C
C      DIMENSION IPARM(5)
C      IPARM(5)=ITYPE
C      CALL LOOKEV (ITYPE+0, IPARM, TIME+0., 0)
C      ITH = TIME
C      ITS = (TIME - ITH) * 3600
C      CALL PUTEVT (IPARM, ITH, ITS, ICHECK)
C      IF(ICHECK.NE. 0) THEN
C          WRITE(6,30) ICHECK
C          CALL CONTRL(TIME)
C      END IF
C
C      RETURN
30 FORMAT(' TOO MANY EVENTS -- ',I6)
END

```

```

qq.      SUBROUTINE SERVER
        SUBROUTINE SERVER(IPARM)
C
C**** THIS ROUTINE PROCESSES A SERVER WHEN IT BECOMES AVAILABLE.
C      IT CHECKS THE WAIT QUEUES FOR TRUCKS AND SCHEDULES A RESUPPLY
C      EVENT -- ATP OR ASP --, IF AN APPROPRIATE TYPE IS FOUND. IF
C      NONE IS FOUND, THE SERVER IS PUT INTO ITS WAITING QUEUE.
C
C      CALLED BY ASPAR2, UNTARV
C      CALLS      PUTQUE, SCHED, GETQUE, IQ, FINTK, OPERA
C
C**** LOREN IVERSON  NOV 81
C
C**** SCHEDULES -- ATP, LOADING OF A UNIT RESUPPLY TRUCK
C                ASP, LOADING OF A UNIT RESUPPLY TRUCK
C                ASPAR1, SERVER AND S&P INTO THE ASP
C                ASPAR2, RETURN OF EMPTY S&P
C                UNTARV, RETURN OF IDLE SERVER
C
C**** IPARM(1) = ATP - ASP NUMBER
C**** IPARM(2) = SERVER NUMBER
C**** IPARM(3) = FLAG, EQUALS 1 IF SERVER IS FROM ASP, 0 IF FROM ATP
C
C**** LOCAL VARIABLES
C      MANART -- SET TO 1 FOR MANUEVER, 2 FOR ARTY
C      NUMTK  -- THE NUMBER OF THE SUPPLY TRUCK TO BE LOADED
C      NUMQ   -- THE QUEUE NUMBER FOR SERVER
C      IQUE   -- THE QUEUE NUMBER FOR UNIT TRUCK
C      NCKSP  -- THE CHECK TRUCK FOR LASP S&P IN QUEUE
C      DTIME  -- DELAY TIME
C      NTQUE  -- TIME SPENT IN QUEUE
C
C      INCLUDE LOG,LIST
C      DIMENSION IPARM(5)
C
C      NUMSVR = IPARM(2)
C      IAPTYP = IPARM(3)
C
C ** CHECK FOR SERVER
C      IF(ITRUCK(NUMSVR,1).NE.9.AND.ITRUCK(NUMSVR,1).NE.8)THEN
C          WRITE(LUOUT,5)NUMSVR
C          5   FORMAT(2X,I5,' IS NOT A SERVER')
C          RETURN
C      END IF
C      NFLAG = 0
C**** SET SERVER STATUS AS AVAILABLE
C      ITRUCK(NUMSVR,3) = 1
C
C**** DETERMINE IF THE UNIT IS AN ATP OR ASP(=1)
C      IF(IAPTYP .EQ. 1) GO TO 40
C
C****      HAVE AN ATP
C      NATP = IPARM(1)
C * * * IF EQ, ATP HAS BEEN INTERDICTED -- PUT SERVER IN QUE

```

```

        CALL PUTQUE(NUMSVR,ISERV(3))
        ITRUCK(NUMSVR,3) = 9
        RETURN
    END IF
C
C**** IATP(NATP,16) IS ATP SERVER REMOVAL COUNTER
    IF(IATP(NATP,16) .LT. ISERV(1) .AND. NATP .NE. IATPSD(4)) THEN
        DTIME = ISERV(7)
        IPARM(4) = DTIME
    8    CALL SCHED(8,IPARM,TIME+DTIME)          @ UNTARV
        ITRUCK(NUMSVR,3) = 5
        IATP(NATP,16) = IATP(NATP,16) + 1
        WRITE(6,9)NUMSVR,NATP,TIME
    9    FORMAT(5X,'SERVER',I5,' AT ATP',I3,' IS MOVING REARWORD',
        $      ' AT TIME',F8.1)
        IF(IATP(NATP,16) .LT. ISERV(1))THEN
            CALL GETQUE(NUMSVR,IATP(NATP,10))
            IF(NUMSVR .EQ. 0)RETURN
            IPARM(2) = NUMSVR
            GO TO 8
        END IF
        RETURN
    END IF
    IF(ITRUCK(NUMSVR,1) .EQ. 9) GO TO 20
C
C****      HAVE A FORKLIFT
    IQUE = IATP(NATP,12)
    CALL GETQUE(NUMTK,IQUE)
    IF(NUMTK .NE. 0) THEN
C****      HAVE A MANUEVER TRUCK SET UP FOR ATP EVENT
        MANART = 2
    ELSE
        IQUE = IATP(NATP,11)
        CALL GETQUE(NUMTK,IQUE)
        IF(NUMTK .NE. 0) THEN
            NCKTK = NUMTK
    12        IF((ITRUCK(NUMTK,5) .EQ. 10) .OR. (ITRUCK(NUMTK,5) .EQ. 40)) THEN
                CALL PUTQUE(NUMTK,IQUE)
                CALL GETQUE(NUMTK,IQUE)
                IF(NUMTK .EQ. NCKTK) THEN
                    CALL PUTQUE(NUMTK,IQUE)
                    GO TO 10
                ELSE
                    GO TO 12
                END IF
            END IF
        END IF
C****      HAVE AN ARTY TRUCK SET UP ATP EVENT
        MANART = 1
    ELSE
C
C****      PUT THE SERVER IN THE QUEUE
    10        NUMQ = IATP(NATP,10)
        CALL PUTQUE(NUMSVR,NUMQ)
        RETURN

```

```

        END IF
        NFLAG = 1
        GO TO 50
C
C****    HAVE A CRANE
20    IQUE = IATP(NATP,11)
        CALL GETQUE( NUMTK,IQUE)
        IF( NUMTK .NE. 0) THEN
            NCKTK = NUMTK
22    IF((ITRUCK(NUMTK,5).EQ.10).OR.(ITRUCK(NUMTK,5).EQ.40))THEN
            CALL PUTQUE(NUMTK,IQUE)
            CALL GETQUE(NUMTK,IQUE)
            IF(NUMTK .EQ. NCKTK)THEN
                CALL PUTQUE(NUMTK,IQUE)
                GO TO 25
            ELSE
                GO TO 22
            END IF
        END IF
C****    HAVE A ARTY TRUCK SET ATP EVENT
        MANART = 1
    ELSE
C****    PUT THE SERVER IN THE QUEUE
25    NUMQ = IATP(NATP,10)
        CALL PUTQUE(NUMSVR, NUMQ)
        RETURN
    END IF
C
50    MIX = ITRUCK(NUMTK,5)
        JLOOP = LPPAR(1)
        DO 55 I = 1,LPPAR(1)
            IF(IMIX(MIX,I) .EQ. 0)GO TO 55
            IAM = I
            GO TO 60
        55 CONTINUE
C
C**** CHECK IF AN S&P IS AVAILABLE FOR DELOADING
60    NATPQ = IQ(2,NATP)
70    CALL FINTK(NATPQ,IAM,NATPTK,0)
        IF(NATPTK .NE. 0)GO TO 80
        IF(NATPQ .NE. IQ(3,NATP))THEN
            NATPQ = IQ(3,NATP)
            GO TO 70
        ELSE
            CALL PUTQUE(NUMTK,IQUE)
            IF(NFLAG .EQ. 1)THEN
                GO TO 10
            ELSE
                GO TO 25
            END IF
        END IF
C
C**** HAVE FOUND A TRUCK, PUT BACK IN QUEUE
80    CALL PUTQUE(NATPTK,NATPQ)

```

```

C**** SCHEDULE THE ATP EVENT
      IPARM(4) = NUMSVR
      IPARM(3) = NUMTK
      IPARM(2) = NATP
      IPARM(1) = MANART
      CALL SCHED(6,IPARM,TIME)           @ ATP
      ITRUCK(NUMSVR,3) = 4
      RETURN

C
C**** HAVE AN ASP, LOOK FOR A TRUCK
      40 NASP = IPARM(1) - 10
C * * * IF EQ, ASP HAS BEEN INTERDICTED -- PUT SERVER IN QUE
      IF((NASP + 10) .EQ. ISERV(6)) THEN
        CALL PUTQUE(NUMSVR,ISERV(4))
        ITRUCK(NUMSVR,3) = 9
        RETURN
      END IF
C**** IASP(NASP,16) IS ASP SERVER REMOVAL COUNTER
      IF (IASP(NASP,16) .LT. ISERV(2) .AND. IASP(NASP,2) .EQ. 0) THEN
        NRASP = IASP(NASP,11) - 10
        DIST = IASP(NASP,1) - IASP(NRASP,1)
        TRTM = 60. * DIST / ITYPE(5,IDAY+3) + 30.    @ 30 MIN LOAD/UNL
        IPARM(4) = TRTM
      45 ITRUCK(NUMSVR,3) = 5
        ITRUCK(NUMSVR,4) = NRASP + 155
        IASP(NASP,16) = IASP(NASP,16) + 1
        CALL SCHED(8,IPARM,TIME+TRTM)    @ SEND TO REAR ASP(UNTARV)
        WRITE(6,18)NUMSVR,NASP,NRASP+10,TIME
      48 FORMAT(5X,'SERVER',I5,' AT ASP',I3,' MOVED TO ASP',I3,
        * ' AT TIME',F8.1)
        IF(IASP(NASP,16) .LT. ISERV(2)) THEN
          CALL GETQUE(NUMSVR,IASP(NASP,7))
          IF(NUMSVR .EQ. 0) RETURN
          IPARM(2) = NUMSVR
          GO TO 45
        END IF
        RETURN
      END IF
C * * PREPARATION OF HELICOPTER SLING LOAD
      IF(IASP(NASP,15) .GT. 0) THEN
        IASP(NASP,15) = IASP(NASP,38) - 1
        IPARM(4) = 47
        CALL SCHED(8,IPARM,TIME + 45.)    @ UNTARV
        ITRUCK(NUMSVR,3) = 4
        RETURN
      END IF
C**** LOCATE UNIT TRUCK TO BE LOADED
      NUMQ = IASP(NASP,9)
      CALL GETQUE(NUMTK,NUMQ)
      IF(NUMTK .GT. 0) THEN
        NCKTK = NUMTK
        CALL PUTQUE(NUMTK,NUMQ)
      400 CALL GETQUE(NUMTK,NUMQ)
        MIX = ITRUCK(NUMTK,5)

```

```

      NUMAM = MIX
      IF (MIX .GT. LPPAR(7)) NUMAM = MIX - LPPAR(7)
C * * * DETERMINE IF SUFFICIENT AMMO AT PARENT ASP
      IF (IASP(NASP,NUMAM*3+19) - IASP(NASP,NUMAM*3 + 19) .GE.
$      IMIX(MIX,NUMAM)) THEN
C**** HAVE A TRUCK, SCHEDULE ASP EVENT
      IPARM(1) = 1
      IPARM(2) = NASP + 10
      IPARM(3) = NUMTK
      IPARM(4) = NUMSVR
      CALL SCHED(7,IPARM,TIME)          @ ASP
      ITRUCK(NUMSVR,3) = 4
      RETURN
      ELSE
      CALL PUTQUE(NUMTK,NUMQ)
      IF(NCKTK .EQ. NUMTK) GO TO 600
      GO TO 400
      END IF
    ELSE
C
C*****
C
C
C * * * CHECK FOR ASP-ATP S&P TO LOAD
    600 CALL GETQUE(NATPSP,IQ(11,NASP))
      IF(NATPSP .EQ. 0)GO TO 700
C * * * SERVER AVAILABLE PUT IN QUEUE AND SCHED ASPAR1
      CALL PUTQUE(NUMSVR,IQ(10,NASP))
      NATP = ITRUCK(NATPSP,4) - 85
      IPARM(1) = NATP
      IPARM(2) = NATPSP
      IPARM(3) = NASP + 10
      IPARM(4) = 0
      CALL SCHED(12,IPARM,TIME)          @ ASPAR1
C      FIND THE QUEUE WAIT TIME AND ADD IT TO TOTAL QUEUE TIME
      NTQUE = TIME - ITRUCK(NATPSP,3)
      ITRUCK(NATPSP,12) = ITRUCK(NATPSP,12) + NTQUE
      JASP(NASP,8) = JASP(NASP,8) + NTQUE
C      IF THIS IS THE LONGEST QUEUE WAIT RECORD IT
      IF(NTQUE.GT.JASP(NASP,9)) JASP(NASP,9) = NTQUE
      RETURN
C
C*****
C*****LOOK FOR S & P TO OFF LOAD
    700 NUMQ = IASP(NASP,4)
      CALL GETQUE(NUMSP,NUMQ)
      IF(NUMSP.NE.0)GO TO 300
C
C***** NO TRAILERS IN QUE,PUT SERVER BACK IN HIS QUE
      NUMQ = IASP(NASP,7)
      CALL PUTQUE(NUMSVR,NUMQ)
      RETURN
      END IF
C

```

```

800 IF (ITRUCK(NUMSP,6).EQ.0) THEN
      NCKSP = NUMSP
850   CALL PUTQUE(NUMSP,NUMQ)
      CALL GETQUE(NUMSP,NUMQ)
      IF (ITRUCK(NUMSP,6).NE.0) GO TO 900
      IF (NUMSP.EQ.NCKSP) THEN
        CALL PUTQUE(NUMSP,NUMQ)
        NUMQ = IASP(NASP,7)
        CALL PUTQUE(NUMSVR,NUMQ)
        RETURN
      END IF
      GO TO 850
    ELSE
900   WRITE(18,950)NUMSVR,NUMSP,ITRUCK(NUMSP,5),ITRUCK(NUMSP,6),TIME
950   FORMAT(/,10X,'SERVER NO.',I5,' IS OFFLOADING',I5,
1     ' MIX ',I2,' % ',I5,' AT TIME ',F8.1)
      MIX = ITRUCK(NUMSP,5)
      ITRUCK(NUMSVR,3) = 4
C
C***** CHECK FOR FAILURE & SCHEDULE AVAIL. OF SFRVER
      TLOAD = IMIX(MIX,32)* (ITRUCK(NUMSP,6)/10000.)
      CALL OPERA(NUMSVR,TLOAD,TFAIL)
      DTIME = TIME + TLOAD + TFAIL
      CALL SCHED(8,IPARM,DTIME) @ UNTARV
      ITRUCK(NUMSVR,5) = ITRUCK(NUMSVR,5) + TLOAD
C      TOT TIME OFF-LOADING S&PS AT ASF
      ITRUCK(NUMSVR,9) = ITRUCK(NUMSVR,9) + 1 @SERVER OFF-LOAD CTR
C
C***** DECREMENT AMMO ON TRAILER
      ITRUCK(NUMSP,6) = 0
      IPARM(2) = NUMSP
      IPARM(3) = 0
      IPARM(4) = 555
      IF (TFAIL .GT. 0.) DTIME = DTIME - TFAIL
      CALL SCHED(13,IPARM,DTIME) @ ASPAR2
    END IF
C
    RETURN
  END

```

```

rr.      SUBROUTINE SEVENT
          SUBROUTINE SEVENT
C
C*****
C
C      THIS ROUTINE ADDS THE EVENTS CREATED SEPARTELY
C      AND STORED ON FILE TO BE SCHEDULED IN RUNS OF ARM
C      (INPUT THRU EVENTSCI--).
C
C
C      ADDED 1 MAY 82                      ANN HILLS
C*****
C
C      VARIABLES USED:
C      ITYP...THE TYPE OF EVENT
C      IIPARMS...THE PARAMETERS OF THE EVENT
C      SSTIME...THE TIME THE EVENT IS TO BE SCHEDULED
C
C*****
C      FILES USED:
C      FILE 11 WHICH CONTAINS THE EVENTS
C      ROUTINES CALLED:
C      SCHED TO SCHEDULE THE EVENTS
C      CALLED BY INIT
C*****
C
C      DIMENSION ITYPE(500),IPARM1(500),IPARM2(500),IPARM3(500)
C      DIMENSION IPARM4(500),IPARM5(500),STIME(500),IIPARM(5)
C
C      ZERO OUT THE ARRAYS
C      DO 100 I=1,500
C          ITYPE(I)=0
C          IPARM1(I)=0
C          IPARM2(I)=0
C          IPARM3(I)=0
C          IPARM4(I)=0
C          IPARM5(I)=0
C          STIME(I)=0.
100      CONTINUE
C      DO 101 I=1,5
C          IIPARM(I)=0
101      CONTINUE
C
C
C      NOW READ THE EVENTS FROM FILE 11 INTO TEMPORARY ARRAYS
C
C      READ(11) ITYPE,IPARM1,IPARM2,IPARM3,IPARM4,IPARM5,STIME
C
C      LOOP TO SET ARRAY VALUES EQUAL TO NECESSARY PARAMETERS
C      NEEDED TO SCHEDULE EVENTS....IF AT THE END OF THE
C      EVENTS GET OUT OF THE LOOP
C
C      DO 500 I=1,500
C          ITYP=ITYPE(I)

```

