UNCLASSIFIED

MARCH 77

14

TR 3-77

ACN 21698

ADA040958

CACDA JIFFY WAR GAME

PROGRAMERS MANUAL

Technical Report TR 3-77

COPY AVAILABLE TO DDC DOES NOT PERMIT FULLY LEGIBLE PRODUCTION

UNITED STATES ARMY COMBINED ARMS CENTER

COMBINED ARMS
COMBAT DEVELOPMENTS ACTIVITY

Approved for public release
DISTRIBUTION UNLIMITED

COMBAT OPERATIONS ANALYSIS DIRECTORATE

UNCLASSIFIED

Technical Report TR 3-77
March 1977

Directorate of Combat Operations Analysis
US Army Combined Arms Combat Developments Activity
Fort Leavenworth, Kansas 66027

CACDA JIFFY WAR GAME

PROGRAMMERS MANUAL

by
Mr Timothy J. Bailey
Mr Gerald A. Martin
and
Mr Joseph AuBuchon

ACN 21698

Approved by:

Richard C. Rinkel
Chief, Analysis Division

Leland C. Pleger
Technical Director

Reed E. Davis, Jr.
Colonel, IN
Director

BEST

AVAILABLE

COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report 3-77	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CACDA Jiffy War Game Programmers Manual		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER CACDA-TR-3-77
7. AUTHOR(s) Mr Timothy J. Bailey Mr Gerald A. Martin Mr Joseph AuBuchon		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Combined Arms Combat Developments Activity ATTN: ATCA-CAA-A Fort Leavenworth, KS 66027		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 1977
		13. NUMBER OF PAGES 320
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release
Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

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

Jiffy Game
SCORES
Flow Diagrams
FORTRAN Code

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

The CACDA Jiffy War Game Programmers Manual is one report of a set of three reports which were produced to document the Jiffy Game which is a Corps level war game run in support of the TRADOC Scenario Oriented Recurring Evaluation System (SCORES). The programmers manual contains descriptions, flow diagrams, and the FORTRAN codes for all the computer routines of the Jiffy Game. The other two reports of the documentation set are the CACDA Jiffy War Game Technical Manual (methodology and data appendixes) and the CACDA Jiffy War Game Users Manual.

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

The Jiffy Game has existed, as a manual war game, since the late 1960's. In its early stages, the game was completely manual; and correspondingly, its assessment methodology was simplistic, based on the firepower scores of a few key weapon systems. In late 1973, USATRADOC established the Scenario Oriented Recurring Evaluation System (SCORES), the standard scenario development process to be based on the Jiffy Game. With the advent of SCORES, it was recognized that the simplistic, firepower score-driven Jiffy Game, although responsive, was not of adequate resolution to produce the quality product expected from SCORES. Thus, the Jiffy Game underwent major methodology modifications, which allowed the gaming of the complete spectrum of conventional weapon systems and upgraded the assessment methodologies to use weapon characteristics instead of firepower scores as the basis for assessments. However, as the level of detail increased, the number of manual calculations and the amount of data required to make the calculations also increased. Finally, it became necessary to automate the assessment calculations to maintain the Jiffy Game's responsiveness. The automation process was completed in May 1975. This methodology was developed principally by MAJ Karl Lowe, assisted by LTC Tom Buff, MAJ Ken Nash, and MAJ Bob Riddick, and was documented in July 1975 with the publishing of the USACACDA SCORES "JIFFY" War Gaming Methodology.

In the fall of 1975, as a quality assurance measure, the Jiffy Game methodology was subjected to sensitivity analysis. A Jiffy Game improvement program was initiated as a result of the analysis. The improvement program consisted basically of three tasks. First, the assessment methodology needed further modification and improvement in certain areas. Second, the capability to maintain on computer files a hierarchy of units consistent with the overall gaming methodology was to be added to the Jiffy Game. Finally, detailed documentation of the revised methodology and all supporting computer programs was to be published. This report was produced as a result of the improvement program as a portion of the Jiffy Game documentation.

The authors of this report wish to acknowledge the SCORES war gaming staff of the Combined Arms Combat Developments Activity (CACDA) who served as consultants during the preparation of this report. Special thanks are given to Mrs. Elizabeth Etheridge who served as technical editor for this report and to Miss Laura B. Weishaar who typed the report.

ABSTRACT

This report is one of a set of three reports produced to document the automated features of the Combined Arms Combat Developments Activity (CACDA) "Jiffy" war gaming process. This process was developed to support the USATRADOC Scenario Oriented Recurring Evaluation System (SCORES) scenario development and force evaluation efforts. This report consists of descriptions, logic flow diagrams, and the FORTRAN code for all the programs and routines associated with the "Jiffy" war gaming process. The other two reports in the set are the CACDA Jiffy War Game Technical Manual and the CACDA Jiffy War Game Users Manual. The technical manual consists of two parts. Part 1 contains the methodologies used in the automated routines of the Jiffy Game, the computer model run in support of the CACDA "Jiffy" war gaming process, and an unclassified data base. Part 2 contains all classified data and its sources used in the Jiffy Game during secure production runs. The users manual contains a discussion of the manual aspects and the automated features of the gaming process and also presents an unclassified sample run.

TABLE OF CONTENTS

FOREWORD ii

ABSTRACT iii

LIST OF TABLES vi

LIST OF FIGURES vii

SCOPE 1

GENERAL 1

FORCE STRUCTURE GENERATION PROGRAMS 1

General 1

File Organization 2

Program Descriptions 2

SRC Program 2

UNIT Program 2

PARENT Program 10

FORCE Program 10

JIFFY GAME 25

General 25

Program Descriptions 25

Overlay 0 (SUPER, INIT, INDEX5, LOSS and DISPLAY) 25

Overlay 1 (ROFA) 55

Overlay 2 (TANK) 55

Overlay 3 (INFANT) 55

Overlay 4 (MINE and FASCAM) 73

Overlay 5 (AHAD) 73

Overlay 6 (CANNON and CLGP) 82

TABLE OF CONTENTS (CONCLUDED)

Overlay 8 (SUPRES) 104

Overlay 9 (RESULT) 104

Overlay 10 (FORCE) 104

Overlay 11 (APPORT) 104

Overlay 12 (BUILD) 127

APPENDIXES

A. Indexed Sequential File Creation Programs A-1

B. SRC Program Listing B-1

C. UNIT Program Listing C-1

D. PARENT Program Listing D-1

E. FORCE Program Listing E-1

F. OVLY 0 Program Codes and Lists of Variables F-1

G. OVLY 1 Program Codes and Lists of Variables G-1

H. OVLY 2 Program Codes and Lists of Variables H-1

I. OVLY 3 Program Code and Lists of Variables I-1

J. OVLY 4 Program Code and Lists of Variables J-1

K. OVLY 5 Program Code and Lists of Variables K-1

L. OVLY 6 Program Code and Lists of Variables L-1

M. OVLY 8 Program Code and Lists of Variables M-1

N. OVLY 9 Program Code and Lists of Variables N-1

O. OVLY 10 Program Code and Lists of Variables O-1

P. OVLY 11 Program Code and Lists of Variables P-1

Q. OVLY 12 Program Code and Lists of Variables Q-1

R. Distribution R-1

LIST OF TABLES

		Page
1.	Control point gamer decisions.	35
2.	OVLY 10 force manipulation options.	108
3.	Types of displays.	109
B-1.	List of variables for SRC program.	B-2
C-1.	List of variables for UNIT program.	C-2
D-1.	List of variables for PARENT program.	D-2
E-1.	List of variables for FORCE program.	E-2
F-1.	Jiffy Game common variables.	F-2
F-2.	Program variables for SUPER.	F-4
F-3.	Program variables for INDEX5.	F-12
F-4.	Program variables for LOSS.	F-14
F-5.	Program variables for DISPLAY.	F-16
G-1.	Program variables for OVLY 1 (ROFA).	G-2
H-1.	Program variables for OVLY 2 (TANK).	H-2
I-1.	Program variables for OVLY 3 (INFANT).	I-2
J-1.	Program variables for MINE.	J-2
J-2.	Program variables for FASCAM.	J-9
K-1.	Program variables for OVLY 5 (AHAD).	K-2
L-1.	Program variables for CANNON.	L-2
L-2.	Program variables for CLGP.	L-16
M-1.	Program variables for SUPRES.	M-2
N-1.	Program variables for OVLY 9 (RESULT).	N-2
O-1.	Program variables for FORCE.	O-2
P-1.	Program variables for APPORT.	P-2
Q-1.	Program variables for BUILD.	Q-2

LIST OF FIGURES

		Page
1.	Jiffy Game file formats.	3
2.	SRC program logic flow diagram.	4
3.	UNIT program logic flow diagram.	11
4.	PARENT program logic flow diagram.	18
5.	FORCE program logic flow diagram.	26
6.	Jiffy game functional flow diagram.	34
7.	SUPER flow diagram.	36
8.	INIT flow diagram.	48
9.	INDEX5 flow diagram.	49
10.	LOSS flow diagram.	50
11.	DISPLAY flow diagram.	51
12.	ROFA (OVLY 1) flow diagram.	56
13.	TANK (OVLY 2) flow diagram.	62
14.	INFANT (OVLY 3) flow diagram.	69
15.	MINE flow diagram.	74
16.	Subroutine FASCAM flow diagram.	79
17.	OVLY 5 (AHAD) flow diagram.	83
18.	CANNON flow diagram.	94
19.	Subroutine CLGP flow diagram.	102
20.	SUPRES flow diagram.	105
21.	OVLY 9 (RESULT) flow diagram.	106
22.	FORCE flow diagram.	110
23.	APPORT flow diagram.	128

LIST OF FIGURES (CONTINUED)

		Page
A-1.	Create program for SRC file.	A-2
A-2.	Create program for UNIT file.	A-3
A-3.	Create program for PARENT file.	A-4
A-4.	Create program for FORCE file.	A-5
A-5.	Create program for HISTORY file.	A-6
B-1.	SRC program code.	B-3
C-1.	UNIT program code.	C-3
D-1.	PARENT program code.	D-3
E-1.	FORCE program code.	E-3
F-1.	SUPER program code.	F-5
F-2.	INIT program code.	F-11
F-3.	INDEX5 program code.	F-13
F-4.	LOSS program code.	F-15
F-5.	DISPLAY program code.	F-17
G-1.	OVLY 1 (ROFA) program code.	G-3
H-1.	OVLY 2 (TANK) program code.	H-5
I-1.	OVLY 3 (INFANT) program code.	I-4
J-1.	MINE program code.	J-4
J-2.	FASCAM program code.	J-10
K-1.	OVLY 5 (AHAD) program code.	K-6
L-1.	CANNON program code.	L-5
L-2.	CLGP program code.	L-17
M-1.	OVLY 8 (SUPRES) program code.	M-3

LIST OF FIGURES (CONCLUDED)

		Page
N-1.	OVLY 9 (RESULT) program code.	N-5
O-1.	OVLY 10 (FORCE program code.	O-3
P-1.	OVLY 11 (APPORT) program code.	P-4
Q-1.	OVLY 12 (BUILD) program code.	Q-3

CACDA JIFFY WAR GAME
PROGRAMMERS MANUAL

1. SCOPE. This manual was prepared to document the computer programs associated with the CACDA "Jiffy" war gaming process. The documentation of each subroutine, program, and overlay includes a discussion of the functions performed by the routine, a logic flow diagram, a list of variables, and a listing of the FORTRAN code of the routine.

2. GENERAL. The interactive programs and data files that support the CACDA "Jiffy" war gaming process reside in permanent file storage on the Control Data Corporation (CDC) 6400/6500 multiprocessor computer located at Fort Leavenworth, Kansas. The programs are written in FORTRAN and are machine dependent due to extensive use of CDC Extended FORTRAN file handling features. There are basically two groups of programs that support the CACDA "Jiffy" war gaming process:

- a set of four programs that create and maintain the files necessary for force structure generation

- the Jiffy Game program.

The four force structure generation programs are small programs that allow the gamers to build interactively a hierarchy of files based on the Army's concept of Tables of Organization and Equipment (TOE) with which they can generate task organized forces for combat assessments in the Jiffy Game. The Jiffy Game operates on these forces and determines the number of personnel casualties and weapon system losses each force suffers in combat. In addition, the Jiffy Game generates a file containing a history of the forces and the losses they incurred for the combat it has processed.

3. FORCE STRUCTURE GENERATION PROGRAMS.

a. General. A hierarchy of four interactive programs has been developed to provide nontechnical military personnel with the capability to develop systematically a set of data files from which they can generate task organized forces for assessment evaluation in the Jiffy Game. The force structure generation is based on the US Army TOE standard requirements codes (SRCs). The SRCs define the types and quantities of weapon systems found in specific subunit organizations; e.g., an infantry squad or a tank platoon. The first program of the force generation hierarchy interactively develops a data base file of SRCs for each force. Since there is little variation in the composition of these subunit SRCs, the SRC data base, once completed, will be readily available for immediate application to any Jiffy Game-supported study. The second of the force generation programs uses the SRC data base to build interactively a file of the combat units through specification of a unique name and all SRCs that compose each unit. The file of units is then task organized into higher echelon organizations called parent units. A file of the parent units is created interactively by the third program of the hierarchy. Finally, the information on the SRC, unit, and parent unit files is consolidated into a file of the forces to be considered for combat assessments in the Jiffy Game.

b. File Organization. The type of files used in the force structure generation process and the Jiffy Game HISTORY file are CDC index sequential-random access files. These files are created and manipulated by file handling macros unique to the CDC operating systems. The files used for this application are random access files whose keys are contained in the first 20 characters (two words) of the record (the HISTORY file uses 30 character keys). The keys are arranged in sequential order in the random access index table, which allows sequential, in addition to random, accessing of the records on the file. The record formats for the four force generation files and the HISTORY file are illustrated in figure 1. Before any operations may be performed on these files, they must be created and put into permanent file storage space. This initialization process is accomplished through the execution of a small file creation program, which simply specifies the parameters essential for proper file definition. The FORTRAN programs for the creation of all five index sequential-random access files are contained in appendix A to this volume.

c. Program Descriptions.