```

      IF(ITYP.LT.0.OR.ITYP.GT.19)THEN
        PRINT*, 'THE FOLLOWING EVENT IS INCORRECT'
        PRINT 12, ITYP, IPARM1(I), IPARM2(I), IPARM3(I), IPARM4(I), IPARM5(I)
12      FORMAT(T2, 'EVT # ', I3, ' WITH PARMS ', 5I8)
        PRINT*, 'THIS EVENT IS NOT SCHEDULED'
        GO TO 500
      END IF
      IIPARM(1)=IPARM1(I)
      IIPARM(2)=IPARM2(I)
      IIPARM(3)=IPARM3(I)
      IIPARM(4)=IPARM4(I)
      IIPARM(5)=IPARM5(I)
      SSTIME=STIME(I)
      CALL SCHED(ITYP, IIPARM, SSTIME)
500    CONTINUE
      C
      RETURN
      END

```

```

SS.      SUBROUTINE SETQUE
          SUBROUTINE SETQUE (ITEM, NUMQUE)
C**** SETS UP NUMQUE EMPTY QUEUES FOR ITEM.
C        EX.  CALL SETQUE(176,1400) CLEARS ALL QUEUES.
C
C        CALLED BY TRKPUT
C
C**** H. JONES    DEC 78
          INCLUDE QUENUM,LIST
          INCLUDE QUEPNT,LIST
          DO 10 I=1,NUMQUE
10 IHEAD(I) = 0
C
          DO 20 I=1,ITEM
20 ITEMS(I) = 0
C
          RETURN
          END

```

```

tt.  SUBROUTINE TRKPUT
      SUBROUTINE TRKPUT
C**** ALLOWS INTERACTIVE TRUCK QUEUE RE-ASSIGNMENT
C
C      CALLED BY CONTROL (OR EDIT IN THE EDIT PROGRAM)
C      CALLS      READF, NXTQUE, PUTQUE, GETQUE, SETQUE, GETONE
C
C**** H. JONES      FEB 79
C
      CHARACTER*20 IWORD
      DIMENSION INTGR(10), REAL(10), IWORD(10)
C
      5 WRITE(6,10)
      10 FORMAT(1X,'COMMAND EXAMPLES :',/,
        $ 1X,'GET 3 FROM 35 ',/,
        $ 1X,'PUT 3, 10 IN 105 ',/,
        $ 1X,'LIST 105 ',/,
        $ 1X,'TAKE OUT ',/,
        $ 1X,'END ',/)
C
      20 WRITE(6,30)
      30 FORMAT('...')
         CALL READF (5, 10, INTGR, REAL, IWORD)
         IF(IWORD(1) .EQ. 'GET' .OR. IWORD(1) .EQ. 'G') GO TO 40
         IF(IWORD(1) .EQ. 'PUT' .OR. IWORD(1) .EQ. 'P') GO TO 60
         IF(IWORD(1) .EQ. 'LIST' .OR. IWORD(1) .EQ. 'L') GO TO 80
         IF(IWORD(1) .EQ. 'TAKE' .OR. IWORD(1) .EQ. 'T') GO TO 110
         IF(IWORD(1) .EQ. 'END' .OR. IWORD(1) .EQ. 'E') GO TO 120
         IF(IWORD(1) .EQ. 'HELP' .OR. IWORD(1) .EQ. 'H') GO TO 5
         PRINT*, 'WHAT???'
         GO TO 20
C
C**** GET TRUCK FROM QUEUE WITHOUT RE-ORDERING QUEUE
      40 I1 = INTGR(1)
         I2 = INTGR(2)
         IF(INTGR(3) .NE. 0) THEN
            PRINT*, 'INPUT ERROR'
            GO TO 20
         END IF
         CALL GETONE(I2, I1, IFLAG)
         IF(IFLAG .NE. 0) PRINT*, I1, ' NOT GOT'
         GO TO 20
C
C**** PUT TRUCK IN QUEUE
      60 IF(INTGR(1) .LE. 0 .OR. INTGR(2) .LE. 0 .OR. INTGR(1) .GT. 1440) THEN
         PRINT*, 'RE-ENTER '
         GO TO 20
      END IF
      I1 = INTGR(1)
      I2 = INTGR(2)
      I3 = INTGR(3)
      IF(INTGR(3) .EQ. 0) THEN
         I3 = INTGR(2)
         I2 = INTGR(1)

```

```

      DO 70 I=I1,I2
70  CALL PUTQUE (I, I3)
      GO TO 20

C
C**** LIST TRUCKS IN QUEUE
      80 CALL NXTQUE (IFIRST, INTGR(1))
        IF(IFIRST .EQ. 0) GO TO 20
      90 CALL GETQUE(NUMTK, INTGR(1))
        CALL PUTQUE(NUMTK, INTGR(1))
        WRITE(6,100) NUMTK
      100 FORMAT(1X,I5)
        CALL NXTQUE (INEXT, INTGR(1))
        IF(INEXT .NE. IFIRST) GO TO 90
        GO TO 20

C
C**** TAKE TRUCKS OUT OF QUEUES
      110 PRINT*, ' ENTER QUEUE NUMBER (999 TO TAKE OUT FROM ALL QUEUES)'
        READ(5,*)IGNUM
        IF(IGNUM .NE. 999) THEN
      200   CALL GETQUE(NUMTK,IGNUM)
           IF(NUMTK .EQ. 0) GO TO 20
           GO TO 200
        ELSE
           CALL SETQUE(1400,176)
        END IF
        GO TO 20

C
      120 RETURN
      END

```

```

uu.  SUBROUTINE TRKTIM
      SUBROUTINE TRKTIM
      C      INITIALIZES TRUCKS' TIME TO NEXT FAIL(EXPONENTIAL DISTRIBUTION).
      C
      C      CALLED BY INIT
      C      CALLS      NOTHING
      C
      C      INCLUDE LOG,LIST
      C      CHARACTER*3 IANS
      C
      C      WRITE(6,10)
      10  FORMAT(' INITIALIZE TRUCKS' TIME SINCE LAST FAILURE? (YES/NO)')
      C      READ(5,20)IANS
      20  FORMAT(A3)
      C
      C      IF(IANS .EQ. 'NO ' .OR. IANS .EQ. 'N ') GO TO 40
      C
      C**** LOOP THROUGH TRUCKS
      DO 30 I = 1,LPPAR(4)
      C      ITRUCK(I,1)
      C      IF NOT ACTIVE BYPASS
      C      IF (ITYP .EQ. 0) GO TO 30
      C      FIND MTBF
      C      XMTBF = ITYPE(ITYP,5)
      C      COMPUTE TIME TO NEXT FAILURE(EXPONENTIAL DISTRIBUTION)
      C      UNRN = RANF(DUM)
      C      XRN = (-XMTBF*ALOG(1-UNRN))
      C      STORE TIME SINCE MAINT. FOR THIS TRUCK
      C      ITRUCK(I,7) = XRN
      30  CONTINUE
      C
      40  RETURN
      END

```

```

VV.      SUBROUTINE TRUCK
          SUBROUTINE TRUCK (L)
C**** WRITES STATUS OF UNIT TRUCKS
C
C      CALLED BY REPORT
C      CALLS      NOTHING
C
C**** D REMEN JUN 79
C
          CHARACTER*10 AUNIT
          INCLUDE LOG,LIST
          INCLUDE AUNIT,LIST
C**** WRITE HEADER
          WRITE(14,10)AUNIT(L,2)
          10 FORMAT(1X,///,8X,' TRUCK STATUS REPORT FOR UNIT ',A10,///,
             $      '      TRK      NM      STATUS      MIX      PCLDAD      NXFAIL',/)
C
C**** LOOP THROUGH THE TRUCKS
          DO 30 J = 1,LPPAR(4)          @ MAX # TRUCKS
C              IF TRUCK NOT OF THIS UNIT, BYPASS
              IF(ITRUCK(J,4) .NE. L)GO TO 30
              IF(ITRUCK(J,2) .NE. 1 .AND. ITRUCK(J,2) .NE. 9)GO TO 30
C              HAVE TRUCK OF THIS UNIT PRINT INFO
              WRITE(14,20) J, ITRUCK(J,2), ITRUCK(J,3), ITRUCK(J,5),
             $      ITRUCK(J,6), ITRUCK(J,7)
              20  FORMAT (2X,6I7)
              30 CONTINUE
C
          RETURN
          END

```

```

WW.      SUBROUTINE UNTARV
          SUBROUTINE UNTARV (IPARM)
C**** EVENT UNTARV -- ARRIVAL OF TRUCK AT UNIT.
C              (ALSO MLRS(LEAVING ASP/ATP), AND SERVERS)
C              FOR UNIT TRUCKS, PUT IN QUEUE (MAY GENERATE
C              A RELOAD). FOR SERVERS, CALL SERVER TO
C              TO PUT THEM IN AVAILABLE QUEUE.
C
C          EVENT TYPE 8
C          CALLED BY MAINARM
C          CALLS      DEPASP, SERVER, IQ, PUTQUE, SCHED
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- TRUCK NUMBER (0 OR -1 FOR MLRS)
C**** IPARM(4) -- 666 FOR UNIT TRUCK FAILED ON RELOAD
C
C**** SCHEDULES      -- RELOAD IF DEMAND EXISTS.
C**** SCHEDULED BY  -- ASP, ASPAR1, ATP, DUALMX, RELOAD, SRFRER
C
C**** CHANGES      -- UNIT AM. J ON TRUCKS
C                   -- UNIT TRUCK QUEUE
C
C          INCLUDE LOG,LIST
C          DIMENSION IPARM(5)
C
C**** LOCAL VARIABLES :
C**** MIX  -- INDEX OF AMMO MIX
C**** IND  -- INDEX FOR IUNIT AMMO TYPE
C**** NUMAM -- AMMO TYPE FOR UNIT AMMO TYPE I
C**** NUMR  -- NUMBER OF ROUNDS OF TYPE NUMAM ON THE TRUCK
C**** IRESFL-- RESUPPLY FLAG (0 = NO RESUP, 1 = SCHED RESUP)
C**** NUMQ  -- UNIT TRUCK QUEUE NUMBER
C**** ISTAT -- 0 IF UNIT TRUCK FAILED ON RELOAD
C
C          NUNIT = IPARM(1)
C          NUMTK = IPARM(2)
C
C**** IF THIS IS A MLRS TRUCK LEAVING THE ASP OR ATP CALL DEPASP
C          IF(NUMTK .EQ. 0 .OR. NUMTK .EQ. -1) THEN
C              CALL DEPASP(IPARM)
C              RETURN
C          END IF
C
C**** CHECK IF THE TRUCK IS A SERVER
C          IF(ITRUCK(NUMTK,1) .EQ. 8 .OR. ITRUCK(NUMTK,1) .EQ. 9) THEN
C              CALL SERVER(IPARM)
C              RETURN
C          END IF
C
C**** INITIALIZE RELOAD FLAG
C          IRESFL = 0
C          ISTAT = 4

```

```

C**** DETERMINE AMMO MIX
      MIX = ITRUCK(NUMTK, 5)
      IF(MIX .LE. 0) THEN
        WRITE(6,10) NUMTK
10      FORMAT(1X,'UNTARV -- ZERO MIX ON TRUCK ',I3)
        RETURN
      END IF
C
C**** PUT TRUCK IN UNIT QUEUE
      NUMQ = IQ(1, NUNIT)
      CALL PUTQUE(NUMTK, NUMQ)
C
      IF(ITRUCK(NUMTK,3) .LE. 1 .OR. IPARM(4) .EQ. 666) ISTAT = 0
      ITRUCK(NUMTK, 3) = 1
C
C**** ADD AMMO TO UNIT AVAILABLE AMMO AND CHECK FOR GENERATING RELOAD
      DO 40 I = 1, LPPAR(6)
        IND = I*13 - 5
        NUMAM = IUNIT(NUNIT, IND)
        IF(NUMAM.EQ.0) GO TO 40
C
C**** IF NO AMMO OF THIS TYPE ON TRUCK GO TO 40
        NUMR = (IMIX(MIX,NUMAM) * ITRUCK(NUMTK,6) + 9999) / 10000
        IF(NUMR .LE. 0) GO TO 40
C
C**** HAVE THIS TYPE OF AMMO, ADD TO UNIT
        IF(ISTAT .NE. 4) GO TO 25
        IUNIT(NUNIT,IND+8) = IUNIT(NUNIT,IND+8) + NUMR
        IF(ISTAT.EQ.4) IUNIT(NUNIT,IND+12)=(IUNIT(NUNIT,IND+12))-NUMR
        ITRUCK(NUMTK,10) = ITRUCK(NUMTK,10) + 1 @ 1 OF ARVLS FROM ATP-ASP
C
C**** IF NOT A FARP GO TO 30
25      IF(IUNIT(NUNIT,1).NE.8) GO TO 30
        IF(IUNIT(NUNIT,IND+4).GT.IUNIT(NUNIT,IND+1)*IUNIT(NUNIT,I
$ND+7)+IUNIT(NUNIT,IND+1)*IUNIT(NUNIT,IND+6)) GO TO 40
C**** SET RELOAD FLAG
        IRESFL=1
        GO TO 40
C**** IF RELOAD IS NOT REQUIRED GO TO 40; ELSE SET RELOAD FLAG=IRESFL
30      IF(IUNIT(NUNIT,IND+7) * IUNIT(NUNIT,IND+1) .LE.
$ IUNIT(NUNIT,IND+4))GO TO 40
C
C**** SCHEDULE RELOAD FLAG
        IRESFL = 1
40      CONTINUE
        IPARM(4) = 0
        IF(IRESFL .EQ. 1) CALL SCHED(2, IPARM, TIME)          @ RELOAD
C
      RETURN
      END

```

```

.XX.      SUBROUTINE UNTDEF
          SUBROUTINE UNTDEF (IPARM)
C**** EVENT UNTDEF -- DEPARTURE OF TRUCK FROM UNIT.
C              (HAVING BEEN EMPTIED IN A RELOAD) MAY
C              CHANGE MIX ON THE WAY TO ATP/ASP
C
C          EVENT TYPE 3
C          CALLED BY MAINARM
C          CALLS      OPERA, INTRDK, SCHED
C
C**** J. FOX      JAN 79
C
C**** IPARM(1) -- UNIT NUMBER
C**** IPARM(2) -- TRUCK NUMBER
C
C**** SCHEDULES   -- ATPARV, ARRIVAL OF TRUCK AT ATP  OR
C              -- ASPARV, ARRIVAL OF TRUCK AT ASP
C
C**** SCHEDULED BY -- DUALMX, RELOAD
C
C**** CHECKS      -- DELAY IN ARRIVAL TIME AT ATP OR ASP DUE
C              TO MTBF AND INTERDICTION.
C
C**** CHANGES    -- UNIT TRUCK QUEUE
C          IPARM(4) NOW SET TO AMMO MIX ON TRUCK FOR ASP ARRIVALS
C
C          INCLUDE LOG,LIST
C          DIMENSION IPARM(5)
C
C**** LOCAL VARIABLES
C**** MIX        -- THE AMMO MIX INDEX
C**** DIST       -- DISTANCE TO ATP OR ASP
C**** ITKTYP     -- TRUCK TYPE
C**** TVLTIM     -- ROAD TRAVEL TIME
C**** IASPF6     -- LOCAL FLAG  = 1 IF GOING TO ASP
C              = 2 IF GOING TO ATP
C
C**** TOTIM      -- TIME OF ARRIVAL AT ATP OR ASP
C**** RNEED      -- REAL NEED PER AMMO TYPE
C**** PERSV      -- USED(WITH RNEED) TO FIND WORST NEEDED TYPE
C**** JTYP       -- UNIT TYPE
C
C          NUNIT = IPARM(1)
C          NUMTK = IPARM(2)
C
C          FIND THE UNIT TYPE
C          JTYP = IUNIT(NUNIT,1)
C
C**** DETERMINE AMMO MIX INDEX
C          MIX = ITRUCK(NUMTK, 5)
C          IF(MIX .EQ. 10 .OR. MIX .EQ. 40) THEN
C              NUMAM = 10
C              GO TO 28
C          END IF
C

```