(1) SRC program. The SRC program interactively builds the TOE SRC data base file. As noted above, this file is an indexed sequential-random access file. Each record of the SRC file contains an SRC identification word (1 to 10 alphanumeric characters) and up to 22 groups of weapon system item codes (Technical Manual, Part 2, Appendix A, table A-1) and the quantity of each type of weapon system assigned to the SRC. The format of the records of the SRC file is illustrated in figure 1(a). In addition to creating the SRC data base file, the SRC program has the capability to review any SRC that exists in the data base, add new SRCs to the file, change the quantity and/or type of personnel or weapon systems in a given SRC, delete specified SRCs, and list all SRCs with the quantity and type of weapon systems and personnel found in them. A logic flow diagram of the SRC program is provided in figure 2. A listing of the program code and a list of the program variables is contained in appendix B to this volume.

(2) UNIT program. Execution of the UNIT program is the second step in the force structure generation process. The UNIT program accesses the information stored on the SRC file and defines the combat units to be gamed. The program builds an indexed sequential-random access file whose records correspond to the combat units. The format of the UNIT file records is given in figure 1(b). Each record contains the unit name (1 to 10 alphanumeric characters) and up to 22 valid SRCs (the SRCs must exist on the SRC file). The SRCs specified with a unit correspond to the subunit organizations that compose the unit. For example, the SRCs specified for a tank company could possibly be a tank platoon SRC (specified three times) and a tank company headquarters SRC. In addition to building the UNIT file, the UNIT program has the capability to review the SRCs in a unit already on file, add

FORCE COLOR R/B	SRC #	ITEM CODE #1	QTY OF ITEM CODE #1	ITEM CODE #2	QTY OF ITEM CODE #2	...	...	...	QTY OF ITEM CODE #22

(a) SRC File Record

FORCE COLOR R/B	UNIT ID	SRC #1	SRC #2	...	...	...	...	...	SRC #22

(b) UNIT File Record

FORCE COLOR R/B	PARENT ID	UNIT ID #1	UNIT ID #2	...	...	...	...	...	UNIT ID #18

(c) PARENT File Record

PARENT ID	UNIT ID	FORCE COLOR	SECTOR	CRITICAL INCIDENT	FPS AT 100% STRENGTH	COMBAT INTENSITY LEVEL	PERCENT STRENGTH	BLANK	BLANK
QTY OF ITEM CODE #1	QTY OF ITEM CODE #2	QTY OF ITEM CODE #3	...	...	...	...	...	...	QTY OF ITEM CODE #80

(d) FORCE File Record

CRITICAL INCIDENT	PARENT ID	UNIT ID	SECTOR	FORCE COLOR	FPS AT 100% STRENGTH	COMBAT INTENSITY LEVEL	PERCENT STRENGTH	...	QTY OF ITEM CODE #80

(e) HISTORY File Record

Figure 1. Jiffy Game file formats.

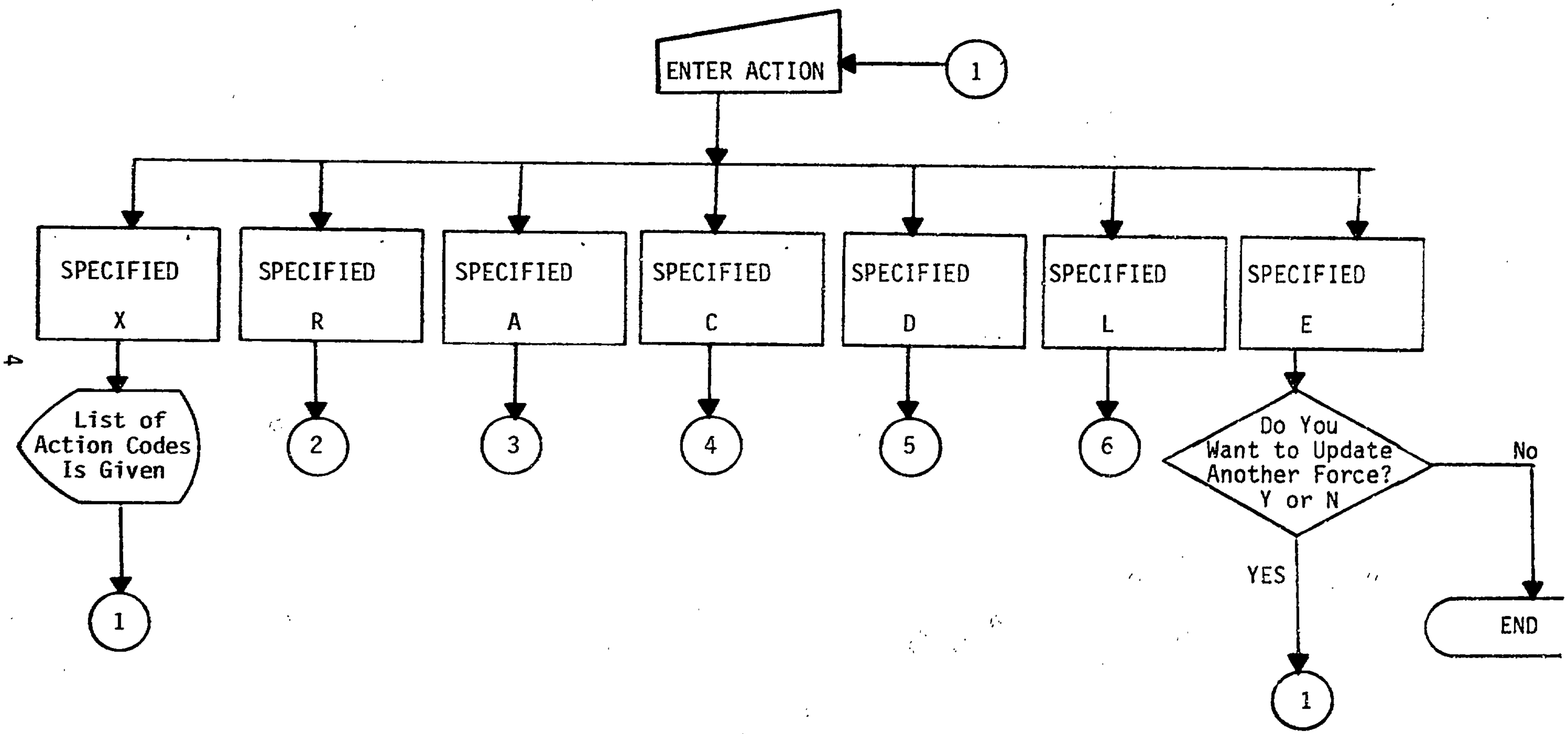

Figure 2. SRC program logic flow diagram. (continued next page)

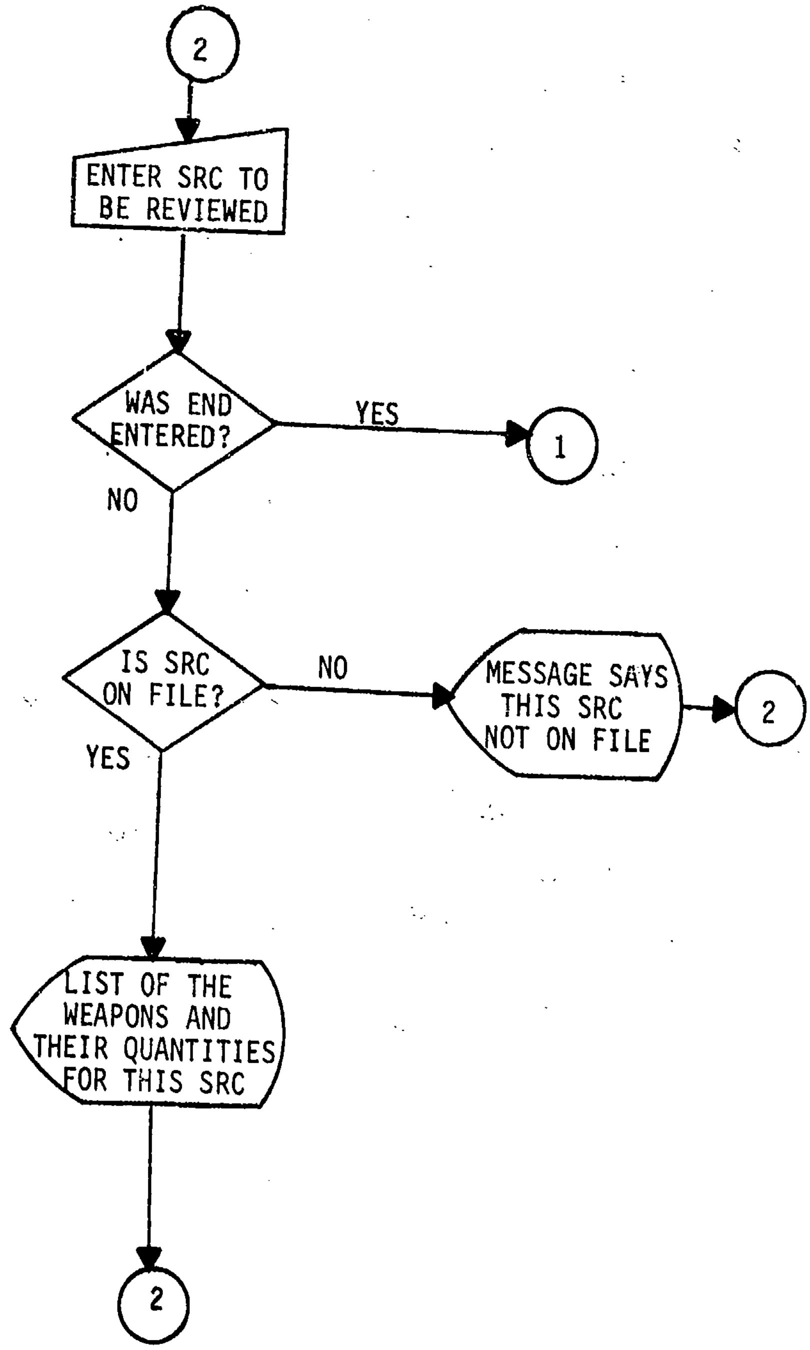

Figure 2. SRC program logic flow diagram (continued).

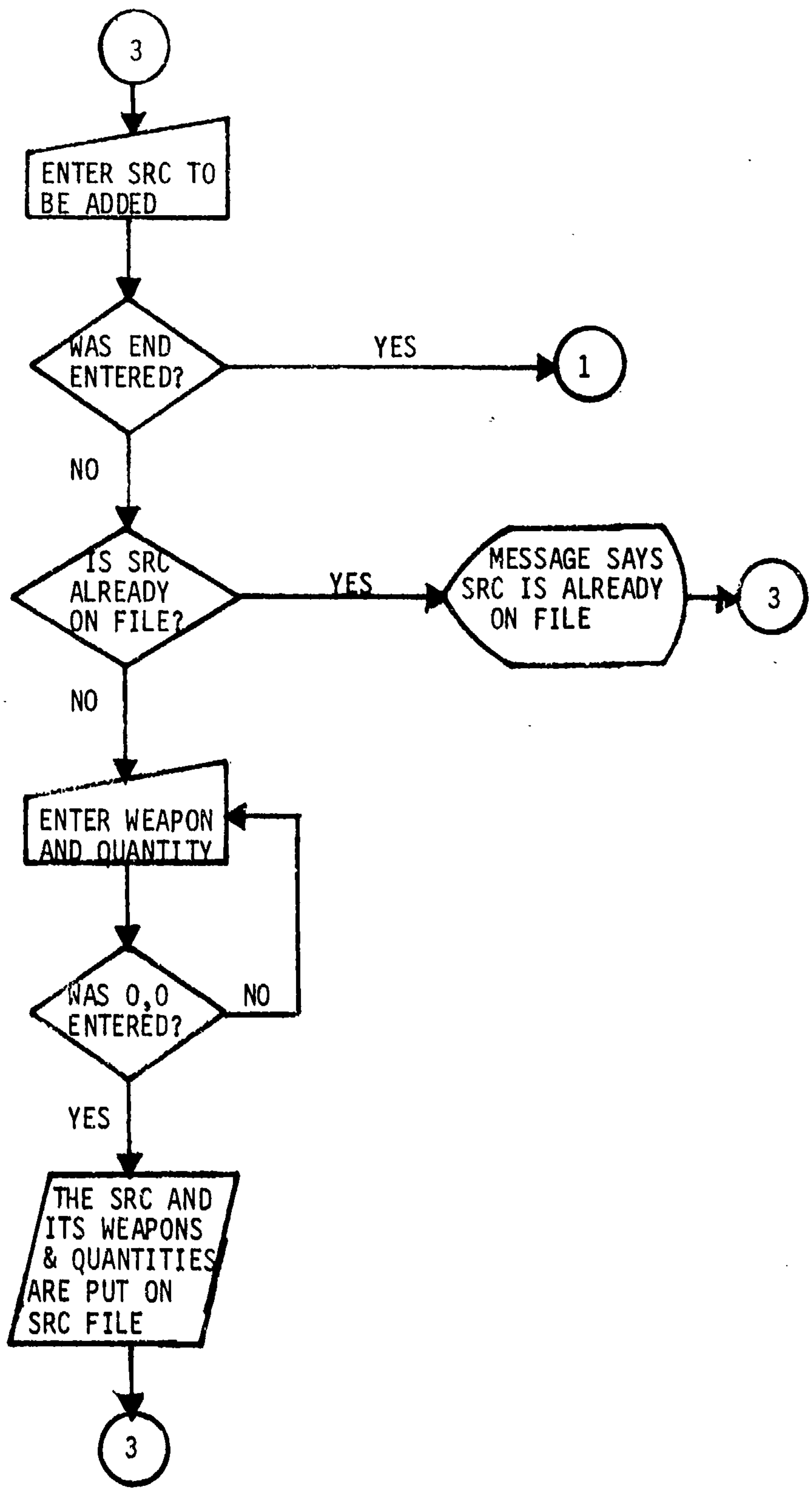

Figure 2. SRC program logic flow diagram (continued).

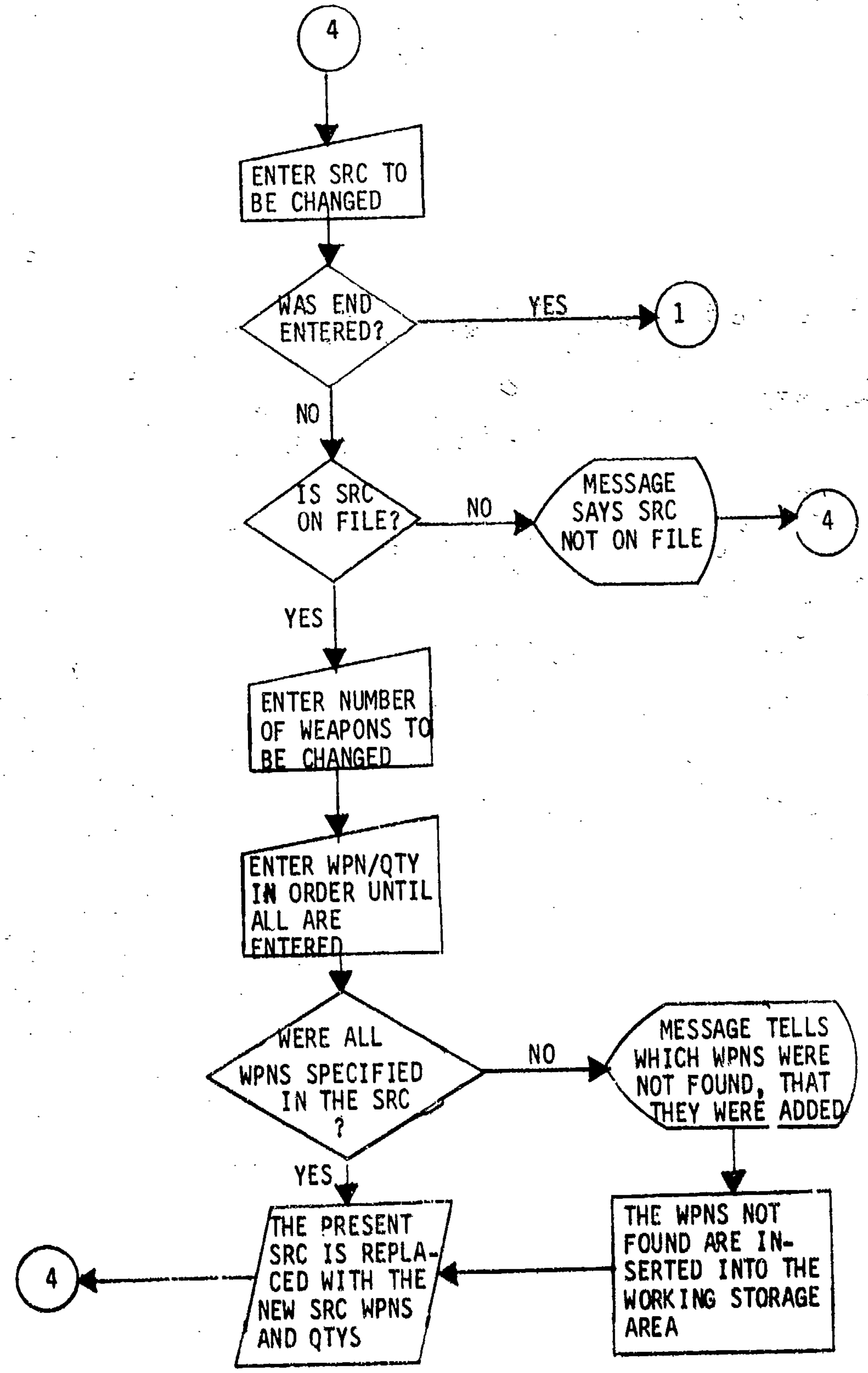

Figure 2. SRC program logic flow diagram (continued).

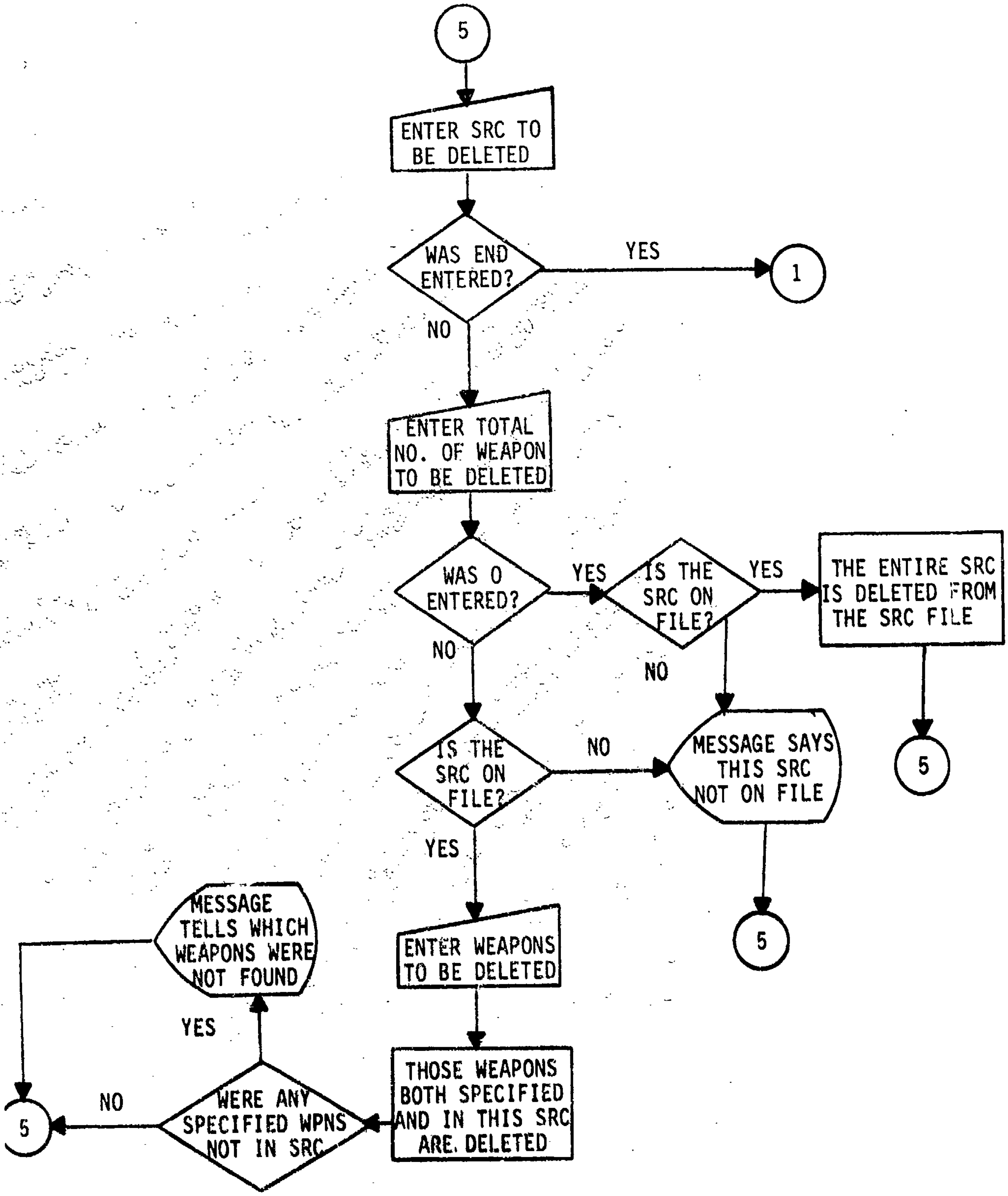

Figure 2. SRC program logic flow diagram (continued).

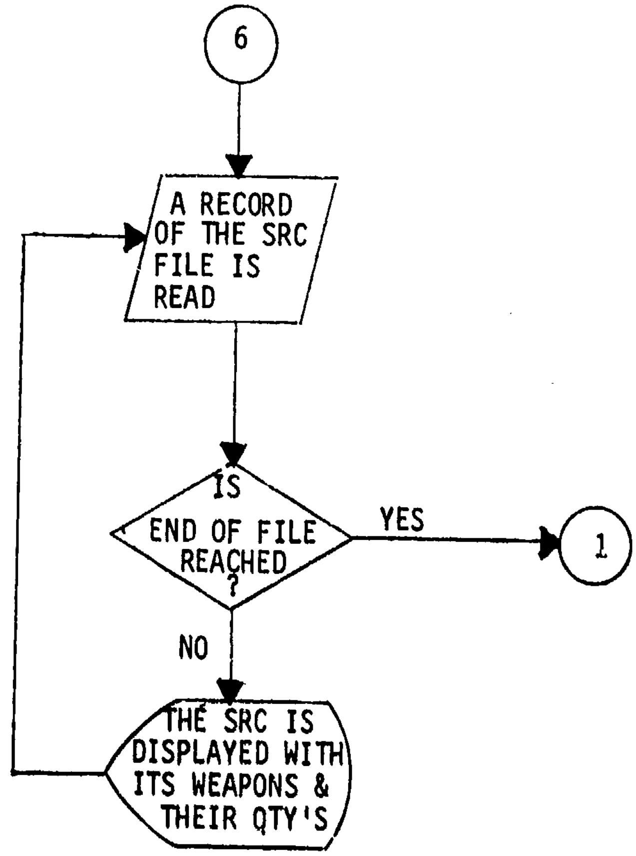

Figure 2. SRC program logic flow diagram (concluded).

more SRCs to a unit already on file, delete specified units, delete particular SRCs from specific units, and list all units with their SRCs. A logic flow diagram of the UNIT program is contained in figure 3. A listing of the program code and a list of the program variables is contained in appendix C to this volume.

(3) PARENT program. The PARENT program is the third part of the force structure generation process. The PARENT program is the tool with which the military gamers can task organize interactively the combat units previously defined on the UNIT file into a file of higher echelon organizations, or parent units. The parent units are created by the program with the definition of a unique parent unit name (1 to 10 alphanumeric characters) and the specification of up to 18 valid units within its organization. The format for the records of the PARENT file is illustrated in figure 1(c). In addition to creating the PARENT file, the program may be used to review the units of parent units already on file, add new parent organizations to the file, add new units to existing parent units, delete specified parent units, delete given units of specific parent units, and list all parent organizations with their subordinate units. A logic flow diagram of the PARENT program is presented in figure 4. A listing of the FORTRAN code and a list of the PARENT program variables are contained in appendix D to this volume.

(4) FORCE program. The FORCE program, the final step in the force structure process, interactively creates a file of the forces to be assessed in the combat routines of the Jiffy Game. The FORCE program consolidates the information defined on the files in the previous three steps of the process. The FORCE file consists of records for each unit of both forces. The format of the records of the FORCE file is presented in figure 1(d). The first 10 words of the record define the unit and its combat environment. Although some of these parameters (sector, critical incident, combat intensity) are redefined in the Jiffy Game during the actual gaming, the first 10 words are initialized in the FORCE program. The remaining 80 words (words 11 to 90 on the record) contain the quantity and indicate the type of weapon system in the unit. The position of the word denotes the type of weapon system (item code equals record word number minus 10). The value of the word is the quantity of that type of weapon system. Besides generating the FORCE file, the FORCE program provides the capabilities to add units of a specified new parent unit to the force file using the information stored in the other three files, delete all the units of a specific parent unit from the file, change the unit effectiveness of any unit on the file, and list all parent units with their subordinate units and their corresponding quantities of weapon systems. It should be noted that when a unit is added to the file, the gamer is asked to input its unit effectiveness, which is the percentage of a unit's existing firepower score compared to its 100 percent firepower score. The number of each type of weapon system loaded into a unit equals the number of that weapon allocated to the unit at 100 percent strength multiplied times the unit's effectiveness. For example, if a unit had 16 tanks at

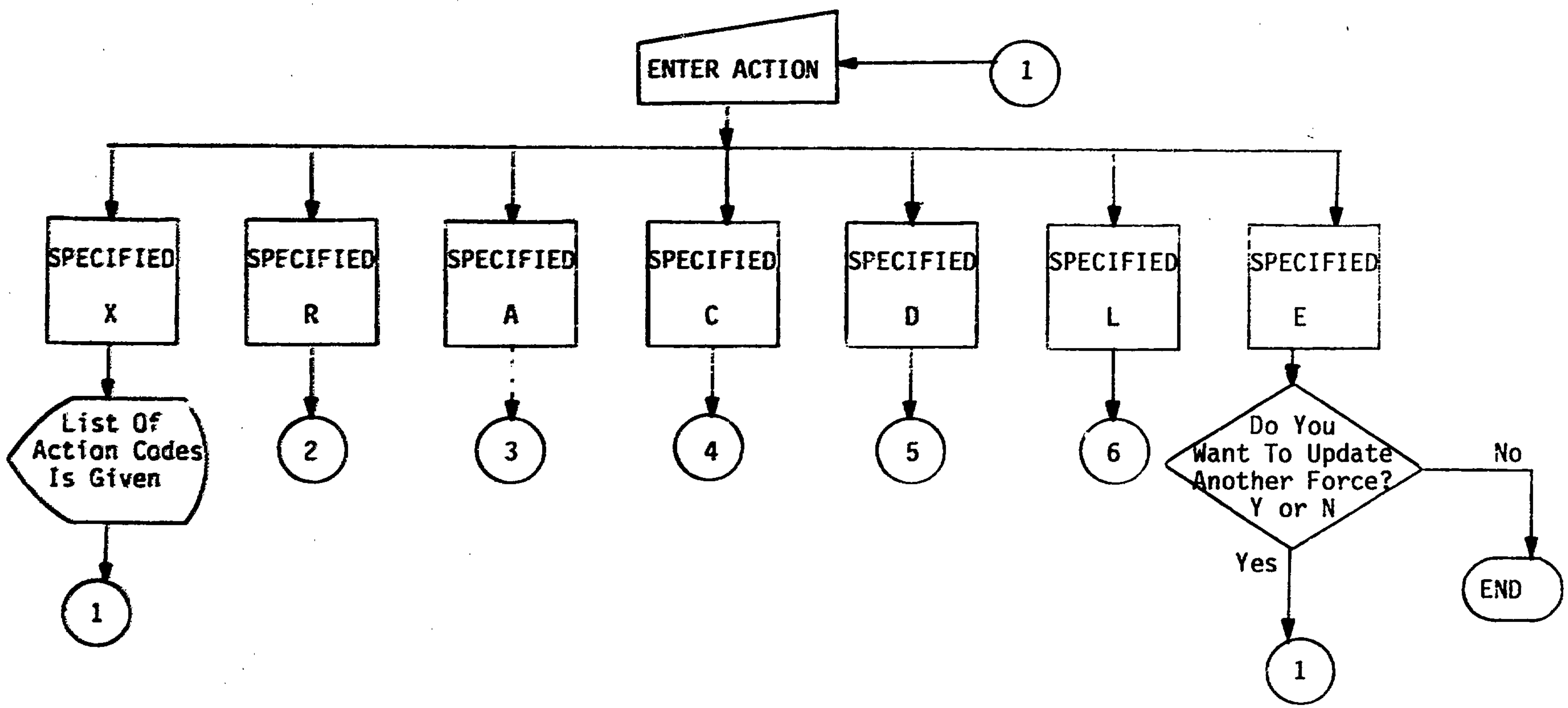

Figure 3. UNIT program logic flow diagram. (Continued next page)