```

DO 5 I = 1,LPPAR(1)
IF(IMIX(MIX,I) .GT. 0)THEN
    NUMAM = I
    GO TO 6
END IF
3 CONTINUE
6 PERSV = 10000.0
DO 10 KK=1,LPPAR(6)
C     THIS LOOP FIGURES NEED FOR ALL AMMO TYPES
C     SEND TRK AFTER TYPE NEEDED WORST
    K = 13 * KK - 5
C     FIND SMALLEST PERCENT OF AMMO AVAILABLE AND STORE IT
    IF(IUNIT(NUNIT,K) .EQ. 0)GO TO 10
    A = IUNIT(NUNIT,K+4)           @ CURR SUPP
    B = IUNIT(NUNIT,K+1)           @ WPNS ALV
    C = IUNIT(NUNIT,K+7)           @ BAL
    D = IUNIT(NUNIT,K+8)           @ ON TRKS
    E = IUNIT(NUNIT,K+12)          @ ON THE WAY
    IF(B .EQ. 0 .OR. C .EQ. 0)GO TO 10
C * * TO COMPENSATE FOR TWO SYSTEMS USING TOW:
    Z = 1
    IF(IUNIT(NUNIT,K) .EQ. 2)Z = 0.5
    RNEED = (A + Z*D + Z*E)/(B * C)
    IF(RNEED .LT. PERSV)THEN
        PERSV = RNEED
        NUMAM = IUNIT(NUNIT,K)
    END IF
10 CONTINUE
    IF(ITRUCK(NUMTK,1) .EQ. 1)THEN
        JLOW = 1
        JHIGH = 30
    ELSE
        JLOW = 31
        JHIGH = 60
    END IF
    DO 15 I = JLOW,JHIGH
        IF(IMIX(I,NUMAM) .GT. 0)THEN
            MIX = I
            GO TO 20
        END IF
15 CONTINUE
20 ITRUCK(NUMTK,5) = MIX           @ GET WORST NEEDED TYPE
C ***** ADD THE ROUNDS DUE IN TO THE UNIT
28 DO 25 I = 1,LPPAR(6)
    K = I * 13 - 5
    IF(IUNIT(NUNIT,K) .EQ. 0)GO TO 25
    IUNIT(NUNIT,K+12) = IUNIT(NUNIT,K+12) + IMIX(MIX,IUNIT(NUNIT,K))
25 CONTINUE
C***** IF MIX CONTAINS AMMO OTHER THAN THAT AT ATP (1-10) GO TO ASP
    IF(NUMAM .GT. 10 .OR. IUNIT(NUNIT,2) .EQ. 0)GO TO 35
C
C***** TRUCK BOUND FOR ATP. LOOK UP DISTANCE TO ATP (DIST)
    DIST = IUNIT(NUNIT, 4)
    IASPG = 2

```

```

C
C**** TRUCK BOUND FOR ASP. LOOK UP DISTANCE TO ASP (DIST)
35 DIST = IUNIT (NUNIT, 5)
IASPFG = 1
C
C**** DETERMINE TRUCK TYPE (ITKTYP)
40 ITKTYP = ITRUCK(NUMTK, 1)
C
C**** DETERMINE ROAD TRAVEL TIME
TVLTIM = 60. * DIST / ITYPE(ITKTYP, IDAY+1)
C**** UPDATE TRUCK STATUS CODE
ITRUCK(NUMTK, 3) = 4
C
C**** COMPUTE DELAY DUE TO INTERDICTION (TMIND)
CALL INTRDK (NUMTK, TMIND)
IF(TMIND .GT. 0) THEN
    IF(IASPFG .EQ. 1)JUNIT(JTYP,5)=JUNIT(JTYP,5)+1
    IF(IASPFG .EQ. 2)JUNIT(JTYP,2)=JUNIT(JTYP,2)+1
    ITRUCK(NUMTK,6) = 0
    IPARM(3) = IUNIT(NUNIT,3)
    IPARM(4) = MIX
    CALL SCHED(5, IPARM, TIME+TMIND)
    RETURN
END IF
C
C**** COMPUTE DELAY DUE TO FAILURE (TFAIL)
CALL OPERA (NUMTK, TVLTIM, TFAIL)
C**** COMPUTE TIME OF ARRIVAL
TOTIM = TIME + TMIND + TFAIL + TVLTIM
IF(IASPFG .EQ. 2)GO TO 49
C
C**** SCHEDULE ASP ARRIVAL.
IPARM(3) = IUNIT(NUNIT,3)
IPARM(4) = MIX
C ADD ONE TO THE NUMBER OF TRUCKS TRAVELING TO THE ASP
JUNIT(JTYP,5) = JUNIT(JTYP,5) + 1
C ADD TO TRUCKS KILLED OR FAILED
IF(TFAIL .GT. 0) JUNIT(JTYP,7) = JUNIT(JTYP,7) + 1
C ADD THE TRAVEL TIME TO THE ACCUMULATIVE TRAVEL TIME
JUNIT(JTYP,8) = JUNIT(JTYP,8) + TVLTIM
CALL SCHED (5, IPARM, TOTIM) @ ASPARV
GO TO 50
C
C**** SCHEDULE ATP ARRIVAL.
49 IPARM(3) = IUNIT(NUNIT,2)
IPARM(4) = MIX
C ADD ONE TO THE NUMBER OF TRUCKS TRAVELING TO THE ATP
JUNIT(JTYP,1) = JUNIT(JTYP,1) + 1
IF(TFAIL .GT. 0) JUNIT(JTYP,3) = JUNIT(JTYP,3) + 1
JUNIT(JTYP,4) = JUNIT(JTYP,4) + TVLTIM
CALL SCHED (4, IPARM, TOTIM) @ ATPARV
C
50 RETURN
END

```

yy. SUBROUTINE GETONE

```
      SUBROUTINE GETONE(NQ, NUMTK, IFLAG)
C
C   GETONE GETS A GIVEN TRUCK 'NUMTK' OUT OF QUEUE NQ.
C   IF IT IS NOT IN THAT QUEUE, IFLAG WILL RETURN NON-ZERO.
C
C   CALLED BY TRKPRT(IN THE EDIT PROGRAM).
C
C   R. CUNNINGHAM                                MAY 83
C
      INCLUDE QUEPNT,LIST
      INCLUDE QUENUM,LIST
      NEXT = IHEAD(NQ)
      IFLAG = 1
      IF(NEXT .EQ. 0) RETURN
      IF(NEXT .EQ. NUMTK) THEN
         IHEAD(NQ) = ITEMS(NEXT)
         ITEMS(NEXT) = 0
         IFLAG = 0
         RETURN
      END IF
C
C
C   100 CONTINUE
C
      IF(NEXT .NE. NUMTK) THEN
         LAST = NEXT
         NEXT = ITEMS(LAST)
         IF(NEXT .EQ. 0) RETURN                                @ NOT IN QUE
         GO TO 100
      ELSE
C
C
C         ITEMS(LAST) = ITEMS(NEXT)
C         ITEMS(NEXT) = 0
C         IFLAG = 0
C         FOUND!!
      END IF
      RETURN
      END
```

2. Edit Program

```

C      PROGRAM EDIT
C      (MAIN PROGRAM FOR EDITING BETWEEN CIs)
C
C      CALLS EDITD, GETQUE, PRINT, PUTQUE, TRKFUT
C      CHARACTER*10 AUNIT
C      INCLUDE LOG,LIST
C      INCLUDE QUENUM,LIST
C      INCLUDE QUEPNT,LIST
C      INCLUDE AUNIT,LIST
C      DIMENSION IVAL(4)
C      CHARACTER*3 IANS
C      DEFINE FILE 13(75,4,U,ITZ)
C
C      READ(3) IATP,IASP,IUNIT,ITRUCK,ITYPE,IMIX,
$      INTER,IRSTHE,IATPSD,IDAY,TIME,IATPRM,ICSA,
$      LPPAR,IASPRM,LUOUT,TCIST,TOTLNG,LOUS,INHEAD,ITERA,
$      AUNIT,JUNIT,JATP,JASP,IATPSR,IASPSR,IAMLVL,ISFRV
C      DO 100 I=1,75
C          READ(13,I)(IVAL(J),J=1,4)
C          IUNIT(I,2)=IVAL(1)
C          IUNIT(I,3)=IVAL(2)
C          IUNIT(I,4)=IVAL(3)
C          IUNIT(I,5)=IVAL(4)
100  CONTINUE
C      PRINT*, ' ZERO COUNTERS? ( YES OR NO ) '
C      READ(5,90)IANS
C      IF(IANS.EQ. 'N ' .OR. IANS.EQ. 'NO ')GO TO 300
C          TCIST = INT(TIME)
C          INTER(1) = 0
C          INTER(2) = 0
C          DO 200 I = 1,10
C              ISERV(I) = 0
C              IATP(I,16) = 0           @ SERVER WITHHOLD COUNTER
C              IASP(I,16) = 0           @ SERVER WITHHOLD COUNTER
200  CONTINUE
C      300 PRINT*, ' MODIFY INTER, IDAY, TOTLNG, ASP STATUS...'
C      10 WRITE(6,20)
C      15 FORMAT('      (1)-EDIT DATA FILES',/,
$      '      (2)-UPDATE FA CURRENT SUPPLY (TO 100 PER CENT)',/,
$      '      (3)-CLOSE ATP',/,
$      '      (4)-MODIFY TRK QUE',/,
$      '      (5)-PRINT TRUCK RQUEUES',/,
$      '      (6)-STOP')
C      20 FORMAT('      ???')
C      READ(5,*,ERR=25) IOPT
C      IF(IOPT.LT.1.OR.IOPT.GT.6)GO TO 25
C      GO TO (30,40,45,50,60,90),IOPT
C      25 WRITE(6,15)
C      GO TO 10
C
C      30 CALL EDITD
C      GO TO 10
C

```

```

      IF (IUNIT(K,1) .LT. 4 .OR. IUNIT(K,1) .GT. 5) GO TO 41
      DO 41 I = 8,125,13
        IUNIT(K,I+2) = 0
        IUNIT(K,I+3) = 0
        IUNIT(K,I+4) = IUNIT(K,I+1) * IUNIT(K,I+7)
41    CONTINUE
      GO TO 10
C
C      **** THIS CODE MOVES (CFA) ATP SIPS AND AMMO ****
45    PRINT *, ' ENTER (CFA) ATP NUMBER TO BE CLOSED '
      READ (5,*) NATP
      NATPQ = 75 + NATP
47    CALL GETQUE(NUMSP,NATPQ)
      IF (NUMSP .EQ. 0) GO TO 10
      PERCNT = ITRUCK(NUMSP,6)
      IF (PERCNT .LE. 0.) THEN
        CALL PUTQUE(NUMSP,176)
        GO TO 47
      END IF
      MIX = ITRUCK(NUMSP,5)
      NUMAM = MIX - LPPAR(3)
      IAMCOD = NUMAM * 3 + 18
      IF (MIX .EQ. 80) IAMCOD = 51 @ FUZES
      LEAST = 999999
      DO 48 J = 1,3
C      AMAV = ONHAND - DEMAND + ON-THE-WAY
        AMAV = IATP(J,IAMCOD) - IATP(J,IAMCOD + 1) + IATP(J,IAMCOD + 2)
        IF (AMAV .GE. LEAST) GO TO 48
        LEAST = AMAV
        LATP = J
48    CONTINUE
      NUMRDS = IMIX(MIX,NUMAM) * PERCNT / 10000
      IATP(NATP,IAMCOD) = IATP(NATP,IAMCOD) - NUMRDS
      IATP(LATP,IAMCOD) = IATP(LATP,IAMCOD) + NUMRDS
      CALL PUTQUE(NUMSP,LATP + 75)
      GO TO 47
C
50    CALL TRKPUT
      GO TO 10
C
60    CALL PRINT
      GO TO 10
C
90    FORMAT(A3 )
C
90    WRITE(4) IATP,IASP,IUNIT,ITRUCK,ITYPE,IMIX,INTER,
$ IRSTME,IATPSD,IDAY,TIME,IATPAM,ICSA,LPPAR,IASPAM,LUOUT,
$ TCIST,TCILNG,LOOK,IHFAD,ITEMS,AUNIT,JUNIT,JATP,IASP,
$ IATPSP,IASPSP,IAMLVL,ISERV
      PRINT*, ' SSG ARMPL.DF WILL SYN THIS REPORT '
      STOP
      END

```

```

a.      SUBROUTINE PRINT
        SUBROUTINE PRINT
C**** PRINTS OUT THE CONTENTS OF EVERY TRUCK QUEUE
C
C      CALLED BY EDIT
C      CALLS GETQUE, PUTQUE, NXTQUE
C
C**** D. HILLIS APR 79
C
        include quenum,list
        include queent,list
        DIMENSION NTRK(200)
        DO 70 I=1,176
            CALL NXTQUE(IFIRST,I)
            IF(IFIRST.EQ.0) GO TO 50
            DO 10 J=1,200
                CALL GETQUE(NTRK(J),I)
                CALL PUTQUE(NTRK(J),I)
                CALL NXTQUE(INEXT,I)
                IF(INEXT.EQ.IFIRST) GO TO 20
            10  CONTINUE
C
        20  WRITE(2,30) I
        30  FORMAT(/,5X,'QUEUE ',I3,' TRUCKS')
            WRITE(2,40) (NTRK(K),K=1,J)
        40  FORMAT(10(1X,I4))
            GO TO 70
        50  WRITE(2,30) I
            WRITE(2,60)
        60  FORMAT(5X,'NONE')
        70  CONTINUE
C
        RETURN
        END

```

3. Convert Program

```
      PROGRAM CONVERT
C
C      THIS PROGRAM CONVERTS TEVENTS, AN EVENTS FILE, WHICH IS IN
C      CHARACTER FORM TO A BINARY FILE FOR USE IN THE ARM PROGRAM
C
C      *****
C      DIMENSION IRAY1(500),IRAY2(500),IRAY3(500),IRAY4(500)
C      DIMENSION IRAYS(500),IRAY6(500),ARRAY(500)
C
C
C      DO 100 I=1,500
C          IRAY1(I)=0                @ EVENT TYPE
C          IRAY2(I)=0                @ IPARM(1)
C          IRAY3(I)=0                @ IPARM(2)
C          IRAY4(I)=0                @ IPARM(3)
C          IRAYS(I)=0               @ IPARM(4)
C          IRAY6(I)=0               @ IPARM(5)
C          ARRAY(I)=0.              @ EVENT TIME
100    CONTINUE
C
C      DO 500 I=1,500
C          READ(12,*)IRAY1(I),IRAY2(I),IRAY3(I),IRAY4(I),IRAYS(I),
C          ZIRAY6(I),ARRAY(I)
500    CONTINUE
C          WRITE(11)IRAY1,IRAY2,IRAY3,IRAY4,IRAYS,IRAY6,ARRAY
C          PRINT*, 'FILE 11 IS NOW COMPLETE'
C
C      STOP
C      END
```

4. ADDEVT Program

```

      PROGRAM ADDEVT
C
C      THIS PROGRAM ALLOWS ENTRY OF EVENTS FOR ARM
C      INTERACTIVELY SEPARATE FROM THE MAIN PROGRAM.
C      THESE EVENTS ARE THEN STORED IN A FILE 12 CALLED
C      TEVENTSCI--. THIS IS A SOURCE FILE WHICH CAN
C      THEN BE TAKEN INTO CTS EDITOR AND CORRECTIONS MADE.
C
C*****
C      VARIABLES USED:
C      ITYPE....THE NUMBER OF THE EVENT
C      IPARM1,IPARM2,IPARM3,IPARM4,IPARM5
C      ....THE ASSOCIATED PARAMETERS OF THE EVENT
C
C      FILES USED: 12 THE SOURCE FILE WHICH IS CALLED
C      WHEN ALL EVENTS ARE ENTERED FROM KEYBOARD.
C
C*****
C      DIMENSION ITYPE(500),IPARM1(500),IPARM2(500),IPARM3(500)
C      DIMENSION IPARM4(500),IPARM5(500),STIME(500)
C      CHARACTER*3 IANSW
C
C      DO 100 I=1,500
C          ITYPE(I)=0
C          IPARM1(I)=0
C          IPARM2(I)=0
C          IPARM3(I)=0
C          IPARM4(I)=0
C          IPARM5(I)=0
C          STIME(I)=0.
100  CONTINUE
C
C      SET INITIAL CONTER TO 1 AND PROMPT FOR EVENTS TO BE ENTERED
C
C      ICOUNT=1
C      PRINT*, 'ENTER EVENT TYPE,PARMS,TIME,SEPARATED BY COMMAS'
C      PRINT*, 'ENTER 0,0,0,0,0 TO COMPLETE ADDITION OF EVENTS'
2    PRINT*, '?'
C
C      READ*, ITYPE(ICOUNT),IPARM1(ICOUNT),IPARM2(ICOUNT),IPARM3(ICOUNT),
C      ZIPARM4(ICOUNT),IPARM5(ICOUNT),STIME(ICOUNT)
C      IF (ITYPE(ICOUNT).EQ.0) GO TO 3
C      IF (ITYPE(ICOUNT).LT.0 .OR. ITYPE(ICOUNT).GT.17) THEN
C          PRINT*, 'INCORRECT EVENT TYPE'
C          GO TO 2
C      END IF
C      ICOUNT=ICOUNT+1
C      GO TO 2
C
3    PRINT*, 'DO YOU WISH TO SEE EVENTS---( Y OR N)'
C      READ(5,22) IANSW

```

```

      IF(IANSW.EQ.'Y' .OR. IANSW.EQ.'Y ') THEN
        DO 111 J=1,ICOHNT
          PRINT 11,ITYPE(J),IPARM1(J),IPARM2(J),IPARM3(J),IPARM4(J),
            Z  IPARM5(J),STIME(J)
11      FORMAT(2X,I3,T7,'IPARMS= ',5I5,' SCHEDULED TIME = ',F8.1)
111      CONTINUE
        END IF
      C
      C***** NOW PUT EVENTS ON FILE IF DESIRED
      PRINT*, 'PUT EVNTS ON FILE 12---Y OR N'
      READ 22,IANSW
      IF(IANSW.EQ.'Y' .OR. IANSW.EQ.'Y ') THEN
        DO 222 I=1,500
          WRITE(12,*)ITYPE(I),IPARM1(I),IPARM2(I),IPARM3(I),IPARM4(I),
            Z  IPARM5(I),STIME(I)
222      CONTINUE
          PRINT*, 'EVENTS NOW ON FILE 12'
        END IF
      C
      STOP
      END

```

5. Information