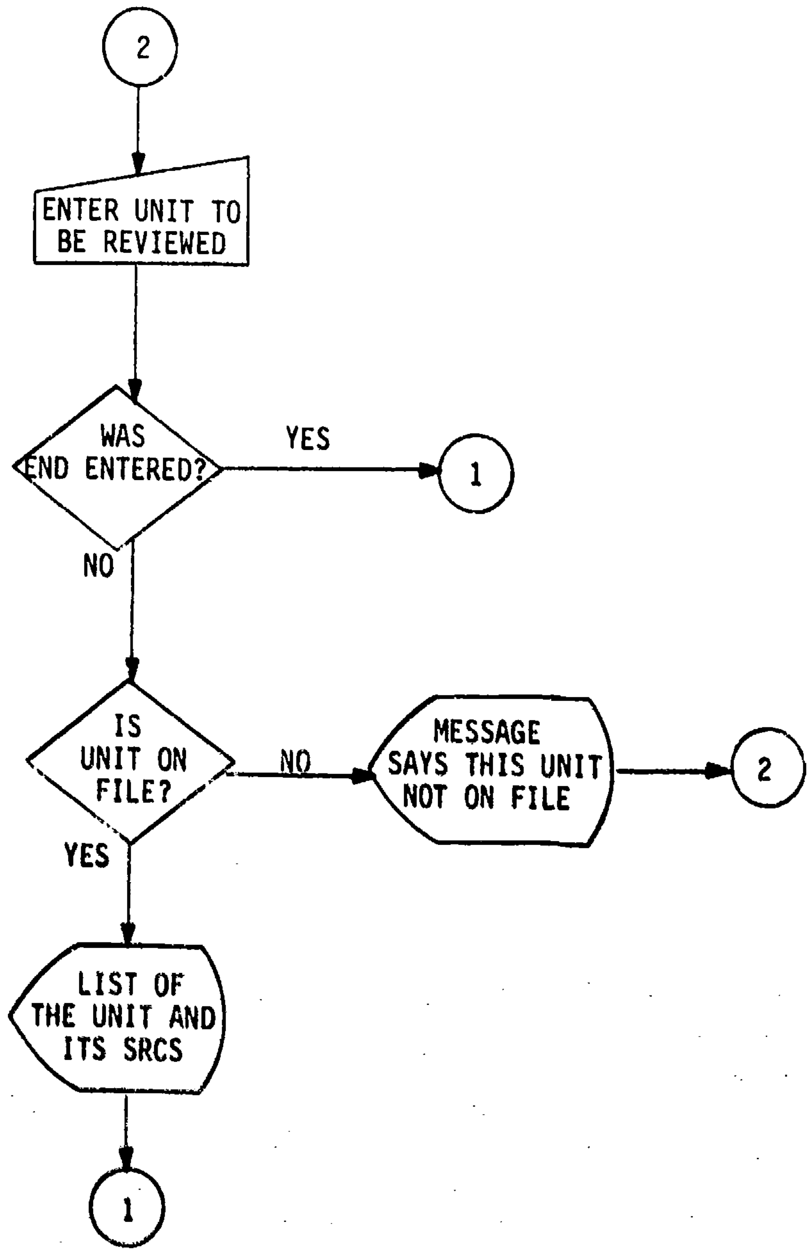

Figure 3. UNIT program logic flow diagram (continued).

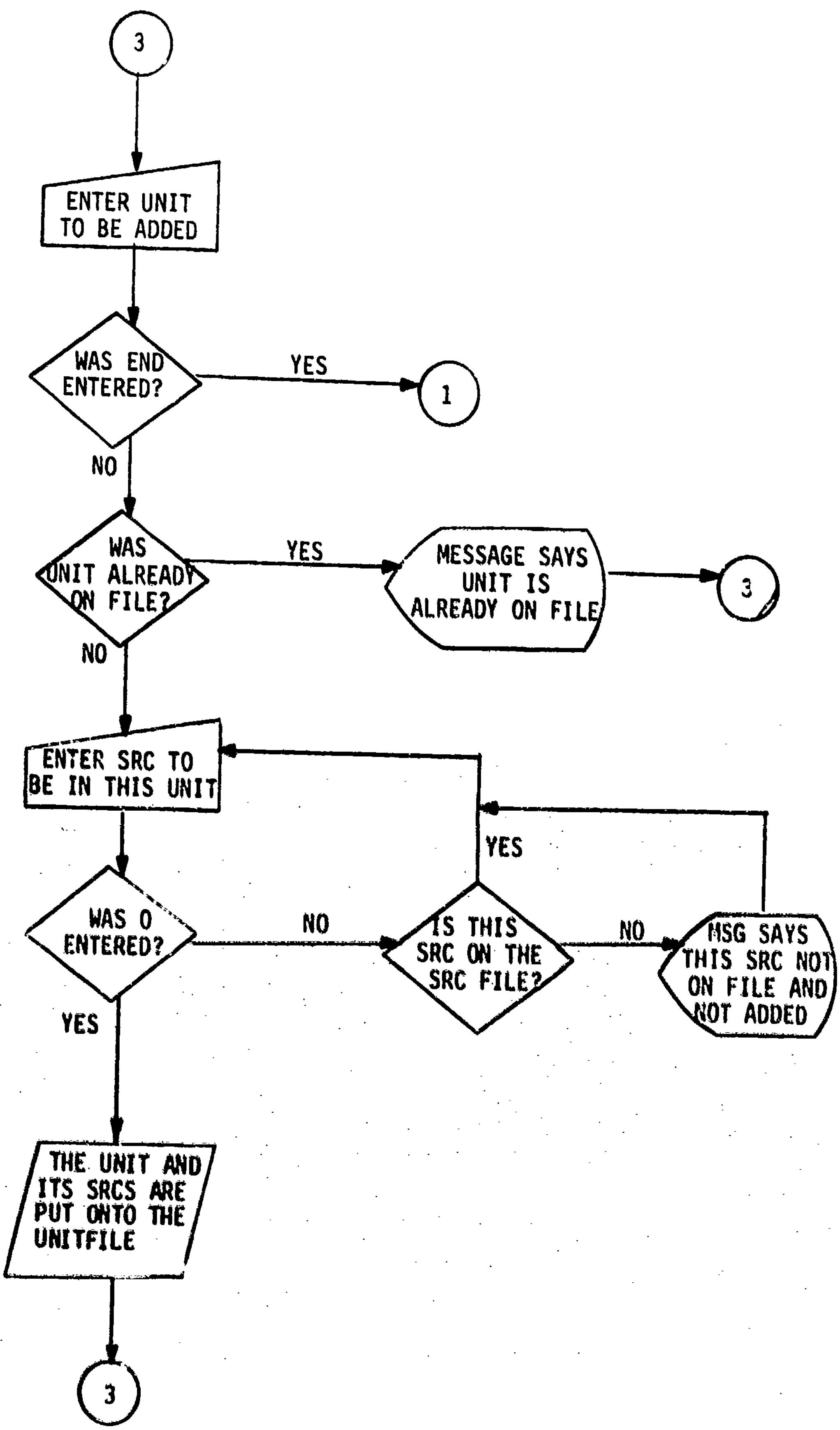

Figure 3. UNIT program logic flow diagram (continued).

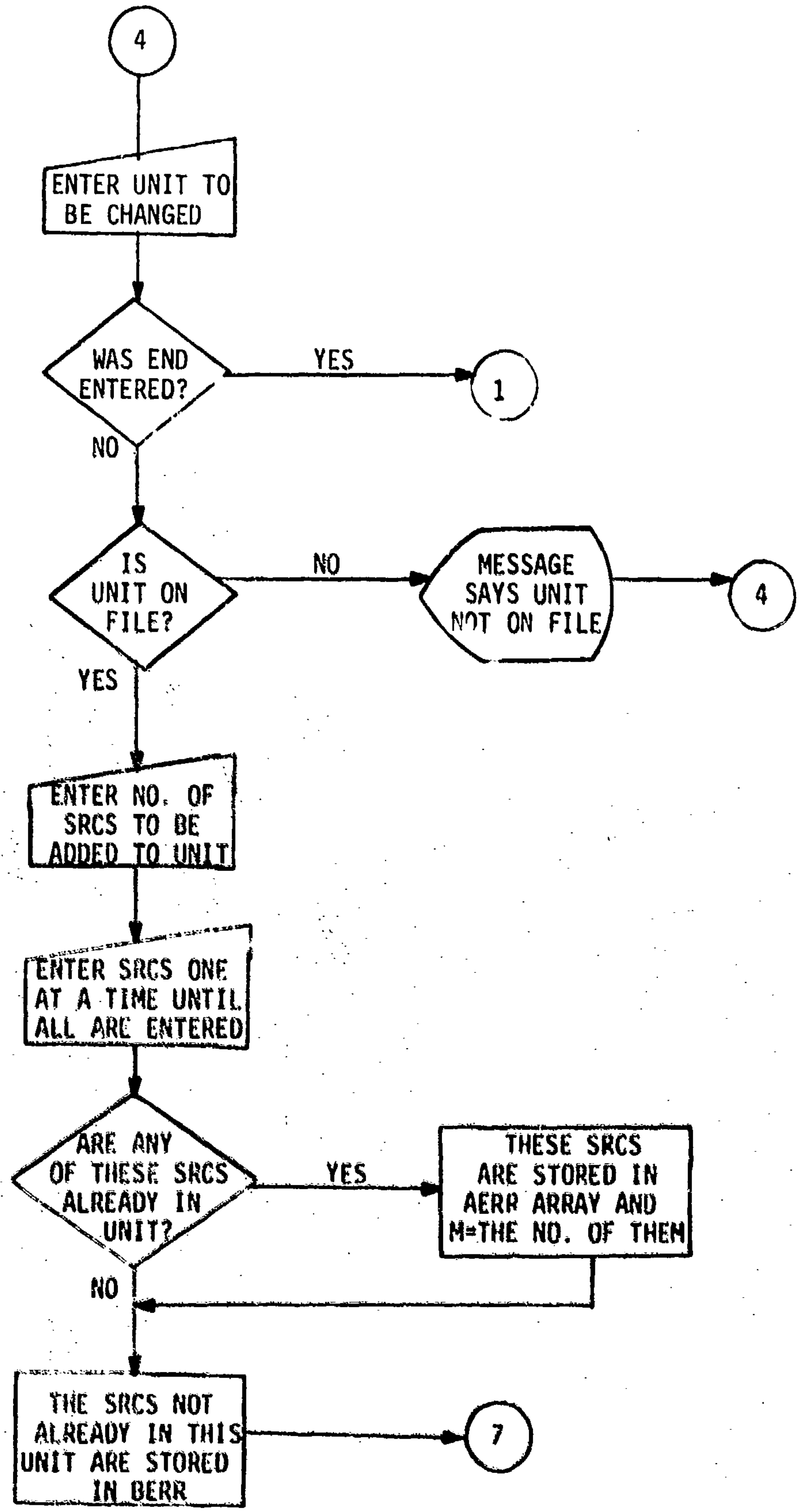

Figure 3. UNIT program logic flow diagram (continued).

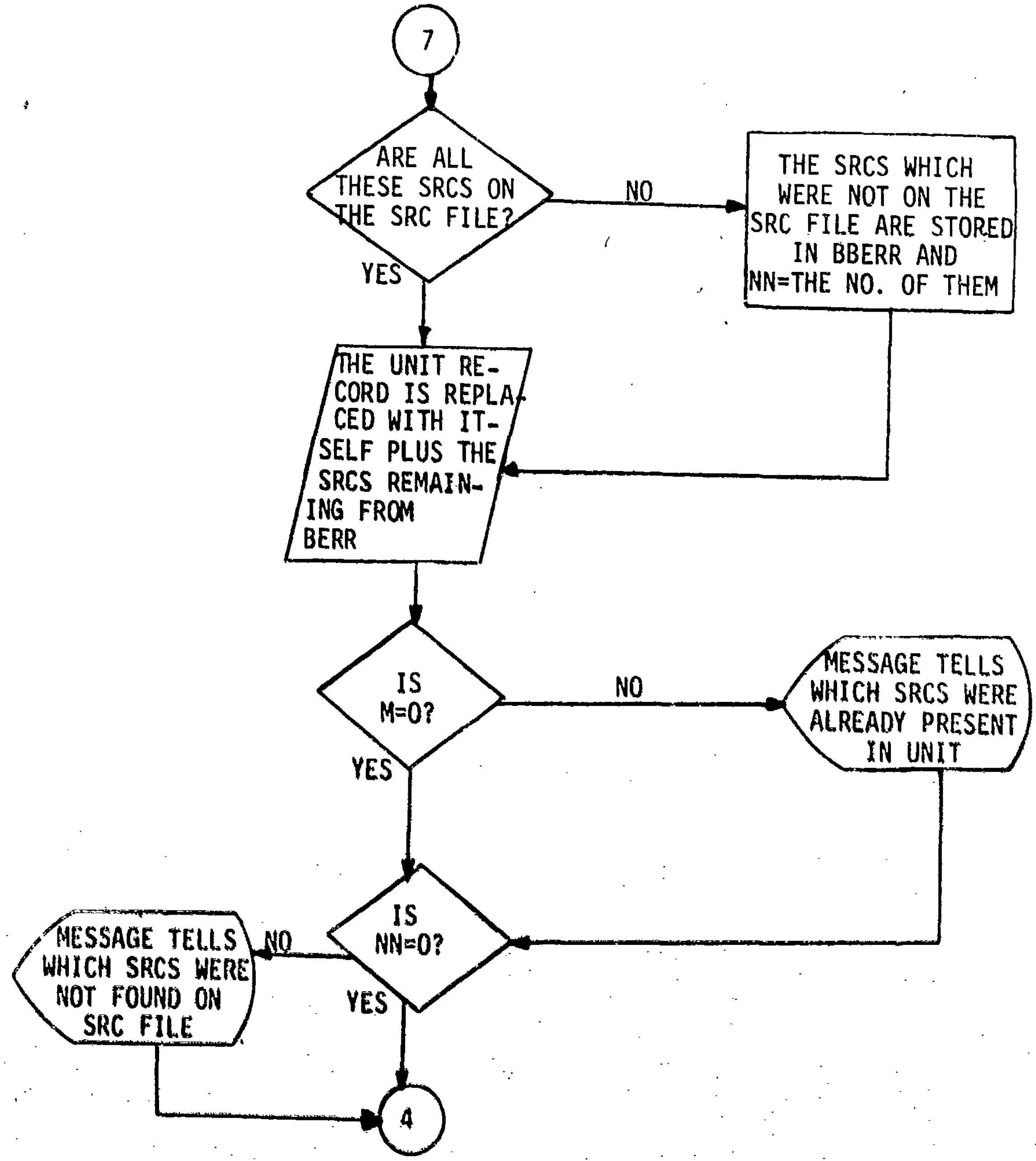

Figure 3. UNIT program logic flow diagram (continued).

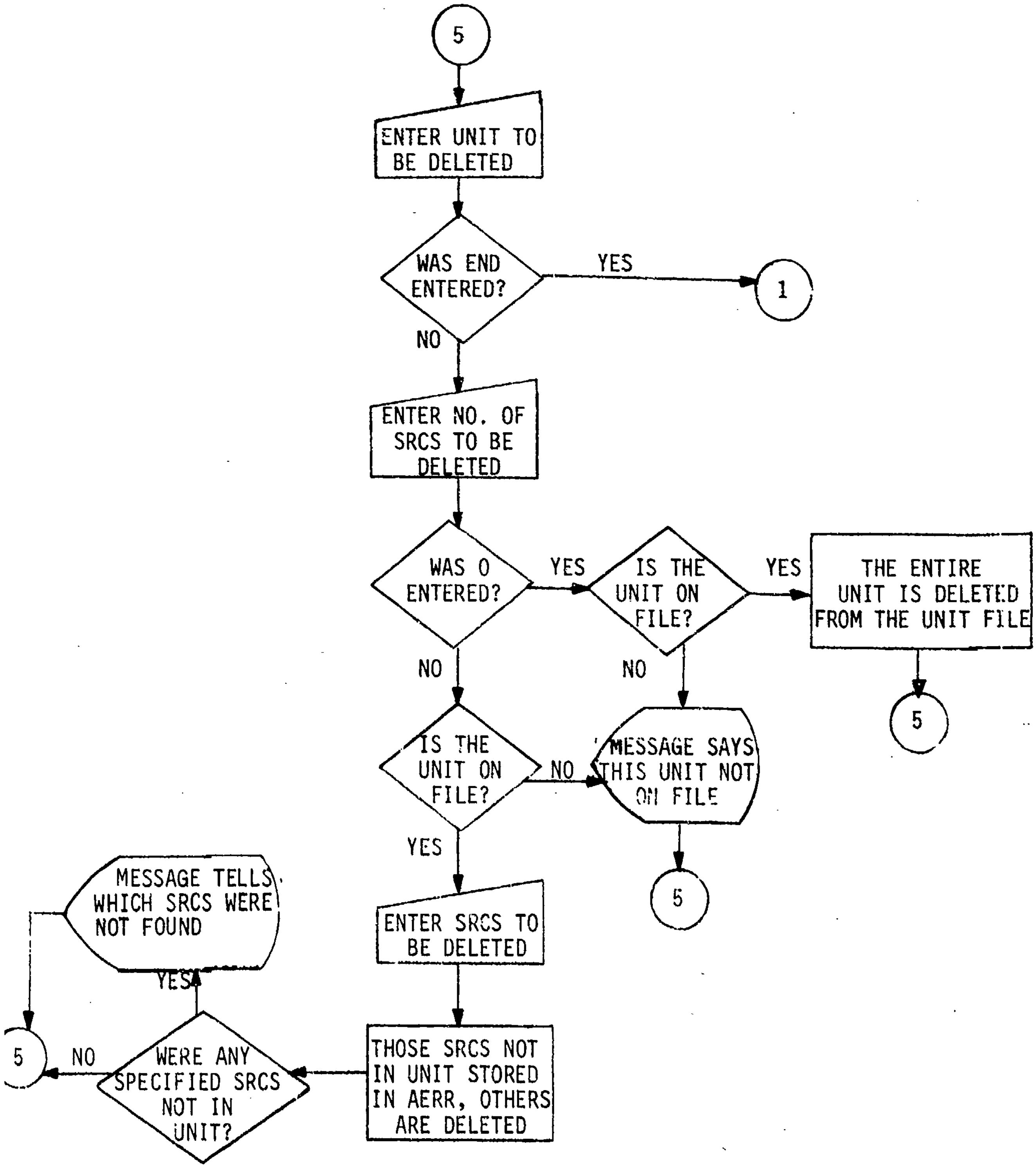

Figure 3. UNIT program logic flow diagram (continued).

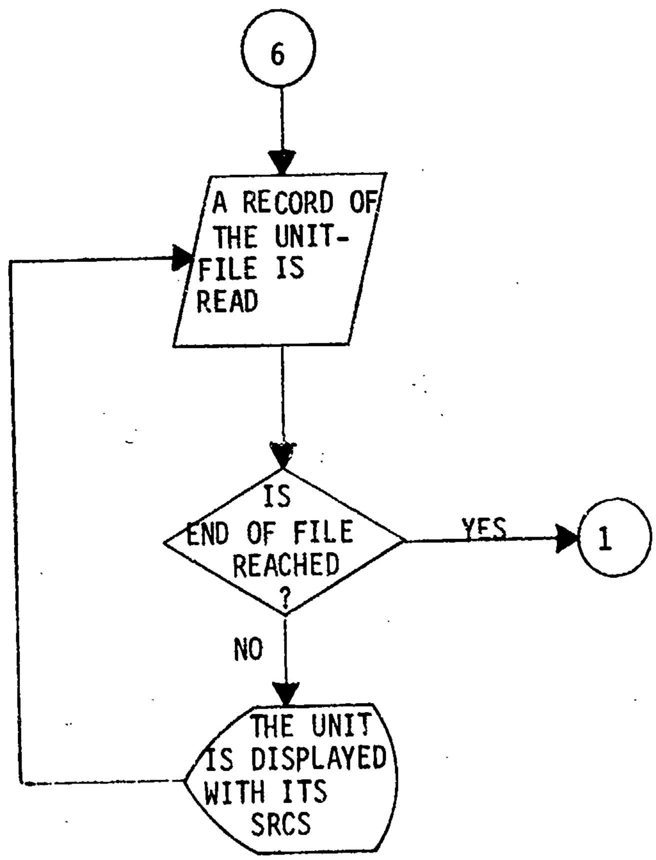

Figure 3. UNIT program logic flow diagram (concluded).

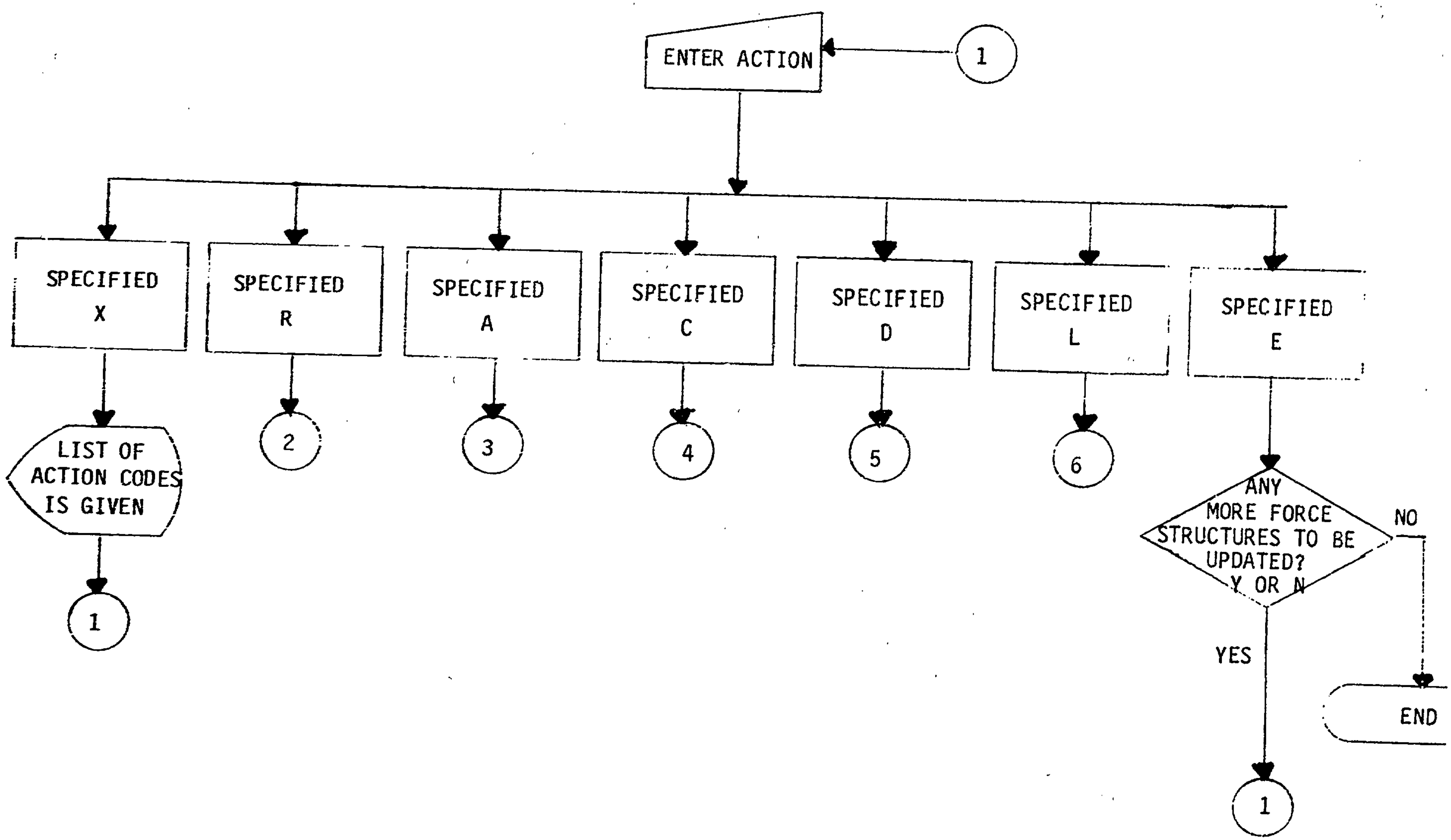

Figure 4. PARENT program logic flow diagram. (Continued next page).

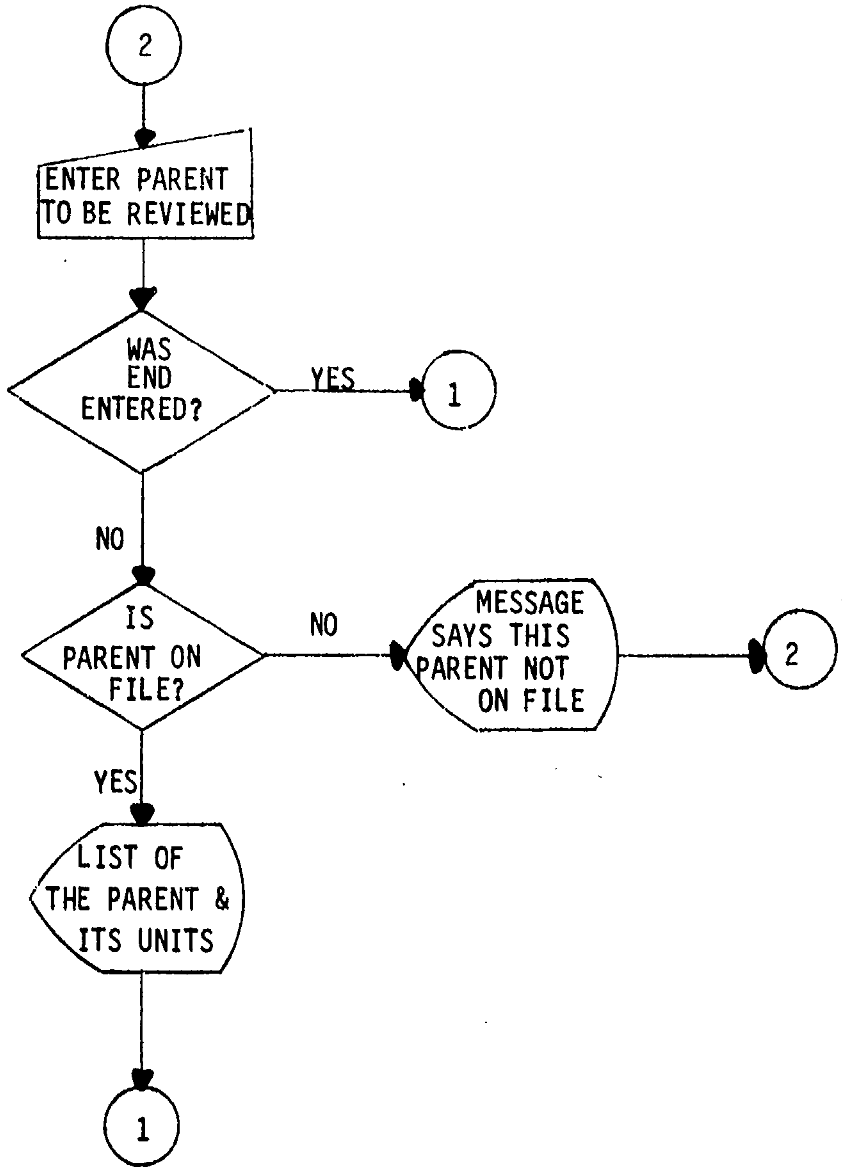

Figure 4. PARENT program logic flow diagram (continued).