```

C      SUBROUTINE INFORM
C              AMMUNITION RESUPPLY MODEL (ARM)
C
C
C
C
C
C      ***** AUNIT *****
C
C      AUNIT(75,2)
C              75 UNITS
C              2 ALPHA FIELDS/UNIT
C      COLUMN 1 = UTM COORDINATES
C      COLUMN 2 = ALPHA UNIT NAMES
C
C
C
C      ***** ATP DATA *****
C
C      IATP (10,53)
C              10 ATP
C              53 WORDS EACH AS FOLLOWS:
C
C      1. DISTANCE TO CSA
C      2. DISTANCE TO ASP
C      3. DIS TO DAO
C      4.
C      5. number of arriving s & p tractors used for return
C          of empty trailers
C      6. ASSOCIATED ASP NUMBER
C      7. QUE FOR ASP S&P
C      8. NO OF MLRS TRUCKS BEING SERVED
C      9. que for csa s&p
C      10. SERVERS QUEUE NUMBER
C      11. ARTILLERY QUEUE NUMBER
C      12. MANUEVER UNIT QUEUE NUMBER
C      13. CONVOY COUNTER AT CSA (IF .GE. 3, SCHEDULER CONVOY FORWARD)
C      14. NUMBER TRUCKS IN ARTILLERY QUEUE
C      15. NUMBER TRUCKS IN MANUEVER UNIT QUEUE
C      16. SERVER REMOVAL KOUNTER
C      17. NUMBER OF CONVOYS SENT FROM CSA
C      18.
C      19.
C      20. NUMBER OF TIMES A SERVER NOT AVAILABLE
C      21. CURRENT AMMO SUPPLY, AMMO 1      (NUMAM*3+18)
C      22. QUEUE AMMO DEMAND, AMMO 1      (NUMAM*3+19)
C      23. ON-THE-WAY (FROM DAO)          (NUMAM*3+20)
C      24-26      AMMO 2
C      27-29      ammo 3
C      30-32      ammo 4
C      33-35      ammo 5
C      36-38      ammo 6
C      39-41      ammo 7

```

45-47 ammo 9
48-50 ammo 10
51 FUZES

IATPSD(5) -- ATP SERVICE DATA

1. lowest asp-atp round-robin s&p number
2. ATP 1ST PRIORITY S & P QUEUE
3. ATP 2D PRIORITY S & P QUEUE
4. CFA ATP OWNER NUMBER
- 5.

***** IATPSP TO OBTAIN ASP % REPLENISHMENT *****

IATPSP(10,22)

11 ATP

1 - 11 #CSA S&Ps ARRIVING BY AMMO TYPE

12 - 22 #ASP S&Ps ARRIVING BY AMMO TYPE

***** ASP DATA *****

IASP (10,110)

10 ASP

110 WORDS EACH AS FOLLOWS:

1. DISTANCE TO CSA
2. ASP STATUS:
-1 INACTIVE 0 ACTIVE/NO CONVOYS 1 ACTIVE/CONVOYS
3. CUMULATIVE HELICOPTER COUNTER
4. QUEUE FOR CSA-ASP S&P TRUCKS
5. NUMBFR TRUCKS TO CSA
6. NUMBER OF EMPTY S & Ps AT ASP
7. SERVERS QUEUE NUMBER
8. NUMBER MLRS BEING SERVED
9. ROUTINE QUEUE NUMBER
10. MLRS QUEUE NUMBER
11. REAR ASP NUMBER
12. NUMBER TRUCKS IN ROUTINE QUEUE
13. NUMBER TRUCKS IN MLRS QUEUE
14. CONVOY COUNTER AT CSA (IF .GE. 7, SCHEDULE CONVOY FORWARD)
15. HELICOPTER SERVER COUNTER
16. SERVER REMOVAL COUNTER
17. NUMBER OF CONVOYS SENT FROM CSA
18. QUEUE FOR ASP-ATP ROUND ROBIN S & P S
- 19.
20. NUMBER OF TIMES A SERVER NOT AVAILABLE FOR UNIT TRUCK
21. CURRENT AMMO SUPPLY, AMMO 1 (NUMAM*3+18)
22. QUEUE AMMO DEMAND, AMMO 1 (NUMAM*3+19)
23. AMMO ON THE WAY (FROM CSA OR DAD), AMMO 1 (NUMAM*3+20)

***** IASPS TO OBTAIN # S & P ARRIVALS BY AMMO TYPE *****

IASPS(10,30)

10 ASP

30 AMMO TYPES

IATPAM (10,40)

AMMO REMOVED FROM ATP

10 ATP, 10 AMMO TYPES

1 - 10 10 TON TRUCKS SERVICED (HLRS W/ TLR)
11 - 20 5 TON TRUCKS SERVICED (HLRS W/O TLR)
21 - 30 10 TON TRUCKS BUMPED
31 - 40 5 TON TRUCKS BUMPED

IASPAH(10,120)

AMMO REMOVED FROM ASP

10 ASP, 30 AMMO TYPES

1 - 30 10 TON TRUCKS SERVICED
31 - 60 5 TON TRUCKS SERVICED
61 - 70 ATP S & P SERVICED
71 - 80 # CSA-ATP ARRIVALS
90 # HELI LOADS REMOVED
91 - 120 # CSA-ASP S & PS ARRIVED

IAMLVL(2,30)

STOCKAGE OBJECTIVES

1 / 1 - 10 ATP S O
2 / 1 - 30 ASP S O
1,29 MAX ATP STOCKAGE %
1,30 MAX ASP STOCKAGE %

ISERV(10)

SERVER MANIPULATIONS

1. # ATP SERVERS TO BE HELD (AS FOR DISPLACEMENT - ALL ATPS)
2. # ASP SERVERS TO BE HELD (AS FOR DISPLACEMENT - ALL ASPS)
3. ATP SERVER HOLD QUEUE
4. ASP SERVER HOLD QUEUE
5. INTERDICTED ATP #
6. INTERDICTED ASP #
7. MINUTES SERVERS TO BE HELD IN HOLD QUEUE AT ATP

9.
10.

IUNIT (75,142)

75 UNITS
142 WORDS EACH AS FOLLOWS:

1. TYPE
2. ATP NUMBER
3. ASP NUMBER
4. DISTANCE TO ATP
5. DISTANCE TO ASP
6. TIME THAT THE LAST TRUCK FOR THIS UNIT WAS INTRODUCED
7. HELD MISSIONS RECEIVED
8. FIRST AMMO TYPE
9. NUMBER WEAPONS ALIVE, FIRST AMMO TYPE
10. NUMBER WEAPONS SHORT AMMO, FIRST AMMO TYPE
11. NUMBER ROUNDS SHORT, (WPNS) FIRST AMMO TYPE
12. CURRENT AMMO SUPPLY, (WPNS) FIRST AMMO TYPE
13. ROUTINE RESUPPLY LEVEL, (PER WPN) FIRST AMMO TYPE
14. CRITICAL RESUPPLY LEVEL, (PER WPN) FIRST AMMO TYPE
15. BASIC AMMO LEVEL, (PER WPN) FIRST AMMO TYPE
16. AMMO ON TRUCKS, FIRST AMMO TYPE
17. NUMBER OF WEAPONS KILLED AT THE END OF CI, 1ST AMMO TYPE
18. NUMBER OF WEAPONS SHORT AMMO FIRST AMMO TYPE
19. TOTAL ROUNDS SHORT THROUGH WHOLE CI FIRST AMMO TYPE
20. STORAGE FOR NO. RDS RESUPPLY ENROUTE
- 21-33 ,SECOND AMMO TYPE
- 34-46 ,THIRD AMMO TYPE
- 47-59 ,FOURTH AMMO TYPE
- 60-72 ,FIFTH AMMO TYPE
- 73-85 ,SIXTH AMMO TYPE
- 86-98 ,SEVENTH AMMO TYPE
- 99-111 ,EIGHTH AMMO TYPE
- 112-124 ,NINTH AMMO TYPE
- 125-137 ,TENTH AMMO TYPE
138. NUMBER OF HELICOPTERS ASSIGNED
139. = 0 IF SINGLE PULSE DEMAND PER CI
- = 1 IF MULTIPLE PULSES PER CI
- 140-142 COUNTERS

IRSTME(23,3) -- RESUPPLY TIME DATA
23 TYPES OF AMMO
3 WORDS EACH AS FOLLOWS

1. WEAPON SET-UP TIME
2. LOAD TIME PER ROUND
3. TRAVEL TIME TO WEAPON

C ITRUCK (1400,15)
 C 1400 TRUCKS
 C 15 WORDS EACH AS FOLLOWS:
 C
 C 1. TRUCK TYPE
 C 2. MISSION TYPE
 C 3. STATUS TYPE
 C 4. OWNER NUMBER
 C 5. AMMO MIX NUMBER // SERVER OFF-LOAD TIME AT ASP
 C 6. PERCENT LOADED // SERVER LOAD TIME
 C 7. TIME SINCE LAST FAILURE
 C
 C TRUCK COUNTERS: UNIT CSA-ATP CSA-ASP ASP-ATP
 C 8. 1 #FAILURES
 C 9. 2 #INTERDICTIONS
 C 10. 3 #ARRIVALS FROM: ASP/ATP CSA CSA ASP
 C 11. 4 #RELOADS # * @ CSA #THRU-PUTS AT ASP
 C 12. 5 QUEUE TIME ASP/ATP CSA CSA ASP
 C 13. 6 (* BUMPED (TIME AT IN (* AT ASP)
 C TO 2D ASP) ATP/ASP Q)
 C 14. 7 (*BUMPED #XEMPTYASP (* BUMPED
 C TO ASP) #XEMPTYASP TO REAR ASP)
 C 15. 8 (* >3 IN
 C MLRS QUE
 C
 C SERVER COUNTERS:
 C 1 #FAILURES
 C 2 #S & P OFF-LOADS
 C 3 #ASP-ATP S & P LOADUPS
 C 4 # 5TON LOAD UPS
 C 5 # 10TON LOAD UPS
 C 6 # LOAD UPS FROM CSA S & PS
 C 7
 C 8
 C
 C ITYPE (9,6)
 C 9 TYPES OF TRUCKS
 C 6 WORDS FOR EACH TYPE TRUCK AS FOLLOWS:
 C
 C 1. SECONDARY ROAD NIGHT SPEED (UNIT TO ASP,ATP)
 C 2. SECONDARY ROAD DAY SPEED (UNIT TO ASP,ATP)
 C 3. HIGHWAY NIGHT SPEED
 C 4. HIGHWAY DAY SPEED
 C 5. MTBF
 C 6. MTRR
 C
 C TRUCK QUEUES: QUEUE TYPE:
 C
 C 1 - 75 1 AT EACH UNIT
 C 76 - 85 2 AT ATPS FOR CSA-ATP S & PS
 C 86 - 95 3 AT ATPS FOR ASP-ATP S & PS
 C 96 - 105 4 AT ATPS FOR UNIT ARTILLERY SERVER
 C 106 - 115 5 AT ATPS FOR UNIT MANUEVER SERVER

C	126 - 135	7	AT ASPS FOR CSA-ASP S & PS
C	136 - 145	8	AT ASPS FOR ROUTINE UNIT TRUCKS
C	146 - 155	9	AT ASPS FOR MILK UNIT TRUCKS
C	156 - 165	10	at ASPs for servers
C	166 - 175	11	AT ASPS FOR ASP-ASP S & PS
C	176	12	AT CSA FOR CSA S & PS

***** AMMO DATA *****

IMIX(91,32)

91 AMMO MIXES

1 - 30 10TON MIXES

31 - 60 5TON MIXES

61 - 90 S & P MIXES

91 HELICOPTER MIX

32 WORDS FOR EACH MIX AS FOLLOWS:

1-30 NUMBER ROUNDS OF EACH AMMO TYPE

31. LOAD TIME AT CSA/ATP

32. LOAD TIME AT ASP

***** MISC DATA *****

EVENT SCHEDULING

COMMON/EVENTS/ JSTAT(6), JEVD(2048,4), IEVS(5,2048)

QUEUE DATA

COMMON /QUENUM/ IHEAD(176)

COMMON /QUEPNT/ ITEMS (1400)

INTERDICTION DATA - COMMON INTER(10)

1 COUNTER FOR ZONE 1 TRUCKS KILLED IN INTRDK

2 COUNTER FOR ZONE 2 TRUCKS KILLED IN INTRDK

3 NUMBER OF TRUCKS TO BE KILLED IN ZONE ONE

4 NUMBER OF TRUCKS TO BE KILLED IN ZONE 2

5 TIME TO REPLACE TRUCK IN ZONE 1

6 TIME TO REPLACE TRUCK IN ZONE 2

7 MODULO OF TRUCKS TO BE KILLED IN ZONE 1

8 MODULO OF TRUCKS TO BE KILLED IN ZONE 2

C 10 NUMBER OF ZONE TWO TRUCKS ENTERING INTRON
 C
 C IDAY 1 = DAY , 0 = NIGHT
 C
 C ICSA(3,32) -- NUMBER OF ROUNDS BY AMMO TYPE FROM CSA
 C
 C (1,1-11) # S&Ps TO ATP
 C (2,1-30) # S&Ps TO ASP
 C (3,1-30) CUMULATIVE AMMO DEMAND OF ALL UNIT
 C
 C 1-30 AMMO TYPES
 C 2,31 COUNTER FOR EMPTY S & Ps AT CSA
 C 3,32 COUNTER FOR EMPTY POL TRUCKS AT CSA
 C
 C LPPAR(1) -- TOTAL NUMBER OF AMMO CODES (30)
 C LPPAR(2) -- NUMBER OF AMMO CODES AT ATP (10)
 C LPPAR(3) -- NUMBER OF MANEUVER UNIT AMMO CODES AT ATP(3)
 C LPPAR(4) -- NUMBER OF TRANSPORTS(TRUCKS) (LT 1400)
 C LPPAR(5) -- NUMBER OF HELICOPTERS AVAILABLE (10)
 C LPPAR(6) -- NUMBER OF AMMO TYPES AT UNITS(LE 10)
 C LPPAR(7) -- NUMBER TO SUBT FROM STON MIX TO GET AMMO TYPE(NUMAM)
 C LPPAR(8) -- NUMBER TO SUBT FROM S&P MIX TO GET AMMO TYPE(NUMAM)
 C
 C TCIST -- TIME OF START OF CI IN DECIMAL MINUTES
 C C*%Z*%Z*%Z*%Z* TCIST MUST BE .0005 FOR CI01 !!!!
 C
 C TCILNG -- TIME OF LENGTH OF CI IN DECIMAL MINUTES
 C
 C
 C TIME -- SIMULATION TIME IN MINUTES (DECIMAL)
 C
 C
 C ***** TYPE CODE DATA *****
 C
 C UNIT TYPE CODES:
 C
 C 1 TANK TASK FORCE
 C 2 MECH TASK FORCE
 C 3 ARMED CAV SQDN
 C 4 155 ARTY BTRY
 C 5 8 INCH ARTY BTRY
 C 6 MLRS BTRY
 C 7 DIVAD GUN PLT
 C 8 CBT AVN PLT
 C
 C
 C TRUCK TYPE CODES :

[illegible]

כ

C
C
C
C
C
C
C
C
C
C
C
C
C

C

C

[illegible]

3

```

C      2   IN ATP QUEUE
C      3   IN ASP QUEUE
C      4   IN TRANSIT  --  OR BUSY IF SERVER
C      5   UNIT TRUCK GOING FROM ATP TO ASP  --  SERVER MOVING TO NEW LOCATION
C      6   TRUCK AWAITING REPAIR
C      7   TRUCK DEAD (INTERDICTED)
C      8   IN CSA QUE
C      9   BUMPED TO 2D ASP  --  SERVER OF INTERDICTED ASP/ATP
C

```

```

C ***** data for turnaround times and queue waits *****
C

```

```

C ***** Junit *****
C

```

```

C   Junit(8,24)
C

```

```

C   for the 8 types of units the following
C

```

- C 1. number of trucks sent to the atp from the unit
- C 2. number of trucks killed on that move
- C 3. number failed on that move
- C 4. total travel time for all trucks on that move
- C 5. number of trucks sent to the asp from the unit
- C 6. number of trucks killed on that move
- C 7. number failed on that move
- C 8. total travel time for all trucks on that move
- C 9. number of trucks sent to the asp from the atp
- C 10. number killed on that move
- C 11. number failed on that move
- C 12. total travel time for all trucks on that move
- C 13. number of trucks sent to the unit from the atp
- C 14. number killed on that move
- C 15. number failed on that move
- C 16. total travel time for all trucks on that move
- C 17. number of trucks sent to the unit from the asp
- C 19. number failed on that move
- C 20. total travel time for all trucks on that move
- C 21. total time spent in reloading weapons
- C 22. total time available for unit trucks
- C 23. number of trucks killed during reload
- C 24.