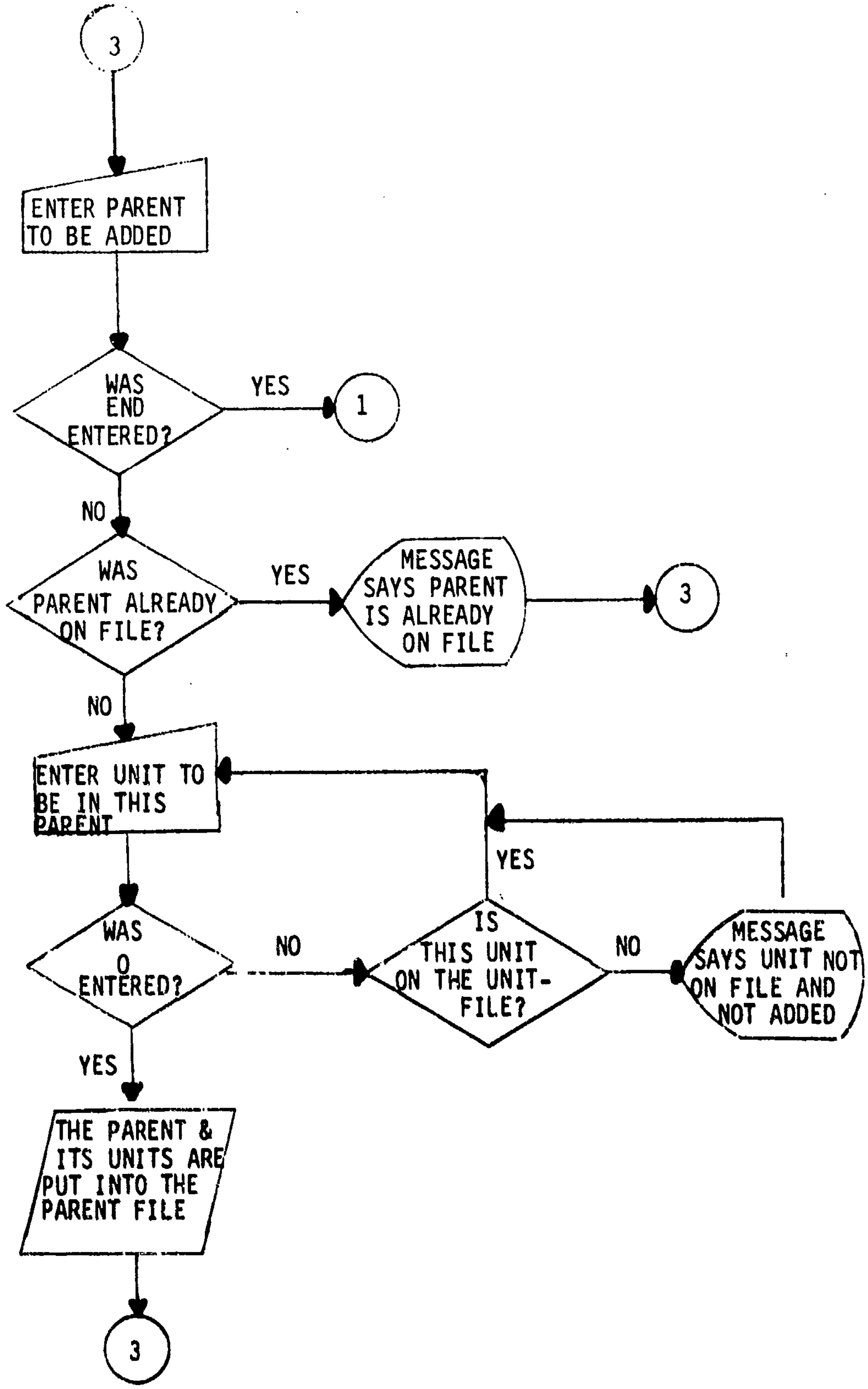

Figure 4. PARENT program logic flow diagram (continued).

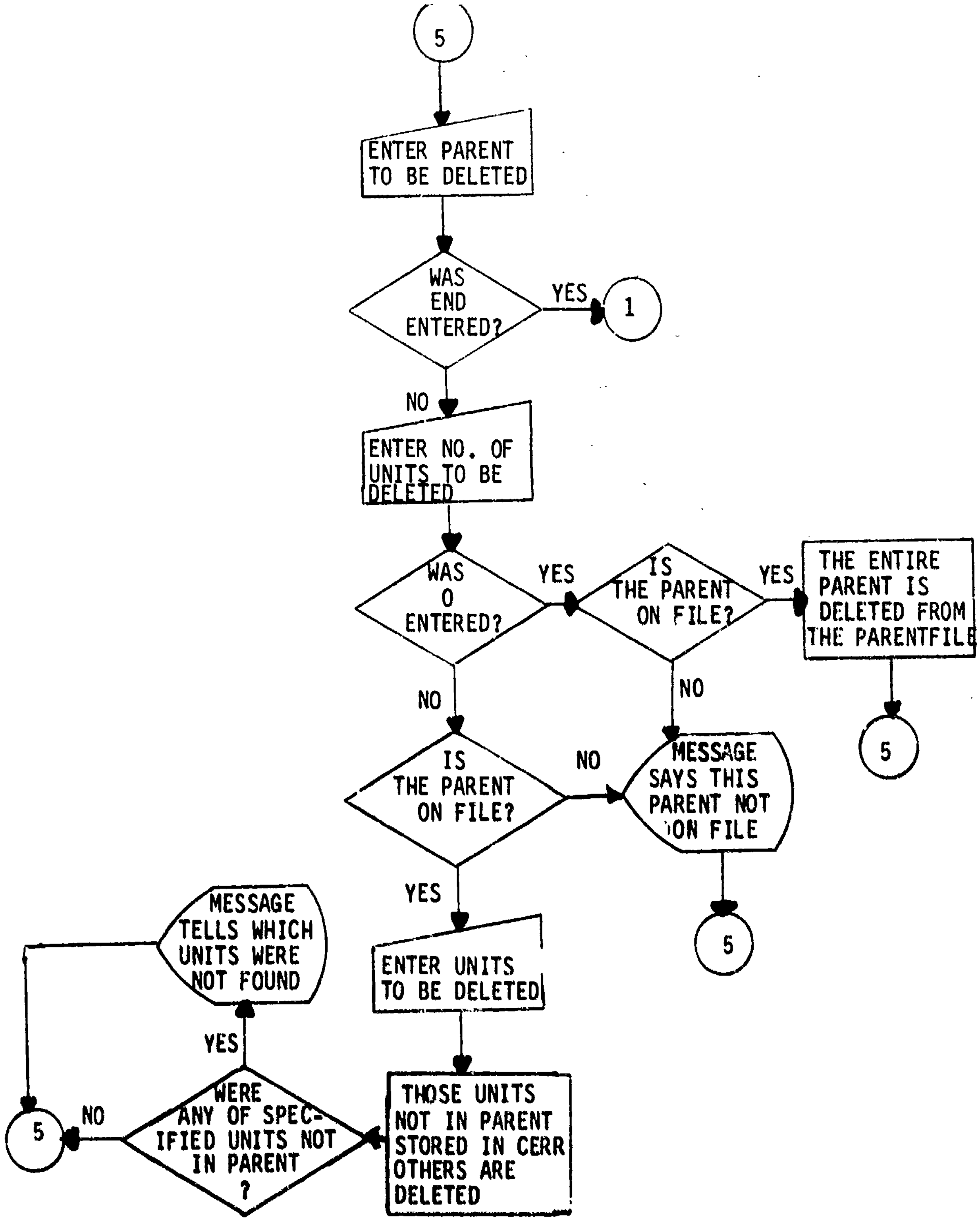

Figure 4. PARENT program logic flow diagram (continued).

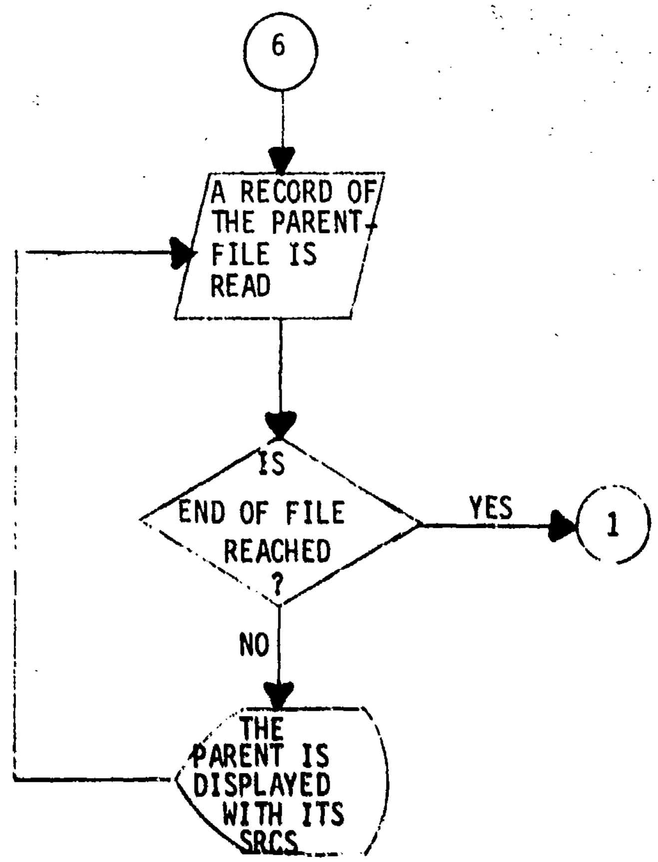

Figure 4. PARENT program logic flow diagram (continued).

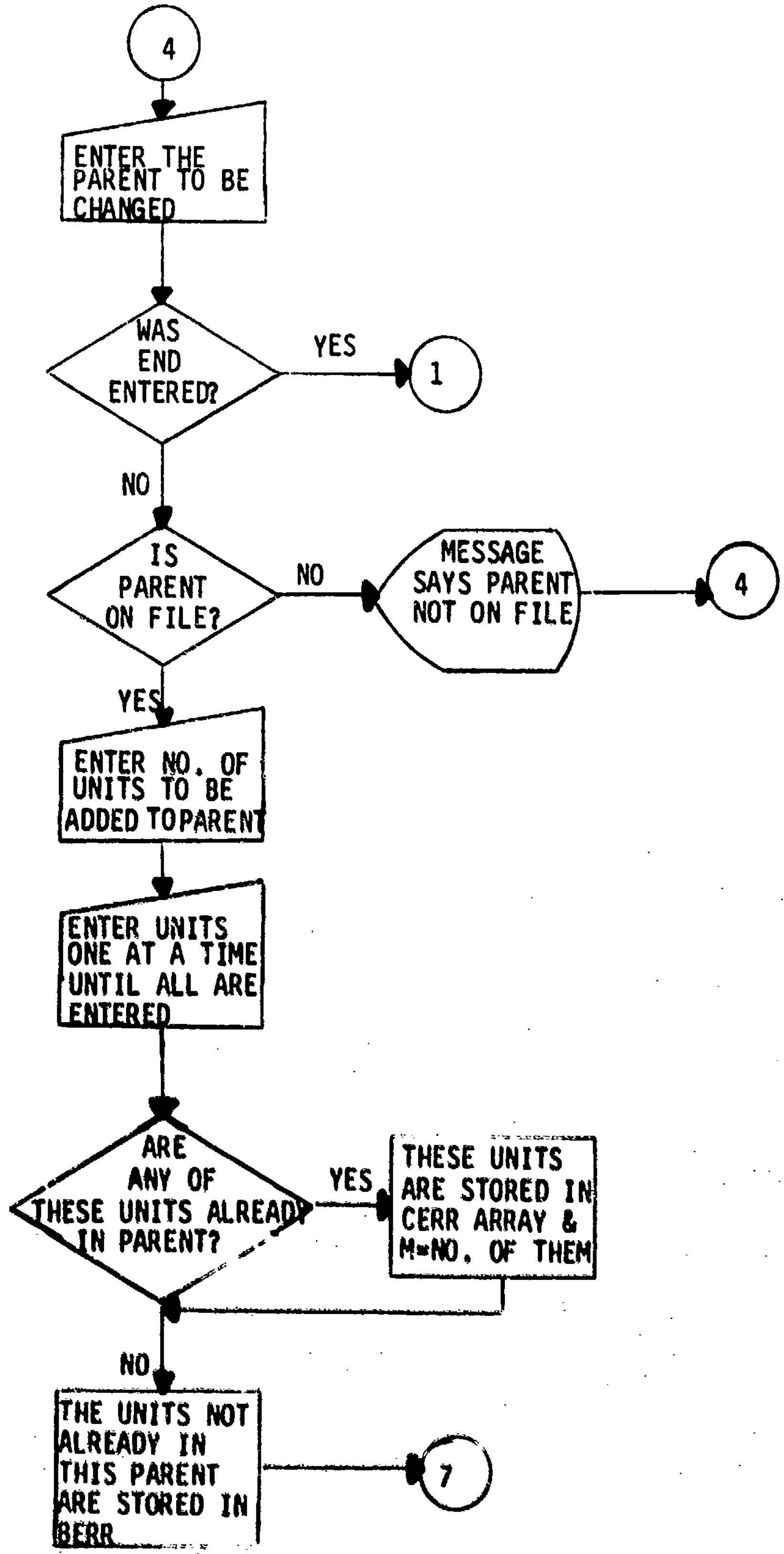

Figure 4. PARENT program logic flow diagram (continued).

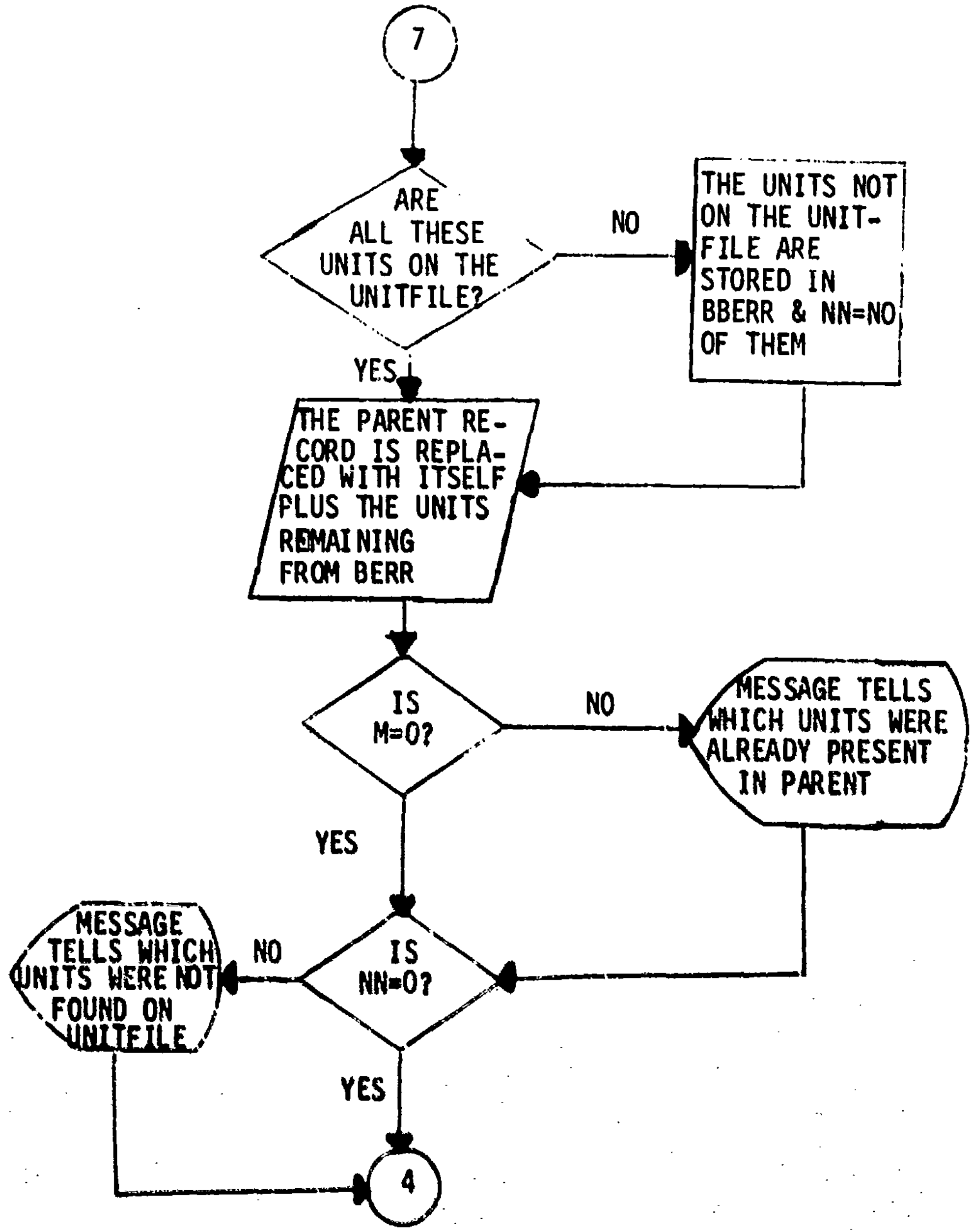

Figure 4. PARENT program logic flow diagram (concluded).

100 percent strength and it was specified to be at 50 percent unit effectiveness, only eight tanks would be loaded into the unit. A logic flow diagram of the FORCE program is contained in figure 5. A listing of the program code and a list of the program variables is contained in appendix E to this volume.

4. JIFFY GAME.

a. General. The Jiffy Game is a two-sided, interactive war game that operates on the FORCE file, the product of the force structure generation process, and determines the personnel casualties and weapon system losses incurred by the units of the two forces on the FORCE file as a result of the five types of combat it plays: indirect fire, minefields, armor/antiarmor, dismounted infantry, and attack helicopter/air defense. In addition to assessing combat, the Jiffy Game handles other administrative functions associated with the war game, such as combat loss apportionment, maintaining the FORCE file, updating the HISTORY file as required, and outputting the statistics of the battles. The Jiffy Game is written in FORTRAN and has utilized some of the features of CDC Extended FORTRAN. The program has been overlayed to fit into 100k words of core on the CDC 6500 for interactive processing. The CPU processing time under the scope 4.2 operating system varies with the size of the forces being gamed, but typical times vary between 10 and 60 CPU seconds per sector of combat gamed.

b. Program Descriptions. A functional flow diagram of the Jiffy Game is presented in figure 6. The following paragraphs describe each overlay and subroutine of the Jiffy Game, discuss the functions performed by the routines, and present their logic flow diagrams, FORTRAN source code listings, and lists of program variables.

(1) OVERLAY 0. The zero level overlay (OVLY0) contains the main program of the Jiffy Game (SUPER) and a few small subroutines, which are accessed by many of the other overlays. These include INIT, INDEX5, LOSS, and DISPLAY. The source code FORTRAN listings and lists of the program variables of the routines in OVLY0 are contained in appendix F to this volume.

(a) SUPER. The primary function of the main program is to serve as a control point from which a gamer can branch to the other overlays. During execution the gamer resides at a control point known as the DECISION POINT. At this point, the gamer has a choice of the nine decisions presented in table 1. Each gamer decision causes SUPER to branch according to the flow diagram of figure 7 and return to the DECISION POINT (block 6), except decision 9. In addition, SUPER performs the following functions:

1. calls INIT for data and array initialization (block 1),

2. displays the game instructions, if requested (blocks 3 and 4),

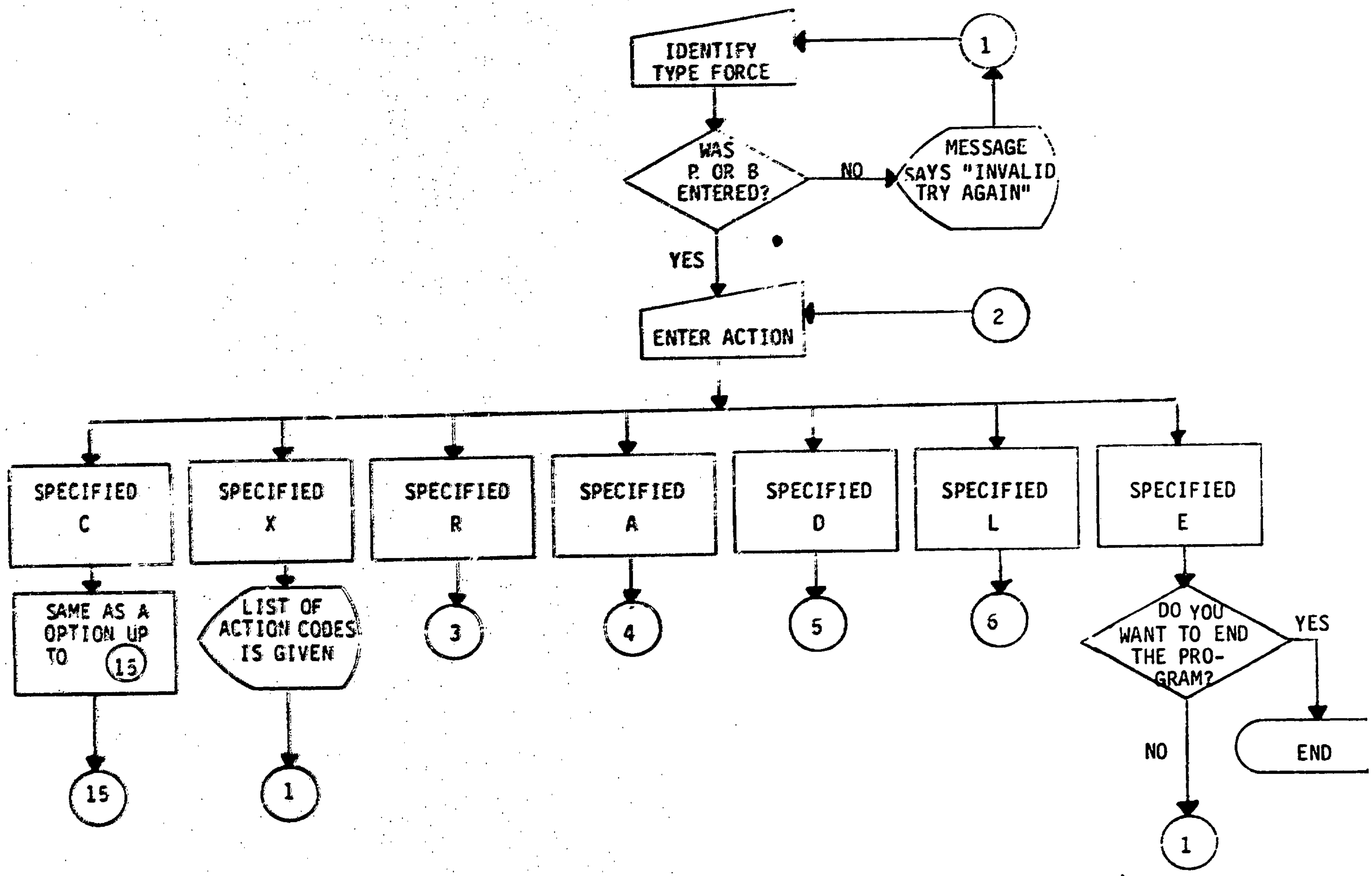

Figure 5. FORCE program logic flow diagram. (Continued next page)

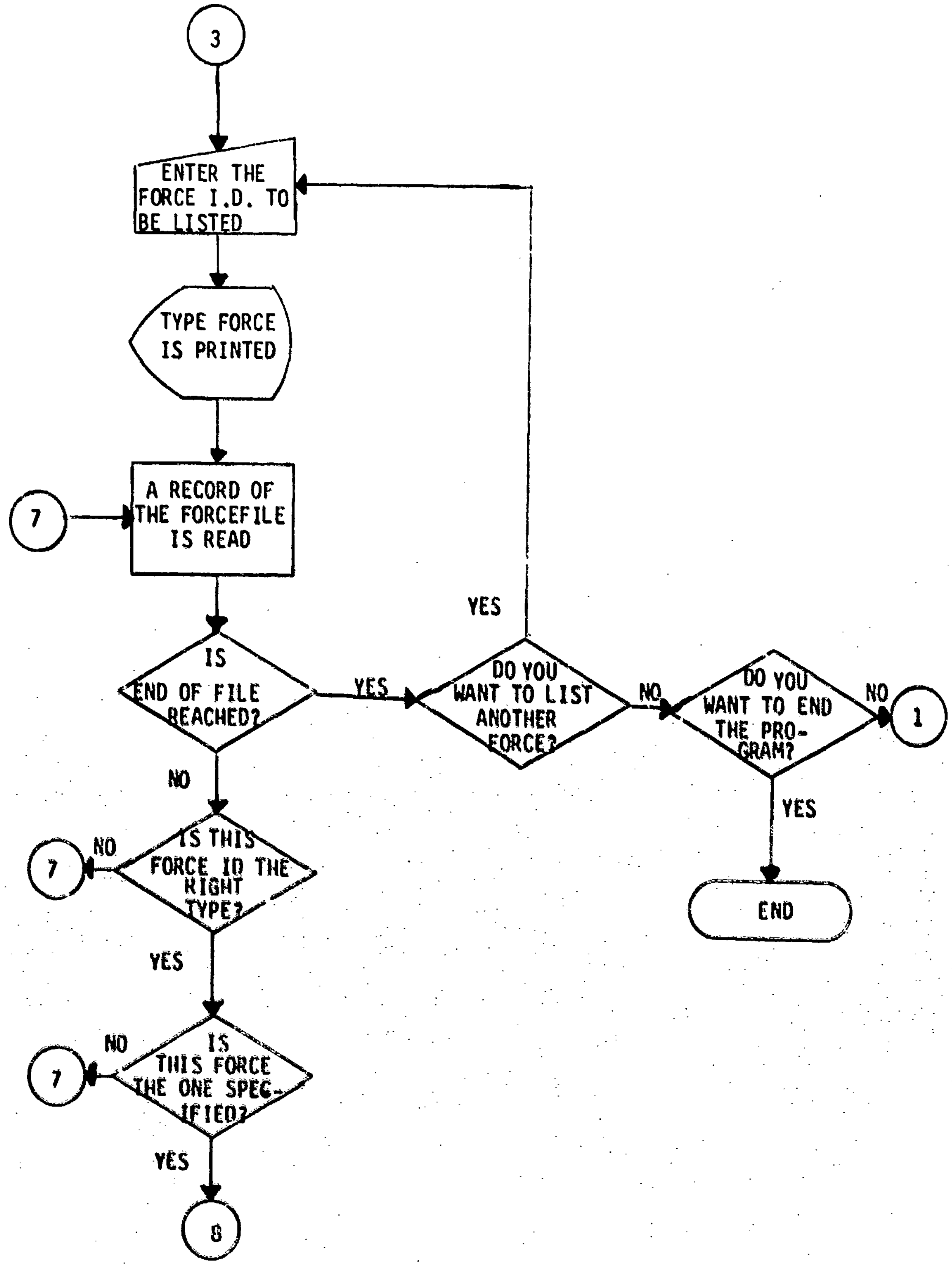

Figure 5. FORCE program logic flow diagram (Continued).