```

C ***** Jatz *****
C

```

```

C   Jatz(10,6)
C

```

```

C   for the 10 different atps the following:
C

```

- C 1. number of trucks served by the maneuver queue
- C 2. total queue wait time for all trucks served
- C 3. maximum wait time for the maneuver queue
- C 4. trucks served through the artillery queue

```

c      5. maximum wait time for the artillery queue
c
c
c
c
c
c
c      ***** Jasp *****
c
c      Jasp(10,9)
c      for the 10 different asps the followings:
c
c      1. trucks served through the routine queue
c      2. total wait time for all trucks served
c      3. maximum wait time in the routine queue
c      4. trucks served through the mlrs queue
c      5. total wait time for all trucks served
c      6. maximum wait time in the mlrs queue
c      7. trucks served through the ASP-ATF queue
c      8. total wait time for all trucks served
c      9. maximum wait time in the ASP-ATF queue
c
c      end
c
c      end

```

6. Demand Generation Program

```

C          *** RANMOD ***
C  RANMOD creates and edits ammunition demand files for the
C  ARM model and generates new demand. Files are assigned
C  in an SSG runstream (RUNSG). RANMOD can run independently,
C  but only temporary files are created and data is not stored.
C          *****
C          DATA SECTION
C          *****
C  INV holds the input values:
C  INV:   no change, MOPP*100,   intensity, phase,
C         rets indx, change,    method,   1st unit,
C         last unit, cur unit,   1st copy,  last copy,
C         file trmnt, ammo code, no. of CIs
C  IWPV holds values which change for each wpn in the unit.
C  IWPV:  wpn sys code, max exp limit, min exp limit,
C         initial draw, phase draw,   exp rate diff.
C  IND holds the status for options available (CHNGS)
C  (1=do change; 0=don't change).
C  IND:   alive, short, wpn type, ah's, mnt ret, war res,
C         inst dmd, copy, print, save, demand, by ammo code,
C  IAMAT holds the attributes for the specified unit in
C  the specified CIs. It is the window for the files.
C  IAMAT: ammo code, wpn alive, wpn short, ah's, demand
C  KROW and KCOL = row and col indices for IAMAT.
C  IANS used primarily for user responses.
C  LEAP holds the offset for the INFO subroutine.
C  MNT and KMB: mnt rets & war reserves; RETS: total rets
C  WPN holds the name of each weapon in the unit.
C  AMON holds the name of any new ammo type.
C  RNDG: rounds/tube; RNDT: tot rnds (RNDG*wpns short).
C  DRAW and CHM = intermediate steps for RNDT
C  NOCI holds the CI number (1-55) of each CI involved.
C  DRN holds the phase categories (day, night).
C
C  DIMENSION IAMAT(30,3),NOCI(3),RNDG(6,3),IND(12),KMB(6,3),MNT(6,3)
C  DIMENSION IFIL(3), AMON(6), INV(15), IWPV(6,6), RNDT(3), DRN(2), WPN(6)
C  DIMENSION KWPSYS(14), WPSYS(14), MAXAR(13,7), MINAR(13,7), UNAME(75)
C  DIMENSION KHW(6), MAXV(12), CHNGS(12), PER(8), LOAD(14), INHWL(13)
C
C  CHARACTER*8 UNAME
C  CHARACTER*10 WPSYS,CHNGS,AMON,WPN
C  CHARACTER*5 DRN
C
C  MAXV holds upper limits for testable variables:
C  MAXV:  CI no.,   MOPP*100,   intensity, phase,
C         ret opts, operation, method,   unit no.,
C         wpn sys,  status,    file size, no of CIs
C
C  DATA MAXV/55,100,8,2,4,5,3,5,14,12,30,3/
C
C  MAXAR and MINAR hold ammo expenditure limits for the 13
C  standard wpn types included in the data base.

```

```

DATA ((MAXAR(I,J),J=1,7),I=1,13)
Z/ 93, 60, 35, 58, 30, 87, 0,
A 400, 240, 350, 0, 0, 0, 0,
B 1200, 720, 1100, 0, 0, 0, 0,
C 20, 12, 8, 15, 5, 20, 0,
D 12, 8, 10, 8, 4, 12, 0,
E 420, 295, 135, 232, 84, 350, 0,
F 750, 480, 200, 480, 200, 300, 0,
G 380, 261, 186, 0, 0, 380, 0,
H 550, 375, 250, 0, 0, 550, 0,
I 210, 145, 100, 0, 0, 210, 0,
J 720, 458, 300, 0, 0, 630, 0,
K 16, 16, 16, 16, 16, 16, 16,
L 500, 500, 500, 500, 500, 500, 500/

```

C

```

DATA ((MINAR(I,J),J=1,7),I=1,13)
Z/ 35, 24, 10, 15, 5, 30, 0,
A 200, 120, 180, 0, 0, 0, 0,
B 700, 420, 650, 0, 0, 0, 0,
C 8, 5, 4, 3, 1, 8, 0,
D 5, 2, 2, 2, 1, 5, 0,
E 240, 135, 80, 92, 24, 190, 0,
F 550, 300, 60, 300, 60, 100, 0,
G 300, 181, 86, 0, 0, 300, 0,
H 450, 275, 150, 0, 0, 450, 0,
I 75, 85, 50, 0, 0, 150, 0,
J 540, 298, 180, 0, 0, 450, 0,
K 12, 12, 12, 12, 12, 12, 12,
L 400, 400, 400, 400, 400, 400, 400/

```

C

C

C

Unit names

```

DATA UNAME/'TF1A', 'TF2A', 'TF3A', 'TF4A', 'TF5A',
a/'TF6A', 'TF7A', 'TF8A', 'TF9A', 'TF10A', 'TF11A',
b/'TF12A', 'TF13A', 'TF14A', 'TF15A', 'TF16A', 'TF17A',
c/'FA13ACR', 'A11FA', 'B11FA', 'C11FA', 'A12FA', 'B12FA',
d/'C12FA', 'A13FA', 'B13FA', 'C13FA', 'A14FA', 'B14FA',
e/'C14FA', 'A15FA', 'B15FA', 'C15FA', 'A16FA', 'B16FA',
f/'C16FA', 'A17FA', 'B17FA', 'C17FA', 'A18FA', 'B18FA',
g/'C19FA', 'A19FA', 'B19FA', 'C19FA', 'A20FA', 'B20FA',
h/'C20FA', 'A21FA', 'B21FA', 'C21FA', 'A22FA', 'B22FA',
i/'C22FA', 'A23FA', 'B23FA', 'C23FA', 'A24FA', 'B24FA',
j/'C24FA', 'A25FA', 'B25FA', 'C25FA', 'A26FA', 'B26FA',
k/'C26FA', 'A1ADA', 'B1ADA', 'C1ADA', 'A12ADA', 'B12ADA',
l/'1ACAS', '2ACAS', '3ACAS',

```

C

C

C

Weapon system names

```

DATA WPSYS/'105MM TANK', '50CAL TANK', '7.62 MG', 'CFV TOW',
Z/'IFV TOW', 'MORTAR', 'BUSHMASTER', '8-IN HOW', '155 HOW',
A/'MLRS', 'DIVAD', 'HELLFIRE', '30MM ATK H', 'OTHER'

```

C

C

C

WPSYS holds ammo codes for each wpn type; KHW holds
ammo codes for howitzers. They = IAMAT, col 1.

```

DATA KWFSYS/ 1,23,24, 2,25,16,17, 6, 4,10,13, 9,14,99/
C
C      Names of available options. Values held in IND array.
C
DATA CHNGS/ 'WPNS ALIVE','WPNS SHORT','WPN TYPE ', 'ATK HELIS ',
Z          'MAINT RET ', 'COMBAT RSU', 'INS DEMAND', 'COPY UNIT',
Z          'AUTO LIST ', 'AUTO SAVE ', 'NEW DEMAND', 'R/ AMMO NO'/
C
DATA (INV(I),I=1,15)/0,100,1,1,11*0/, MNT,KMB/36*0/
C      IFIL holds base no for input and output units (files)
C      IFIL:  binary read,  binary write,  report file
DATA WPN,AMON/12*' ' //, IFIL/9,11,14/
DATA IND /12*0/, IWPV/36*0/, RND5/18*0.0/, RNDT/3*0.0/
C      Basic load factors for wpn systems, used for calculating
C      demand increases due to mnt returns and combat reserves.
DATA LOAD/1,0,0,12,6,115,350,0,0,12,1078,16,1200,0/
DATA IHOWL/21,83,14,5,5,8,30,6,2,2,10,34,6/
C
C      PER holds percentages for howitzer ammo types.
DATA PER/0.16,0.65,0.11,0.04,0.04,0.20,0.68,0.12/
DATA IAMAT/90*0/, NOCI/3*' ' //, DRN/' DAY ', 'NIGHT'/
DATA IANS,LEAP/2*0/, KROW,KCOL/2*1/, RETS,DRAW,CUM/3*0.0/
C
C      *****
C      ***** MAIN PROGRAM *****
C      *****
C      Determine how the files will be handled.
1000  LEAP=9
      PRINT*, ' ENTER FILE TREATMENT. (INFO=0)'
      WRITE(6,1001)
1001  FORMAT(/,1X,'1. NEW FILES FROM OLD.  2. NEW FILES,'
Z      'FROM SCRATCH.  3. COPY SECTIONS.')
      READ*, INV(13)
      IF(INV(13).EQ.0)CALL INFO(LEAP,*1000)
      CALL VERIFY(INV(13),3,*1000)
C      *****
C      Enter the no. of CIs, each CI no. and verify. CI nos
C      are for listings only. There is no affect on files.
      LEAP=6
30  PRINT*, ' ENTER THE NO. OF CIs AND EACH CI NO. (INFO=0,0)'
      READ*, INV(15),(NOCI(I),I=1,INV(15))
      IF(INV(15).EQ.0)CALL INFO(LEAP,*30)
      CALL VERIFY(INV(15),MAXV(12),*30)
      IANS=0
      DO 39 I=1,INV(15)
          IF(NOCI(I).LT.1.OR.NOCI(I).GT.MAXV(1))IANS=-1
39      CONTINUE
      CALL VERIFY(IANS,0,*30)
C      *****
C      open the files
DO 32 I = 1,INV(15)
      OPEN(UNIT=(IFIL(1)+I),ACCESS='DIR',RCDS=MAXV(8),RECL=MAXV(11),
Z          ASSOC=NRED)
      OPEN(UNIT=(IFIL(2)+I),ACCESS='DIR',RCDS=MAXV(8),RECL=MAXV(11),

```

```

32  CONTINUE
C      *****
      IF(INV(13).EQ.2)THEN
          CALL NUFIL
          CALL UNTNOS
      ELSE
          IF(INV(13).EQ.1)CALL COPFIL  @Copy read files
          CALL METHUD  @Establish method
          CALL UNTNOS  @obtain unit nos.
      ENDIF
C      *****
60  CALL BRANCH  @branch to chosen method
90  LEAP=NXTSTP(dummy)  @determine next step
C
      GOTO(92,93,94,95,96,97)LEAP
93  CALL METHUD  @change methods
92  CALL UNTNOS  @new unit nos, same method
      GOTO 60
C
94  CALL COPYUN(*90)  @copy sequence of units
C
95  CALL NUCI(*1000)  @new run
C
96  CALL PRNTFL(*90)  @print file (DPFD or screen)
C
97  DO 98 I=1,INV(15)
      CLOSE(UNIT=(IFIL(1)+I))
      CLOSE(UNIT=(IFIL(2)+I))
98  CONTINUE
      PRINT*, ' PROGRAM IS FINISHED.'  @tpe
      STOP
C
C      *****
C      ***** SUBROUTINE SECTION *****
C      *****
SUBROUTINE METHUD
C      Set counter (LEAP) for use with INFO routine.
C      Obtain type of operation and method to be used.
C      Test responses for validity.
100  LEAP=1
      PRINT*, ' ENTER OPERATION AND METHOD. (INFO=0,0)'
      PRINT*, ' TYPE OF OPERATION:          METHOD TO BE USED:'
      PRINT*, ' 1 = EDIT/NEW DEMAND          1 = UNITS IN SEQUENCE'
      PRINT*, ' 2 = COPY SEQUENCES           2 = UNITS AT RANDOM'
      PRINT*, ' 3 = PRINT FILES              3 = UNITS BY GROUP'
      PRINT*, ' 4 = CHANGE BY ROW'
C      PRINT*, ' 5 = CREATE NEW FILES'  @Assigned in NUFIL
      READ*, INV(6), INV(7)
      IF(INV(6).EQ.0.OR.INV(7).EQ.0)CALL INFO(LEAP,*100)
      CALL VERIFY(INV(6),MAXV(6),*100)
      CALL VERIFY(INV(7),MAXV(7),*100)
      RETURN
C      END          METHOD
C      *****

```

```

C      Set LEAP for use with info routine.
C      User enters the range of unit numbers (1st and 1st
C      nos. when METHOD = sequence or group, one no. when
C      METHOD = random.
35  LEAP=3
    IF(INV(7).NE.2)THEN
      PRINT*, ' ENTER FIRST AND LAST UNIT NOS. (INFO=0,0)'
      READ*, INV(8),INV(9)
      CALL VERIFY(INV(8),INV(9),*35)
    ELSE
      PRINT*, ' ENTER UNIT NUMBER. (INFO=0)'
      READ*,INV(8)
      INV(9)=INV(8)          ?Initialize last unit no.
    ENDIF
    IF(INV(9).EQ.0)CALL INFO(LEAP,*35)
    INV(10)=INV(8)          ?Initialize current unit no.
    CALL VERIFY(INV(9),MAXV(8),*35)
    RETURN
C  END          UNTNOS
C  *****
SUBROUTINE NUFIL
C      When FILE TREATMNT = 2, initialize new file.
80  CALL ZOT
    DO 93 L=1,INV(15)
      DO 84 K=1,MAXV(8)
        WRITE(UNIT=(IFIL(2)+L),REC=K,ERR=89) (IAMAT(J,L),J=1,MAXV(11))
84  CONTINUE
83  CONTINUE
    INV(6)=5
    LEAP=9
95  PRINT*, ' ENTER METHOD TO BE USED. (INFO=)'
    PRINT*, ' 1. UNITS IN SEQUENCE  2. RANDOM UNITS  3. GROUPS'
    READ*, INV(7)
    IF(INV(7).EQ.0)CALL INFO(LEAP,*85)
    CALL VERIFY(INV(7),MAXV(7),*85)
    RETURN
89  CALL FILERR(*80)
C  END          NUFIL
C  *****
SUBROUTINE COFFIL
C      When FILE TRTMNT = 1, copy read file into write file.
    DO 209 J=1,INV(15)
207  DO 208 I=1,MAXV(8)
      READ(UNIT=(IFIL(1)+J),REC=I,ERR=29) (IAMAT(L,J),L=1,MAXV(11))
      WRITE(UNIT=(IFIL(2)+J),REC=I,ERR=29) (IAMAT(L,J),L=1,MAXV(11))
208  CONTINUE
209  CONTINUE
    RETURN
28  CALL FILERR(*207)
C  END          COFFIL
C  *****
SUBROUTINE LNLUN
C      load & list a specified unit from the files.
    CALL FETCH

```

```

      RETURN
C      END      LNLUN
C      *****
SUBROUTINE FETCH
C      obtain unit specified in INV(10) from each file
C      and load it into IAMAT array. IFI designates which
C      files will be read from.
      IFI=IFIL(2)
      IF(INV(13).EQ.3)THEN      @if FILE TRTMNT=3
        PRINT*, ' IS UNIT TO BE READ FROM SAME FILE? (1=YES)'
        READ*, IANS
        IF(IANS.NE.1)IFI=IFIL(1)
      ENDIF
40     DO 41 J=1,INV(15)
      READ(UNIT=(IFI+J),REC=INV(10),ERR=43)(IAMAT(L,J),L=1,MAXV(11))
41     CONTINUE
      RETURN
43     CALL FILERR(*40)
C      END      FETCH
C      *****
SUBROUTINE WINDOW
C      print the attribute array for each CI.
33     IF(INV(15).EQ.3)THEN
      WRITE(6,11) UNAME(INV(10)),(((IAMAT(I,J),I=K,K+4),
Z      J=1,INV(15)),K=1,MAXV(11),5)
      ELSE
      IF(INV(15).EQ.2)THEN
      WRITE(6,12) UNAME(INV(10)),(((IAMAT(I,J),I=K,K+4),
Z      J=1,INV(15)),K=1,MAXV(11),5)
      ELSE
      WRITE(6,13) UNAME(INV(10)),(IAMAT(I,1),I=1,MAXV(11))
      ENDIF
      ENDIF
11     FORMAT(/,1X,A8,4I4,I5,4I4,I5,4I4,I5/, (9X,4I4,I5,4I4,I5,4I4,I5))
12     FORMAT(/,1X,A8,3X,S15,3X,S15/, (12X,S15,3X,S15))
13     FORMAT(/,1X,A8,3X,S15/, (12X,S15,3X))
      RETURN
C      END      WINDOW
C      *****
SUBROUTINE BRANCH
C      Branch to operation as specified by user.
      GOTO(61,62,63,64,65)INV(6)
C
61     CALL EDDEM      @enter changes
      RETURN
62     CALL COPYUN(*69)      @copy sequence
      RETURN
63     CALL PRNTFL(*69)      @print from files
      RETURN
64     CALL ROWCH      @change by individual row
      RETURN
65     CALL NEWFLS      @enter changes into new files
69     RETURN
C      END      BRANCH

```

```

SUBROUTINE PRNTFL(*)
C      Print data from files regarding units listed.
      PRINT*, ' PRINT LOCALLY OR AT DPFQ? (1=LOCAL)'
      READ*, IANS
      IF(IANS.EQ.1)THEN
        DO 74 IJ=INV(8),INV(9) @unit nos. specified
          INV(10)=IJ
          CALL LNLUN
74      CONTINUE
      RETURN 1
      ELSE
        DO 76 I=1,INV(15)
          DO 78 J=1,MAXV(8)
            READ((IFIL(2)+I),REC=J) (IAMAT(K,I),K=1,MAXV(11))
            WRITE((IFIL(3)+I),1113) J,UNAME(J),(IAMAT(K,I),K=1,MAXV(11))
1113      FORMAT(/,2X,I2,'.',2X,A8,3X,5I5,/, (18X,5I5))
78      CONTINUE
76      CONTINUE
      ENDIF
      RETURN 1
C      END      PRNTFL
C      *****
C      SUBROUTINE EDDM
C      Enter changes. All options in CHNGS available.
      CALL ININD(3,9)
      CALL STATIS(1,MAXV(10))
623     CALL INIT
      CALL FETCH      @change to LNLUN for file check
625     CALL CHANGD
      CALL NXTWPN(*625)
      CALL WINDOW      @chk contents of array
      CALL PLIST
      CALL SAVAR(INV(10))
      CALL NXTUNT(*623)
      RETURN
C      END      EDDM
C      *****
C      SUBROUTINE COPYUN(*)
C      Copy one sequence of units into another. Options
C      incl new demand, save and list. Current units (in
C      INV(8 & 9)) are copied into the unit no. in INV(12)
C      (INV(11) holds 1st new unit). When complete, the
C      copied units are set to current status.
      CALL ININD(1,10)
      IND(8)=1
      CALL STATIS(9,11)
      CALL INIT
      INV(10)=INV(8)
888     PRINT*, ' ENTER FIRST UNIT NO. TO BE COPIED INTO. '
          READ*, INV(11)
          CALL VERIFY(INV(11),MAXV(8),*899)
          INV(12)=INV(11)
          DO 899 ICUR=INV(8),INV(9)
            INV(10)=ICUR

```

```

        CALL DUPE(INV(12))
        INV(12)=INV(12)+1
899      CONTINUE
        INV(8)=INV(11)
        INV(9)=INV(12)-1
        INV(10)=INV(9)
        RETURN 1
C      END          COPYUN
C      *****
SUBROUTINE NEWFLS
C      Enter changes into newly created files. All options
C      available except 'locate by ammo code.'
        CALL ININD(5,9)
        CALL ST 'IS(1,11)
802      CALL INIT
        CALL FETCH
801      CALL CHNGD
        CALL NXTWPN(*801)
        CALL WINDOW          @chk contents of array
        CALL PLIST
        CALL SAVAR(INV(10))
        CALL NXTUNT(*802)
        RETURN
C      END          NEWFLS
C      *****
SUBROUTINE ININD(J,K)
C      Initialize the status indicator. Default value is 1.
C      Status 0 determined by calling routine (J,K=limits).
        DO 610 I=1,(MAXV(10)-1)
            IND(I)=1
610      CONTINUE
        DO 620 I=J,K
            IND(I)=0
620      CONTINUE
        IND(MAXV(10))=0
        RETURN
C      END          ININD
C      *****
SUBROUTINE STATIS(LL,KK)
C      Display options valid for the calling routine and
C      their value (1 = do change, 0 = don't). ICH = 1 if
C      status is altered. LL-KK = options to be displayed.
        LEAP=5
        ICH=0
70      PRINT*, ' STATUS OF OPTIONS: (1=CHANGE, 0=STAY SAME)'
        WRITE(6,1117) (I, CHNGS(I), IND(I), I=LL, KK)
1117      FORMAT(/,3(2X,I3,' ',A10,' ',I1))
C
        PRINT*, ' TO ALTER STATUS, ENTER ITEM NO.(0=NONE;INFO=99)'
71      READ*, IANS
        IF(IANS.EQ.0)THEN
            IF(ICH.GT.3)WRITE(6,1117)(I,CHNGS(I),IND(I), I=LL, KK)
            RETURN
        ENDIF

```

```

      CALL VERIFY(IANS,KK,*71)
      IF(IND(IANS).EQ.0)THEN
        IND(IANS)=1
      ELSE
        IND(IANS)=0
      ENDIF
      ICH=ICH+1                                @set alter indicator
      GOTO 71
C      END          STATIS
C      *****
SUBROUTINE INIT
C      Initialize variables.
      DO 110 I=1,INV(15)
        DO 120 J=1,6
          MNT(J,I)=0
          KMB(J,I)=0
          RNDS(J,I)=0.0
120      CONTINUE
          RNDT(I)=0.0
          DO 122 J=1,MAXV(11)
            IAMAT(J,I)=0
122      CONTINUE
110      CONTINUE
          DO 124 I=1,6
            DO 123 J=1,6
              IWPV(I,J)=0
123      CONTINUE
              WPN(I)=' '
              AMON(I)=' '
124      CONTINUE
          RETS=0.0
          KCOL=1
          KROW=1
          INV(5)=0
          RETURN
C      END          INIT
C      *****
SUBROUTINE CHNGD
C      User enters options as indicated by CHNGS status.
C      *****
C      Locate specific ammo code. Need not be in order.
75      LEAP=10
          IF(IND(12).NE.1)GOTO 73
          PRINT*, ' ENTER AMMO CODE: (INFO=99)'
          READ*, INV(14)
          IF(INV(14).EQ.99)CALL INFO(LEAP,*75)
          KROW=1
74      KCOL=5*KROW-4
          IF(IAMAT(KCOL,1).EQ.INV(14))GOTO 73
          KROW=KROW+1
          IF(KROW.LE.6)GOTO 74
          PRINT*, ' AMMO CODE NOT FOUND. TRY AGAIN.'
          GOTO 75
C      *****

```

```

1175      FORMAT(/,10X,'FOR WPN NO. ',I2,' OF UNIT ',I3,
           ' ENTER THE FOLLOWING:')
C          Determine the weapon system number.
LAP=1      @offset for NUWPN routine
IF(IND(3).EQ.1)CALL NUWPN(LAP,*37,*750)
CALL CURWPN(*750)
C          *****
C          Determine the total no of weapons alive. For
c          information, user enters 99 as 1st entry and
c          any interger(s) for the other(s). To zero out
c          the rest of the array for the entire phase,
c          user enters 9999 as first input. . .etc.
37      IF(IND(1).EQ.1)THEN
          LEAP=11
          PRINT*, ' NO. OF WPNS ALIVE PER CI: (NONE=0;INFO=99 . .)'
          READ*, (IAMAT(KCOL+1,I),I=1,INV(15))
          IF(IAMAT(KCOL+1,1).EQ.99)CALL INFO(LEAP,*37)
          IF(IAMAT(KCOL+1,1).EQ.9999)THEN
              KROW=6          @set row indx to last row
              DO 371 I=1,INV(15)
                  DO 372 J=KCOL,30,5
                      DO 378 K=1,4
                          IAMAT(J+K,I)=0
378                      CONTINUE
372                  CONTINUE
371              CONTINUE
              RETURN
          ENDIF
      ENDIF
C          *****
C          Determine total no of wpns firing (short of ammo).
IF(IND(2).EQ.1)THEN
C          (Artillery receive ammo differently.)
IF(IWPV(KROW,1).GT.7.AND.IWPV(KROW,1).LT.11)THEN
    PRINT*, ' NO. OF WPNS RECEIVING AMMO PER CI:'
ELSE
    PRINT*, ' NO. OF WPNS SHORT PER CI: (0=NONE)'
ENDIF
    READ*, (IAMAT(KCOL+2,I),I=1,INV(15))
ENDIF
C          Verify wpns alive greater than/equal wpns short.
C          For artillery, test does not apply.
IF(IWPV(KROW,1).GT.10.OR.IWPV(KROW,1).LT.8)THEN
    IANS=0
    DO 370 I=1,INV(15)
        IF(IAMAT(KCOL+2,I).GT.IAMAT(KCOL+1,I))IANS=-1
370    CONTINUE
    CALL VERIFY(IANS,0,*37)
ENDIF
C          *****
C          Enter the no. of maintenance returns.
IF(IND(5).EQ.1)THEN
    PRINT*, ' NO. OF MAINTENANCE RETURNS PER CI: (0=NONE)'
    READ*, (MNT(KROW,I), I=1,INV(15))

```

```

                INV(5)=4          @set indicator for returns
ENDIF
C          *****
C          Enter the no. of combat reserves.
IF(IND(6).EQ.1)THEN
    PRINT*, ' NO. OF COMBAT RESERVES PER CI: (0=NONE)'
    READ*, (KMB(KROW,I),I=1,INV(15))
    IF(IND(11).NE.1.AND.IND(7).NE.1)CALL OLDDEM
    INV(5)=4          @set indicator for returns
ENDIF
C          *****
C          Enter the no. of attack helicopters per cell.
IF(IND(4).EQ.1.AND.(IWPV(KROW,1).EQ.12.OR.IWPV(KROW,1).EQ.13))THEN
    PRINT*, ' NO. OF ATK HFLIS/CELL PER CI: (0=NONE)'
    READ*, (IAMAT(KCOL+3,I),I=1,INV(15))
ENDIF
C          *****
C          Enter a preselected demand.
IF(IND(7).EQ.1)THEN
    PRINT*, ' PRESELECTED DEMAND PER CI: (INTEGER ONLY)'
    READ*, (IAMAT(KCOL+4,I),I=1,INV(15))
ENDIF
C          Data for howitzers is duplicated for each ammo type.
IF(IWPV(1,1).EQ.8.OR.IWPV(1,1).EQ.9)THEN
    MX=4          @# of add. ammo types for 155 how
    IF(IWPV(1,1).EQ.8)MX=2 @for 8"
    DO 374 J=1,INV(15)
        DO 376 I=5,5*MX,5
            IAMAT(I+2,J)=IAMAT(2,J)
            IAMAT(I+3,J)=IAMAT(3,J)
            IF(IND(7).EQ.1)IAMAT(I+5,J)=IAMAT(5,J)
376        CONTINUE
        DO 377 K=2,MX+1
            MNT(K,J)=MNT(1,J)
            KMB(K,J)=KMB(1,J)
377        CONTINUE
374        CONTINUE
    ENDIF
C          *****
C          generate new demand.
IF(IND(11).EQ.1)CALL DEMAND
750 RETURN
C          END          CHANGD
C          *****
C          SUBROUTINE DEMAND
C          Demand factors (MOPP, Battle Intens, Phase, New Rets)
C          are polled for each unit. Ammo and wpn types entered
C          if necessary. Expenditure rates difference calculated
C          and verified. New demand generated.
IF(KROW.EQ.1.OR.IND(12).EQ.1)THEN
    CALL DEFACT          @if 1st wpn, check DMD FACTORS
ELSE
    @check if new wpn rets needed
    IF(INV(5).NE.0.AND.INV(5).NE.4)CALL CHKRET

```

```

      CALL CURWPN(*375)           @obtain current wpn type
      IF(IWPV(KROW,1).EQ.14)THEN  @new wpn type
      PRINT*, ' ENTER NAME OF NEW WPN SYSTEM.'
      READ*, WPN(KROW)
    ELSE
      WPN(KROW)=WPSYS(IWPV(KROW,1))
    ENDIF
373  IF(INV(3).EQ.8)THEN          @new ammo type
      PRINT*, 'ENTER NAME OF AMMO TYPE'
      READ*, AMON(KROW)
      PRINT*, ' ENTER HIGH AND LOW EXPENDITURE RATES.'
      READ*, IWPV(KROW,2),IWPV(KROW,3)
    ELSE
      @ammo expenditure rates
      IWPV(KROW,2)=MAXAR(IWPV(KROW,1),INV(3))
      IWPV(KROW,3)=MINAR(IWPV(KROW,1),INV(3))
    ENDIF

C      CALL VERIFY(IWPV(KROW,3),IWPV(KROW,2),*373)
      IWPV(KROW,6)=IWPV(KROW,2)-IWPV(KROW,3)

C      *****
C      demand calculations
C      A random no. is generated and multiplied by the
C      expenditure rate diff., the lower limit is added,
C      and the result rounded and saved as 'day's draw.'
C      The phase period draw, equals a proportion of the
C      total (.6=day;.4=night) * MNPP. The CI draw equals
C      a proportion of phase total. For helicopters, phase
C      total = day total. A new draw is made for each CI.
      IF(IWPV(KROW,1).EQ.12.OR.IWPV(KROW,1).EQ.13)THEN
      IWPV(KROW,4)=0
      DO 48 I=1,INV(15)
      47  DRAW=РАНF(DUM)
      DRAW=DRAW*IWPV(KROW,6)+IWPV(KROW,3)
      IF(NINT(DRAW).LT.IWPV(KROW,3))GOTO 47
      RNDS(KROW,I)=NINT(DRAW)
      IWPV(KROW,4)=IWPV(KROW,4)+RNDS(KROW,I)
      RNDT(I)=RNDS(KROW,I)*IAMAT(KCCL+2,I)  @tot rnds
      48  CONTINUE
      IWPV(KROW,5)=IWPV(KROW,4)  @phase draw=tot draw
    ELSE
      44  DRAW=РАНF(DUM)
      DRAW=DRAW*IWPV(KROW,6)+IWPV(KROW,3)
      IF(NINT(DRAW).LT.IWPV(KROW,3))GOTO 44
      IWPV(KROW,4)=NINT(DRAW)
      IF(INV(4).EQ.1)THEN
      DRAW=DRAW*.6*INV(2)/100.  @day draw
      ELSE
      DRAW=DRAW*.4*INV(2)/100.  @night draw
      ENDIF
      IWPV(KROW,5)=NINT(DRAW)  @phase draw saved
C      *****
C      Determine cumulative rounds per wpn per CI.
      46  CUM=0.0
      DO 42 I=1,INV(15)

```

```

      CUM=CUM+RNDS(KROW,I)
42  CONTINUE
      JST=0
      DO 43 J=1,INV(15)
        RNDS(KROW,J)=NINT(RNDS(KROW,J)/CUM*DRAW)
        IF(RNDS(KROW,J).LT.1.0)RNDS(KROW,J)=1.0
c    Test for divad (a rnd = 90 bursts).
        IF(IWPV(KROW,1).EQ.11)THEN
          RNDS(KROW,J)=NINT(RNDS(KROW,J)/90.0)
          IF(RNDS(KROW,J).LT.1.0)RNDS(KROW,J)=1.0
          RNDS(KROW,J)=RNDS(KROW,J)*90
        ENDIF
C    Tests for howitzers.
        IF(IWPV(KROW,1).EQ.9.AND.RNDS(KROW,J).GT.(300./INV(15)))JST=1
        IF(IWPV(KROW,1).EQ.8.AND.RNDS(KROW,J).GT.(300./INV(15)))JST=1
C    Total rounds = rnds*wns short.
        RNDT(J)=RNDS(KROW,J)*IAMAT(KCOL+2,J) @tot rnds
43  CONTINUE
        IF(JST.NE.0)GOTO 46
      ENDIF

C          *****
C    Incorporate mnt returns & cmb reserves into tot rnds.
C    Calculate howitzer rounds for each ammo type.
C
      DO 45 I=1,INV(15)
        IF(IWPV(1,1).EQ.9)THEN @155 howitzers
          DO 451 J=1,5
            RETS=KMB(J,I)+MNT(J,I)/2.
            IF(INV(10).LT.42)THEN
              IAMAT(J*5,I)=NINT(PER(J)*RNDT(I)+(IHOWL(J)*RETS))
            ELSE
              IAMAT(J*5,I)=NINT(PER(J)*RNDT(I)+(IHOWL(J+5)*RETS))
            ENDIF
            RNDS(J,I)=RNDS(1,I) @dupe rnds/wpn
451    CONTINUE
          ELSE
            IF(IWPV(1,1).EQ.8)THEN @9-inch howitzers
              DO 452 J=1,3
                RETS=MNT(J,I)/2.+KMB(J,I)
                IAMAT(J*5,I)=NINT(PER(J+5)*RNDT(I)+IHOWL(J+10)*RETS)
                RNDS(J,I)=RNDS(1,I) @dupe rnds/wpn
452    CONTINUE
              ELSE
                RETS=KMB(KROW,I)+MNT(KROW,I)/2.
                IAMAT(KCOL+4,I)=NINT(RNDT(I)+LOAD(IWPV(KROW,1))*RETS)
              ENDIF
            ENDIF
          CONTINUE
        RETURN
375  RETURN
C  END DEMAND
C  *****
C  SUBROUTINE NXTWPN *)
C    Determine if another wpn in unit is to be changed.
C    . . All artillery have only one 'wpn' entered.

```

```

        IF(KROW.GE.6)RETURN                                @last row in array
        KROW=KROW+1
        KCOL=5*KROW-4
C      Automatically go to next wfn unless inputting changes.
        IF((IND(8).EQ.2).OR.(INV(6).NE.1.AND.INV(6).NE.5))RETURN 1
        IF(INV(7).EQ.2)THEN
            PRINT*, ' ANOTHER WFN SYSTEM FOR UNIT? (0=NO)'
        ELSE
C      999 used only to escape sequential entry method.
        PRINT*, 'ANOTHER WFN FOR THIS UNIT? (0=NO;999=EXIT SEQ)'
        ENDIF
            READ*, IANS                                ?poll if entering changes.
            IF(IANS.EQ.999)THEN
                IANS=0
                INV(9)=INV(10)
            ENDIF
            IF(IANS.EQ.0)RETURN
        RETURN 1
C      END      NXTWPN
C      *****
C      SUBROUTINE PLIST
C      Print a listings of the inputs and factors pertaining
C      to the current unit onto the screen/local printer.
        IF(IND(9).EQ.1)GOTO 50
        PRINT*, ' LIST THE UNIT STATS? (1=YES)'
        READ*, IANS
        IF(IANS.NE.1)RETURN
50      WRITE(6,1500)
        IF(INV(4).EQ.8)THEN
            WRITE(6,1503) INV(10),UNAME(INV(10)),DRN(INV(4)),(WPN(I),
Z          (IWPV(I,J),J=1,5),AMON(I),I=1,6)
        ELSE
            IF(IWPV(1,1).EQ.9.OR.IWPV(1,1).EQ.8)THEN
                WRITE(6,1506) INV(10),UNAME(INV(10)),DRN(INV(4)),WPN(1),
Z          (IWPV(1,J),J=1,5)
                WRITE(6,1508)
                WRITE(6,1507) ((IAMAT(I,J),I=5,25,5),J=1,INV(15))
            ELSE
                WRITE(6,1505) INV(10),UNAME(INV(10)),DRN(INV(4)),(WPN(I),
Z          (IWPV(1,J),J=1,5),I=1,6)
            ENDIF
        ENDIF
        WRITE(6,1501)
        DO 505 L=1,INV(15)
            WRITE(6,1504) NOCI(L),((IAMAT(5*I-J,L),J=4,1,-1),MNT(I,L),
Z          KMB(I,L),RND5(I,L),IAMAT(5*I,L),I=1,5)
505      CONTINUE
        RETURN
1500  FORMAT(/,5X,'UNIT',5X,'PHASE WEAPON SYSTEM AMMO EXPENDITURE-',
Z      'DEMAND (NEW AMMO)',/,21X,'NAME',7X,'NO (LIMITS)',
Z      'DRAW /PHASE')
1501  FORMAT(/,9X,'CI AMMO',4X,'WEAPONS',4X,'AHS/ RETURNS ROUNDS ',
Z      'SPENT',/,9X,'NO CODE ALIVE SHORT CELL MNT ONR',
Z      '/TUBE TOTAL')

```

```

1503 FORMAT(/,1X,I2,2X,A8,1X,A5,2X,A10,I3,I6,I5,2I7,2X,A10,/, (21X,
z      A10,I3,I6,I5,2I7,2X,A10))
1505 FORMAT(/,1X,I2,2X,A8,1X,A5,2X,A10,I3,I6,I5,2I7,/, (21X,
z      A10,I3,I6,I5,2I7))
C
1504 FORMAT(/,9X,I2,I6,I7,2I6,I5,I4,1X,F6.1,I7,/, (11X,I6,I7,2I6,I5,
z      I4,1X,F6.1,I7))
1506 FORMAT(/,1X,I2,2X,A8,1X,A5,2X,A10,I3,I6,I5,2I7)
1507 FORMAT(/,30X,4I6,I5)
1508 FORMAT(/,31X,'HOWITZER DEMAND BY AMMO TYPE',/,33X,
z      'HE ICM RAP CLGP ETC.')
```

C END PLIST

C *****

 SUBROUTINE SAVAR(N)

C Save unit attributes (IAMAT array) in record ind.

C by N (= unit no). If auto save not specified in

C STATUS, user is polled.

 IF(IND(10).EQ.1)GOTO 501

 PRINT*, ' SAVE? (1=YES)'

 READ*, IANS

 IF(IANS.NE.1)RETURN

C Howitzers: wdns alive copied into wdns short.

501 IF(IWPU(1,1).EQ.8.OR.IWPU(1,1).EQ.9)THEN

 DO 502 I=1,INV(15)

 DO 503 J=2,27,5

 IAMAT(J+1,I)=IAMAT(J,I)

503 CONTINUE

502 CONTINUE

 ELSE Qif divid, set wdns sht=wdns alive

 DO 504 I=1,INV(15)

 DO 506 J=1,8

 IF(IWPU(1,1).EQ.10)IAMAT(5*I-2,I)=IAMAT(5*I-3,I)

506 CONTINUE

504 CONTINUE

 ENDIF

777 DO 778 I=1,INV(15)

 WRITE(UNIT=(IFIL(2)+I),REC=N,ERR=779) (IAMAT(J,I),J=1,MAXU(11))

778 CONTINUE

 RETURN

779 CALL FILERR(*777)

C END SAVAR

C *****

 SUBROUTINE NXTUNT(*)

C See if the unit just modified is to be copied,

C and if it is < the max in sequence. Increment

C current unit no and copy sequentially if INV(7)=3.

 IF(IND(8).EQ.1.AND.INV(7).NE.3)CALL DITTO

 IF(INV(10).GE.INV(9).AND.INV(7).NE.3)RETURN

 INV(10)=INV(10)+1

 IF(INV(7).NE.3)RETURN 1

 DO 939 JK=INV(10),INV(9)

 INV(10)=JK

 CALL DUPE(INV(10))

939 CONTINUE

AD-A131 218

AMMUNITION RESUPPLY MODEL VOLUME 2 PROGRAMMER'S MANUAL

3/3

(U) COMBINED ARMS OPERATIONS RESEARCH ACTIVITY FORT

LEAVENWORTH KS A M RESNICK ET AL. MAY 83

UNCLASSIFIED

CAORA/TR-6/83-VOL-2

F/G 19/1

NL

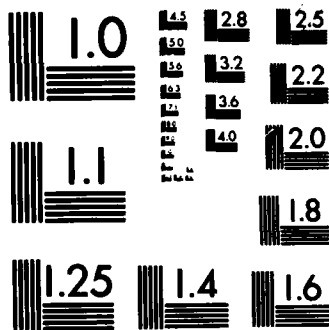


END

FORMED

101

1000



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

C      END          NXTUNT
C      *****
C      FUNCTION NXTSTP(dum)
C      Return value of the next operation desired by user.
900    LEAP=7
        dum=0
        PRINT*, ' WHAT NEXT? (INFO=0)'
        PRINT*, ' 1=CONTINUE METHOD          2=CHANGE METHOD'
        PRINT*, ' 3=COPY UNITS              4=NEW RUN'
        PRINT*, ' 5=PRINT MODIFIED FILE    6=END'
        READ*, IANS
        IF(IANS.EQ.0)CALL INFO(LEAP,*900)
        CALL VERIFY(IANS,6,*900)
        NXTSTP=IANS
C      RETURN
C      END FUNCTION
C      *****
C      SUBROUTINE NUCI(*)
C      Start a new run.
        PRINT*, ' CAUTION. ROUTINE ONLY WORKS IF ADDITIONAL'
        PRINT*, ' UNITS WERE ASSIGNED IN THE SFG RUNSTREAM.'
        READ*, IANS
        IF(IANS.NE.1)RETURN 1
        DO 400 I=1,INV(15)
            CLOSE ((IFIL(1)+I))
            CLOSE ((IFIL(2)+I))
400    CONTINUE
        CALL INIT
        IFIL(1)=IFIL(1)+9
        IFIL(2)=IFIL(2)+9
        IFIL(3)=IFIL(3)+9
        RETURN 1
C
C      END          NUCI
C      *****
C      SUBROUTINE ROWCH
C      Enter changes to each column of row specified in CI
C      specified (1,2,3). Rows may be given in any order.
C      IROW: changes; JR: CI and row nos; LP: index
        DIMENSION IROW(19,5),JR(19,2)
        DATA IROW/90*0/, JR/36*0/
C
        CALL ININD(1,MAXV(10))
        CALL STATIS(8,10)
        CALL INIT
        LEAP=12
660    LP=0
        CALL LNLUN
665    LP=LP+1
661    PRINT*, ' ENTER CI (1-3) AND ROW (1-6). (INFO=0,0)'
        READ*, (JR(LP,I),I=1,2)
        IF(JR(LP,1).EQ.0.OR.JR(LP,2).EQ.0)CALL INFO(LEAP,*661)
            CALL VERIFY(JR(LP,1),INV(15),*661)
            CALL VERIFY(JR(LP,2),6,*661)

```

```

      READ*, (IROW(LP,I),I=1,5)
      IF(LP.LT.19)THEN
        PRINT*, ' ANOTHER CHANGE FOR THIS UNIT? (1=YES)'
        READ*, IANS
        IF(IANS.EQ.1)GOTO 665
      ENDIF
667      DO 668 K=1,LP
          I=JR(K,1)
          J=5*JR(K,2)-5
          DO 664 L=1,5
              IAMAT(J+L,I)=IROW(K,L)
          CONTINUE
664      CONTINUE
668      CONTINUE
      CALL WINDOW
      CALL SAVAR(INV(10))
      IF(INV(7).EQ.2)THEN
        PRINT*, 'ANOTHER UNIT? (NO=0; YES=NEW UNIT NO.)'
        READ*, IANS
        IF(IANS.EQ.0)RETURN
        INV(10)=IANS
        IF(IND(9).EQ.1)GOTO 667
        GOTO 660
      ELSE
        IF(INV(10).GE.INV(9))RETURN
        INV(10)=INV(10)+1
        IF(INV(7).EQ.1)GOTO 660
        IF(INV(7).EQ.3)GOTO 667
      ENDIF
      RETURN
C      END      ROWCH
C      *****
C      SUBROUTINE DEFACT
C      Factors not saved in the files which affect demand
C      are listed. Changes are made by entering the factor
C      no. and its new value. Note: MOPP value is integer
C      and equals 100 times the real MOPP FACTOR.
C      LEAP: for use with info routine; IANS: var to be
C      changed; IANS+1: the index holding the new value.
C
15      PRINT*, 'CURRENT DEMAND FACTORS ARE:'
          WRITE(6,1515) (INV(I),I=2,5)
1515      FORMAT(/,1X,'1. MOPP = ',I3,' 2. INTENSITY = ',I1,
2          ' 3. PHASE = ',I1,' 4. NEW RETS = ',I3,/)
          LEAP=2
          PRINT*, ' ENTER FACTOR AND NEW VALUE. (NONE=0,0;INFO=0,FACTOR)'
20          READ*, IANS, INV(IANS+1)
              CALL VERIFY(IANS,4,*20)
              IANS = IANS+1
              IF(IANS.EQ.1)GOTO 25
              CALL VERIFY(INV(IANS),MAXV(IANS),*15)
              IF(IANS.EQ.3.OR.IANS.EQ.4)THEN @must exclude 0
                  CALL VERIFY((INV(IANS)-1),MAXV(IANS),*15)
              ENDIF
          GOTO 20

```

```

      IF(INV(3).NE.0.AND.INV(5).NE.4)CALL CHKRET   @rets
      RETURN
C      END          DEFACT
C      *****
SUBROUTINE CHKRET
C      Enter new ant/combat rets for wpn.
      IF(INV(5).GT.1)THEN          @new mnt returns
        PRINT*, ' ENTER MNT RETURNS FOR EACH CI.'
        READ*, (MNT(KROW,I),I=1,INV(15))
      ENDIF
C
      IF(INV(5).EQ.1.OR.INV(5).EQ.3)THEN
        PRINT*, ' ENTER COMBAT RESERVES FOR EACH CI.'
        READ*, (KMB(KROW,I),I=1,INV(15))
      ENDIF
      IF(IWPU(1,1).NE.8.AND.IWPU(1,1).NE.9)RETURN
      MX=5
      IF(IWPU(1,1).EQ.8)MX=3
      DO 414 I=1,INV(15)
        DO 415 J=2,MX
          MNT(J,I)=MNT(1,I)
          KMB(J,I)=KMB(1,I)
415      CONTINUE
414      CONTINUE
      RETURN
C      END          CHKRET
C      *****
SUBROUTINE OLDEM
c      If returns have been entered but no demand generated,
c      return loads calculated and added to old demand.
      DO 82 I=1,INV(15)
        IF(IWPU(KROW,1).EQ.9)THEN
          DO 822 J=1,5
            RETS=KMB(J,I)+(MNT(J,I)/2.)
            IF(INV(10).LT.42)THEN
              IAMAT(J*5,I)=NINT(IAMAT(J*5,I)+(IHOWL(J)*RETS))
            ELSE
              IAMAT(J*5,I)=NINT(IAMAT(J*5,I)+(IHOWL(J+5)*RETS))
            ENDIF
822      CONTINUE
          ELSE
            IF(IWPU(KROW,1).EQ.8)THEN
              DO 823 J=1,3
                RETS=KMB(J,I)+(MNT(J,I)/2.)
                IAMAT(J*5,I)=NINT(IAMAT(J*5,I)+(IHOWL(J+10)*RETS))
823      CONTINUE
              ELSE
                RETS=KMB(KROW,I)+(MNT(KROW,I)/2.)
                IAMAT(KCOL+4,I)=NINT(IAMAT(KCOL+4,I)+(LOAD(IWPU(KROW,1))*RETS))
              ENDIF
            ENDIF
82      CONTINUE
      RETURN
C      END SUB OLDEM

```

```

SUBROUTINE CURWPN(*)
C      Obtain current weapon system from file.
      IWPV(KROW,1)=0
      IF(IAMAT(KCOL,1).NE.0)GOTO 38
      PRINT*, ' NO WPN TYPE LISTED. ENTER WPN CODE WANTED'
      PRINT*, ' OR GO TO NEXT WPN. (0=NEXT WPN;INFO=99)'
      READ*, IWPV(KROW,1)
      IF(IWPV(KROW,1).EQ.0)RETURN 1
      LAP=2          @offset for NUWPN routine
      GOTO 387
38      IWPV(KROW,1)=IWPV(KROW,1)+1
      IF(IAMAT(KCOL,1).EQ.KWPSYS(IWPV(KROW,1)))RETURN
      IF(IWPV(KROW,1).LT.MAXV(9))GOTO 38
      CALL ERRWPN(*388)          @no match found
387     CALL NUWPN(LAP,*388,*389)    @to enter new wpn type
389     RETURN
389     RETURN 1
C      END SUB CURWPN
C      *****
SUBROUTINE NUWPN(LIP,*,*)
C      Enter weapon system and assign proper ammo code to
C      array. Howitzers automatically enter all ammo codes.
C      LIP = offset to determine start of routine.
      LEAP=4
      GOTO(36,361)LIP
36      PRINT*, ' THE WPN SYSTEM NUMBER IS: (INFO=99)'
      READ*, IWPV(KROW,1)
361     IF(IWPV(KROW,1).EQ.99)CALL INFO(LEAP,*36)
          IF(IWPV(KROW,1).NE.0)GOTO 360
          DO 606 I=1,INV(15)
              DO 607 J=KCOL,KCOL+4
                  IAMAT(J,I)=0
607         CONTINUE
606         CONTINUE
          RETURN 2
360     CALL VERIFY(IWPV(KROW,1),MAXV(9),*36)
          IF(IWPV(KROW,1).EQ.14)THEN
              PRINT*, ' ENTER NEW AMMO CODE NO.'
              READ*, IANS
              DO 14 I=1,INV(15)
                  IAMAT(KCOL,I)=IANS
14         CONTINUE
          RETURN 1
          ENDIF
          DO 16 I=1,INV(15)
              IAMAT(KCOL,I)=KWPSYS(IWPV(KROW,1))
              IF(IWPV(1,1).EQ.9)THEN    @howitzers
                  DO 161 J=1,4
                      IAMAT((5*J+1),I)=KHW(J)
161         CONTINUE
          ENDIF
          IF(IWPV(1,1).EQ.8)THEN
              DO 162 J=1,2
                  IAMAT((5*J+1),I)=KHW(J+4)

```

```

      ENDIF
16      CONTINUE
      RETURN 1
C      END SUB NUWPN
C      *****
SUBROUTINE COPYWP(NU,LRO,*)
C      Only the weapon whose ammo code is specified is
C      copied into the new unit. If demand is generated
C      for howitzers, it will be for all ammo types.
      DIMENSION KOPY(5,3)
      DATA KOPY/15*0/
      KOL=5*LRO-5
      DO 830 I=1,INV(15)
        DO 831 J=1,5
          KOPY(J,I)=IAMAT(KOL+J,I)
931      CONTINUE
930      CONTINUE
      INV(10)=NU          @new unit no.
      KROW=1
932      KCOL=5*KROW-4
        IF(IAMAT(KCOL,1).EQ.INV(14))GOTO 834
        KROW=KROW+1
        IF(KROW.LE.6)GOTO 932
        PRINT*, 'AMMO TYPE NOT FOUND.'
        RETURN 1
934      LRO=KROW
        DO 836 I=1,INV(15)
          DO 837 J=1,5
            IAMAT(KCOL+J,I)=KOPY(J,I)
837      CONTINUE
936      CONTINUE
      IF(IND(11).NE.1)RETURN 1
      IF(IAMAT(1,1).EQ.4.OR.IAMAT(1,1).EQ.6)THEN
        KROW=1
        KCOL=1
      ENDIF
      CALL DEMAND
      KCOL=5*LRO-4
      RETURN 1
C      END      COPYWP
C      *****
SUBROUTINE DUPE(N)
C      Data for one unit is duplicated into another unit.
C      New demand is generated if indicated. List and
C      save are polled. N=unit (record) number of copy.
      IF(IND(12).EQ.1)CALL COPYWP(N,KROW,*934)
      KROW=1          @move ptrs to start of array
      KCOL=1
      IF(IND(11).EQ.1)THEN
933      CALL DEMAND
        IND(8)=2          @avoid nextwpn poll
        CALL NXTWPN(*933)
        IND(8)=1          @reset copy indicator
      ENDIF

```

```

        CALL SAVAR(N)
        RETURN
C      END      DUPE
C      *****
SUBROUTINE DITTO
C      Copy one unit into many non-sequential locations.
930    PRINT*, ' ENTER NEW UNIT NO. (0=NONE)'
        READ*, IUNS
        IF(IUNS.EQ.0)RETURN
        CALL VERIFY(IUNS,MAXV(8),#930)
        CALL DUPE(IUNS)
        GOTO 930
C      END      DITTO
C      *****
SUBROUTINE ZOT
C      Set attribute array to 0.
        DO 81 I=1,INV(15)
            DO 82 J=1,MAXV(11)
                IAMAT(J,I)=0
82      CONTINUE
81      CONTINUE
        RETURN
C      END      ZOT
C      *****
SUBROUTINE INFO(JMP,*)
C      Print message at JMP offset and return to * location.
C
        GOTO(255,250,256,257,258,261,259,263,264,266,267,268)JMP
C      LEAP=2: demand factors
250    CALL VERIFY(INV(1),3,#249)
        GOTO(251,252,253,254)INV(1)
C      MOFF
251    PRINT*, ' ENTER AN INTEGER EQUAL TO 100*MOFF FACTOR'
        PRINT*, ' IN THE RANGE FROM 0 AND 100.'
        RETURN 1
C      battle intensity
252    PRINT*, ' BATTLE INTENSITY CHOICES INCLUDE:'
        PRINT*, ' 1 = DEFEND,INTENSE      5 = DELAY,MBA-LIGHT'
        PRINT*, ' 2 = DEFEND,MODERATE      6 = DELAY, CFA'
        PRINT*, ' 3 = DEFEND,LIGHT        7 = ATTACK'
        PRINT*, ' 4 = DELAY,MBA-INTENSE    8 = SPECIFY AMMO RATES'
        RETURN 1
C      time of day
253    PRINT*, ' ENTER PHASE, DAY/NIGHT. (1 = day, 2 = night)'
        RETURN 1
C      additional wpn returns
254    PRINT*, ' "MORE RETS" SHOWS IF COMBAT AND MNT RETURNS'
        PRINT*, ' HAVE BEEN ENTERED FOR THIS WPN: 0=NO; 4=YES.'
        PRINT*, ' TO ENTER NEW RETS, USER ENTERS A FIGURE:'
        PRINT*, ' 1=COMBAT RESERVES; 2=MNT RETURNS; 3=20TH.'
        RETURN 1
C      LEAP=6: CI nos.
261    PRINT*, ' THE NO. OF CIS INVOLVED:  THE NO. WORKED'
        PRINT*, ' WITH (1-3)...THE NO. IN THE PHASE. THE'

```

```

PRINT*, ' AND DO NOT AFFECT DATA FILES. ENTER 1 FOR CI.'
RETURN 1
C      LEAP=1: types of operations and methods available.
255 PRINT*, ' 'PRINT' PRINTS FILE DATA FOR UNITS SPECIFIED.'
PRINT*, ' CAN LIST LOCALLY OR AT DFFD. NO CHANGES MADE.'
PRINT*, ' 'EDIT' ACCEPTS NEW VALUES FROM USER. USER'
PRINT*, ' SPECIFIES WHICH OPTIONS TO INCLUDE. DEMAND CAN'
PRINT*, ' CAN BE GENERATED. IN 'COPY,' USER COPIES'
PRINT*, ' ONE SEQUENCE OF UNITS INTO ANOTHER. OPTIONS'
PRINT*, ' FOR PRINT, NEW DEMAND, SAVE ARE AVAILABLE.'
PRINT*, ' 'BY ROW' PERMITS CHANGES ONLY ON INDIV. ROWS.'
PRINT*, ' OF UNIT ARRAY. USER ENTERS ROW #, CI # (1,2,3)'
PRINT*, ' AND VALUES FOR EACH OF THE 5 COLUMNS.'
264 PRINT*, ' 'RANDOM' ALLOWS USER TO WORK ON INDIV. UNITS.'
PRINT*, ' 'SEQUENCE' PERMITS WORK ON UNITS IN SEQ. ORDER.'
PRINT*, ' 1ST AND LAST UNIT NOS ARE ENTERED AND STATUS'
PRINT*, ' OPTIONS SET. EACH UNIT IS CALLED IN TURN.'
PRINT*, ' SEQ. MAY BE INTERRUPTED BY ENTERING 999 IN'
PRINT*, ' RESPONSE TO 'ANOTHER WPN FOR THIS UNIT?''
PRINT*, ' 'GROUP' PERMITS ITEMS FOR ONE UNIT TO BE'
PRINT*, ' DUPLICATED IN ALL UNITS OF THE GROUP.'
      RETURN 1
C      LEAP=3: unit nos. and names
256 PRINT*, ' UNIT NOS. RANGE FROM 1 TO ', MAXV(8), '. '
PRINT*, ' DO YOU WANT A LISTING OF THE NAMES? (1=YES)'
      READ*, IANS
      IF(IANS.NE.1)RETURN 1
      WRITE(6,1256) (I,UNAME(I),I=1,MAXV(8))
1256      FORMAT(/,4(3X,I2,', ',A8))
      RETURN 1
C      LEAP = 4: weapon systems
257 PRINT*, ' WPN SYSTEM TYPES INCLUDE:'
      WRITE(6,1257) (I,WPSYS(I),I=1,MAXV(9)),0,'NONE'
1257      FORMAT(/,4(3X,I2,', ',A10))
      RETURN 1
C      LEAP = 5: options & status
258 PRINT*, ' IF STATUS OF OPTION IS SET TO 1, THAT OPTION'
PRINT*, ' WILL BE USED. IF SET TO 0, IT WONT BE USED.'
PRINT*, ' FOR OPTS 1-6, IF THE STATUS=1, USFR IS ASKED'
PRINT*, ' TO ENTER CHANGES (USER MUST SUPPLY THE CORRECT'
PRINT*, ' NO. OF RESPONSES TO EACH QUESTION. . .IF ASKED'
PRINT*, ' FOR ENTRIES 'PER CI', ONE ANSWER FOR EACH CI'
PRINT*, ' MUST BE GIVEN). IF STATUS=0, THE VALUE OF THE'
PRINT*, ' OPTION WILL REMAIN THE SAME. IF OPTION 7=1,'
PRINT*, ' DEMAND CAN RE INSERTED. IF '8'=1, A UNIT CAN'
PRINT*, ' BE COPIED INTO DIFFERENT LOCATIONS--ONE AT A'
PRINT*, ' TIME. '9'=1 AUTOMATICALLY LISTS UNIT DATA'
PRINT*, ' '10'=1 AUTOMATICALLY SAVES THE DATA. IF ?=0,'
PRINT*, ' OR 10=0, USER IS POLLED AFTER EACH UNIT. IF'
PRINT*, ' 11=1, NEW DFMAND WILL BE GENERATED. IF 12=1,'
PRINT*, ' USER INSERTS AN AMMO CODE TO MAKE INDIVIDUAL'
PRINT*, ' WEAPON CHANGES. CODE MUST MATCH ONE IN ARRAY'
PRINT*, ' AND CODES MAY BE GIVEN IN ANY ORDR.'
PRINT*, ' NOTE: DIFFERENT ROUTINES PERMIT DIFF. OPTS.'

```

```

PRINT*, ' TO ZERO OUT REMAINING UNIT FOR FULL PHASE.'
PRINT*, ' USFR SELECTS OPTION 1 (wpns alive) AND ENTERS'
PRINT*, ' 9999 AS FIRST RESPONSE TO 'WPNS ALIVE.'
RETURN 1
C      LEAP=7: next step
259 PRINT*, ' SELECT MANNER OF CONTINUATION: '1' REPEATS'
PRINT*, ' SAME METHOD. UNIT NOS AND OPTION STATUS CAN BE'
PRINT*, ' CHANGED. '2' PERMITS THE METHOD TO BE CHANGED.'
PRINT*, ' '3' COPIES LAST SEQ OF UNITS. USER IS ASKED TO'
PRINT*, ' ENTER THE FIRST UNIT NO. OF THE NEW SEQUENCE.'
PRINT*, ' '4' BEGINS A NEW RUN WITH DIFF CIS *IFF* EXTRA'
PRINT*, ' FILES WERE ASSIGNED BY THE SSG RUNSTREAM.'
PRINT*, ' '5' PRINTS FROM THE FILES--LAST SEQUENCE IS'
PRINT*, ' LISTED LOCALLY OR ENTIRE FILE IS COPIED INTO'
PRINT*, ' 'REPORT' TO BE SENT TO DPFO . '6' ENDS RUN.'
RETURN 1
C      LEAP=8: file treatment
263 PRINT*, ' IN (1) A NEW FILE OR NEW CYCLE OF AN OLD FILE'
PRINT*, ' IS AUTOMATICALLY COPIED FROM AN OLD ONE.'
PRINT*, ' IN (2) A NEW FILE IS INITIALIZED TO ZERO.'
PRINT*, ' IN (3) ONLY PART OF A FILE IS TO BE COPIED'
PRINT*, ' INTO ANOTHER. USER WILL BE POLLED BEFORE EACH'
PRINT*, ' FETCH TO DETERMINE WHICH FILE IS TO BE READ.'
RETURN 1
C      LEAP = 10: ammo codes
266 PRINT*, ' AMMO CODES MUST MATCH AN ITEM IN THE 1ST COL'
PRINT*, ' IN THE ARRAY SHOWN. POSSIBLE VALUES INCLUDE:'
WRITE(6,1266) (KWPSYS(I),I=1,13)
1266 FORMAT(/,1X,12(I3,' '),I3)
RETURN 1
C      LEAP=11: options for wpns alive
267 PRINT*, ' WPNS ALIVE ARE TO BE ENTERED FOR EACH CI'
PRINT*, ' SEPARATELY. TO ZERO OUT THE REMAINDER OF A'
PRINT*, ' UNIT FOR THE ENTIRE PHASE (ALL CIS INVOLVED)'
PRINT*, ' USER MUST ENTER 9999 AS 1ST ENTRY AND ANY'
PRINT*, ' NOS. FOR REMAINING ENTRIES. TO ALLOW 0 DEMAND'
PRINT*, ' FOR A WPN, 0 MUST BE ENTERED FOR WPNS SHORT'
PRINT*, ' AND NO RETS MUST BE ENTERED. IF A UNIT HAS'
PRINT*, ' RETURNS ONLY (NO. OF WPNS SHORT=0), DEMAND'
PRINT*, ' MUST STILL BE GENERATED (OPT 11=1).'
RETURN 1
C      LEAP=12: changes inserted by row locations
268 PRINT*, ' TO CHANGE BY ROW, USER ENTERS THE CI NO.'
PRINT*, ' (1,2,3) CORRESPONDING TO 1st,2nd or 3rd CI'
PRINT*, ' INVOLVED AND THE ROW (1-4). THE 2ND PROMPT'
PRINT*, ' ASKS THAT ALL 5 VALUES OF ROW BE ENTERED.'
PRINT*, ' CIS AND ROWS CAN BE CHANGED IN ANY ORDER.'
PRINT*, ' UP TO 18 'ROWS' CAN BE CHANGED AT ONE TIME AND'
PRINT*, ' ROW CHANGES CAN BE COPIED INTO ANOTHER UNIT IF'
PRINT*, ' THE COPY UNIT OPTION IS SET TO 1. USER WILL BE'
PRINT*, ' PROMPTED TO ENTER THE UNIT NO. IF NEW CHANGES'
PRINT*, ' ARE DESIRED.'
249 RETURN 1
C      RETURN

```

```

C      *****
C      SUBROUTINE ERRWPN(*)
C      If stored wpn type is not listed, determine which
C      way to proceed: by correcting file or by entering
C      names and expenditure rates.
C      PRINT*, ' BECAUSE WPN SYSTEM IN TABLE IS NOT LISTED,'
C      PRINT*, ' EITHER THE WPN TYPE MUST BE CHANGED OR THE'
C      PRINT*, ' WPN NAME AND EXPENDITURE RATES ENTERED.'
C      PRINT*, ' (1=CHANGE WPN TYPE, 0=ADD WPN NAME AND RATES .)'
C      READ*, IANS
C      IF(IANS.EQ.1)RETURN
C      INV(3)=9
C      IWPV(KROW,1)=14
C      RETURN 1
C      END      ERRWPN
C      *****
C      SUBROUTINE VERIFY(II,JJ,*)
C      Verify range of INV variable.  II=INV; JJ=MAXV.
C      IF(II.GE.0.AND.II.LE.JJ)RETURN
C      PRINT*, ' INAPPROPRIATE ENTRY, PLEASE REPEAT.'
C      RETURN 1
C      END      VERIFY
C      *****
C      SUBROUTINE FILERR(*)
C      file manipulation error routine.
555  PRINT*, ' ERROR IN FILE MANIPULATION.'
C      CALL WINDOW
C      CALL PLIST
C      WRITE(6,1555) (INV(L),L=1,15),(IND(M),M=1,12)
909  PRINT*, ' TRY AGAIN? (1=YES)'
C      READ*, IANS
C      IF(IANS.EQ.1)RETURN 1
1555 FORMAT(/,1X,15I3,2X,12I2)
C      STOP
C      END      9MAIN

```

7. UNIVAC EXECUTIVE LANGUAGE RUNSTREAMS.

a. These runstreams assign the files needed to run the programs associated with ARM. They are written for the UNIVAC Symbolic Stream Generator (SSG processor). Instructions on the use of these runstreams are found in Volume I.

b. The SSG runstream needed for running the ARM simulation is as follows:

```
RETAIN ARMPL.,osinname$,oss#file.
@ASS,CP RPTFILES,1,1,11(+1),F/100//100
@USE 2.,RPTFILES,1,1,11(+1).
@ASS,A DATABFILES,1,1,11.
@USE 3.,DATABFILES,1,1,11.
@ASS,A DEMAND*FILES,1,1,212IN.
@USE 9,DEMAND*FILES,1,1,212IN.
@ASS,A EVFILES,1,1,11.
@USE 7.,EVFILES,1,1,11.
@ASS,CP TDATABFILES,1,1,21(+1),F//25
@USE 4.,TDATABFILES,1,1,21(+1).
@ASS,CP EVFILES,1,1,21(+1),F//15
@USE 8.,EVFILES,1,1,21(+1).
@ASS,A EVENTFILES,1,1,11.
@USE 11,EVENTFILES,1,1,11.
@ASS,CP STATFILES,1,1,11(+1),F//75
@USE 14,STATFILES,1,1,11(+1).
```

c. This runstream assigns the files used in the EDIT program:

```
RETAIN ARMPL.,osinname$,oss#file.
@ASS,A TDATABEDIT,1,1,21.
@ASS,CP DATABEDIT,1,1,21(+1),F//25
@USE 2.,TDATABEDIT,1,1,21.
@USE 4.,DATABEDIT,1,1,21(+1).
@ASS,CP TRKPRTEEDIT,1,1,11(+1),F//4
@USE 2,TRKPRTEEDIT,1,1,11(+1).
@ASS,A DIS4+DISFILESEDIT,1,1,21.
@USE 13,DIS4+DISFILESEDIT,1,1,21.
```

d. The demand generation program may be accessed by using this runstream:

```
#RETAIN
*IF INEW1
*INCREMENT A TO INUMCIS,1,1,11
*SET D TO 11+A
#ASS,UP LOGSNEW,1,1,AIBIN(+1).
#USE I*D1,LOGSNEW,1,1,AIBIN(+1).
*LOOP
*ELSE
*INCREMENT A TO INUMCIS,1,1,11
#ASS,A LOGSOLDFILE,1,1,AIBIN.
*SET C TO A+9
#USE I*D1,LOGSOLDFILE,1,1,AIBIN.
#ASS,UP LOGSNEWFILE,1,1,AIBIN(+1).
*SET D TO 11+A
#USE I*D1,LOGSNEWFILE,1,1,AIBIN(+1).
*LOOP
*END
*IF [REPORT1
*IF INEW1
*INCREMENT B TO INUMCIS,1,1,11
#ASS,UP LOGRPTNEW,1,1,B1(+1).
*SET D TO 14+B
#USE I*D1,LOGRPTNEW,1,1,B1(+1).
*LOOP
*ELSE
*INCREMENT B TO INUMCIS,1,1,11
#ASS,UP LOGRPTNEWFILE,1,1,B1(+1).
*SET D TO 14+B
#USE I*D1,LOGRPTNEWFILE,1,1,B1(+1).
*LOOP
*END
*END
```

END

FILMED

9-83

DTIC