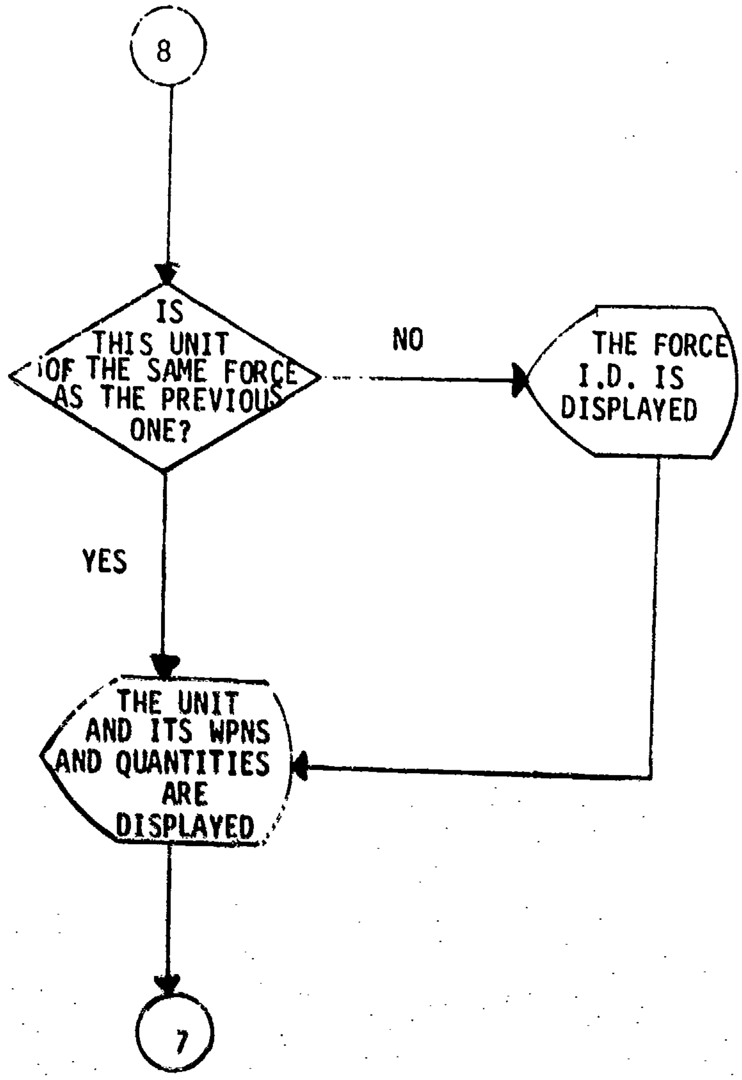

Figure 5. FORCE program logic flow diagram (continued).

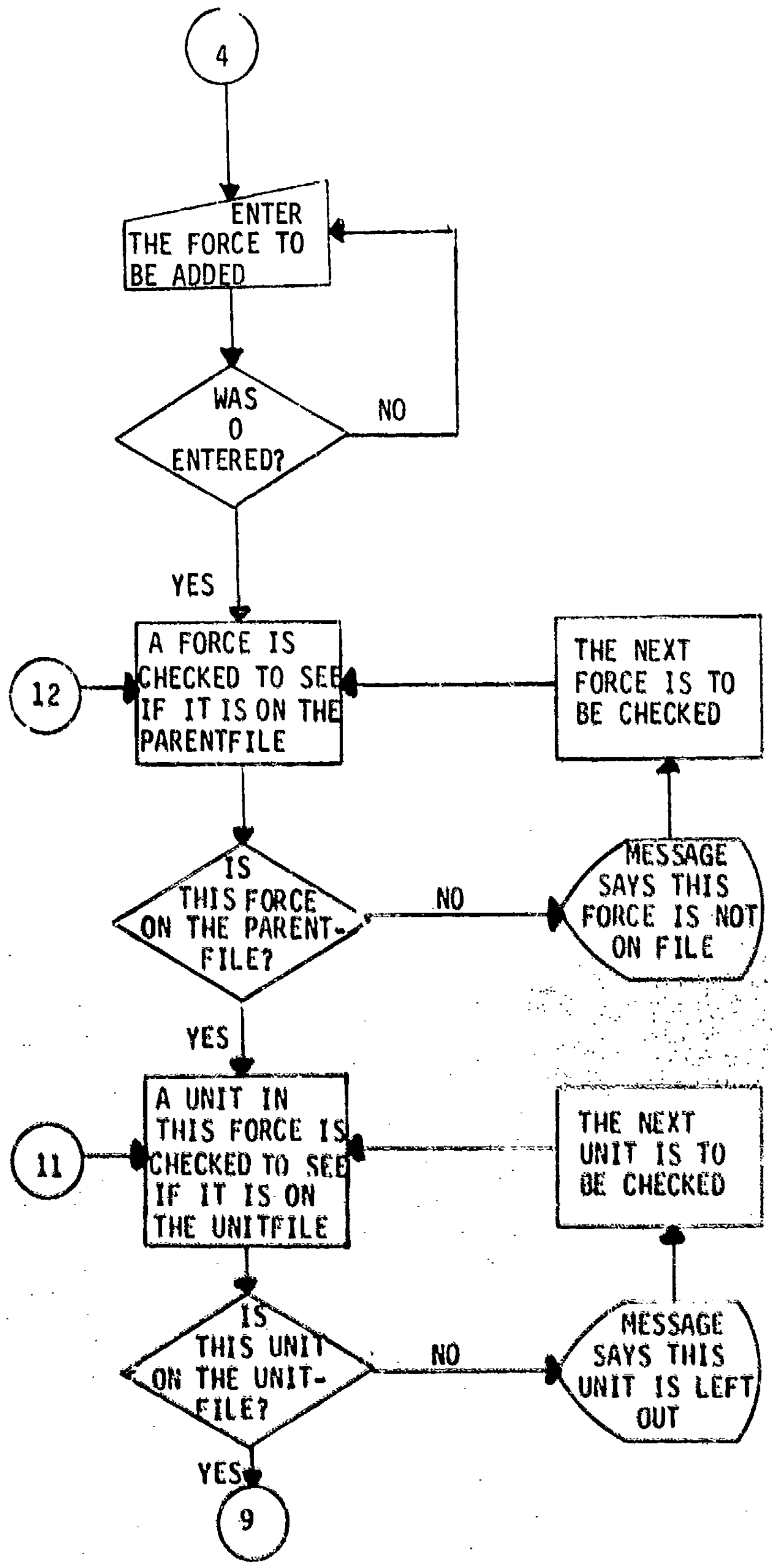

Figure 5. FORCE program logic flow diagram (continued).

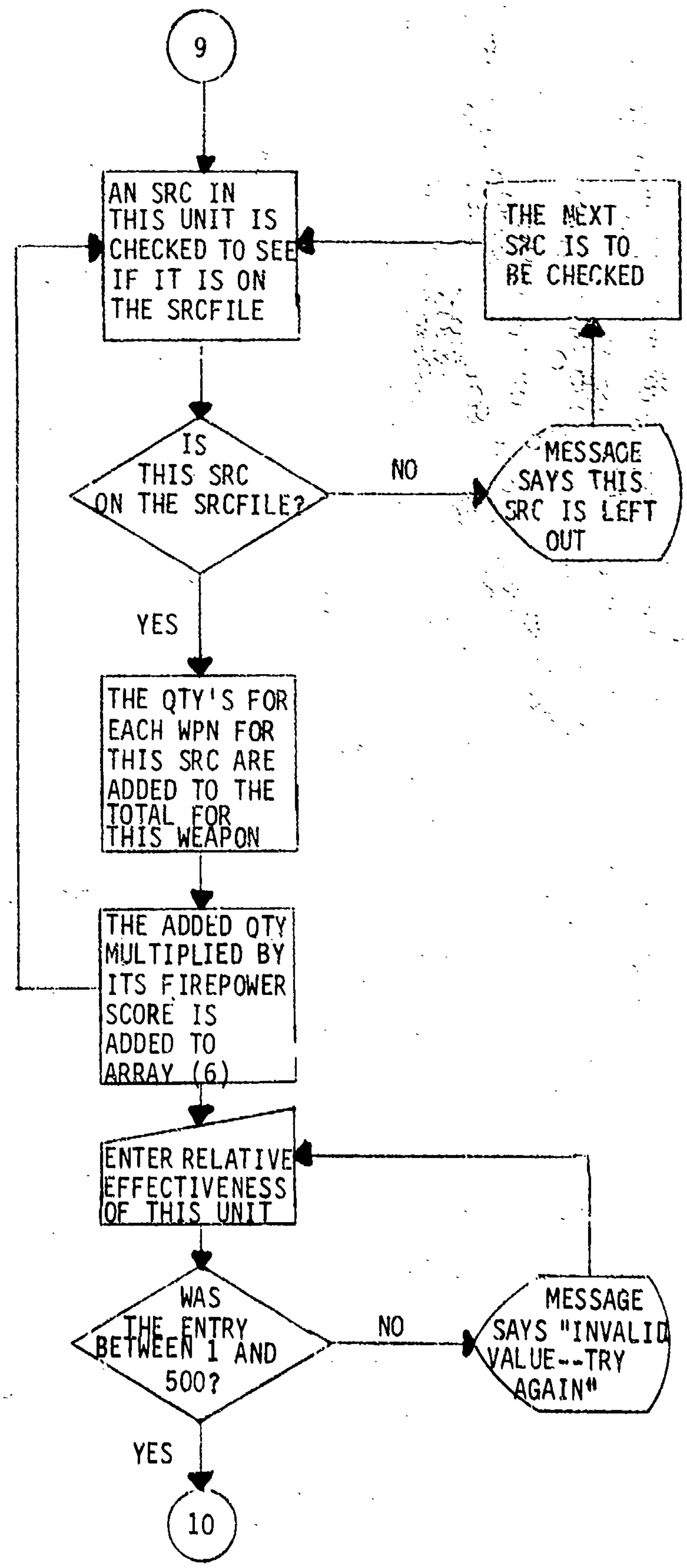

Figure 5. FORCE program logic flow diagram (continued).

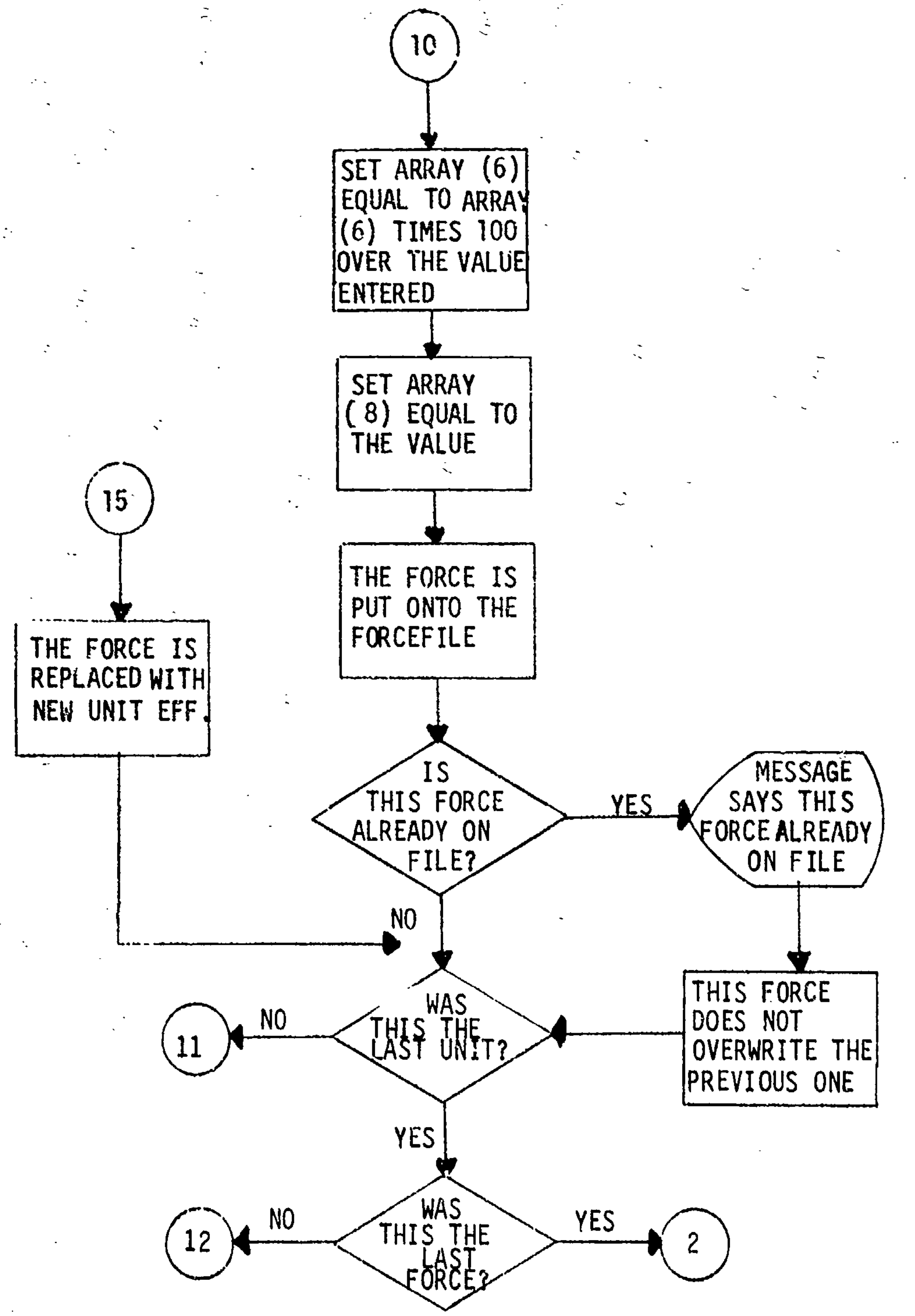

Figure 5. FORCE program logic flow diagram (continued).

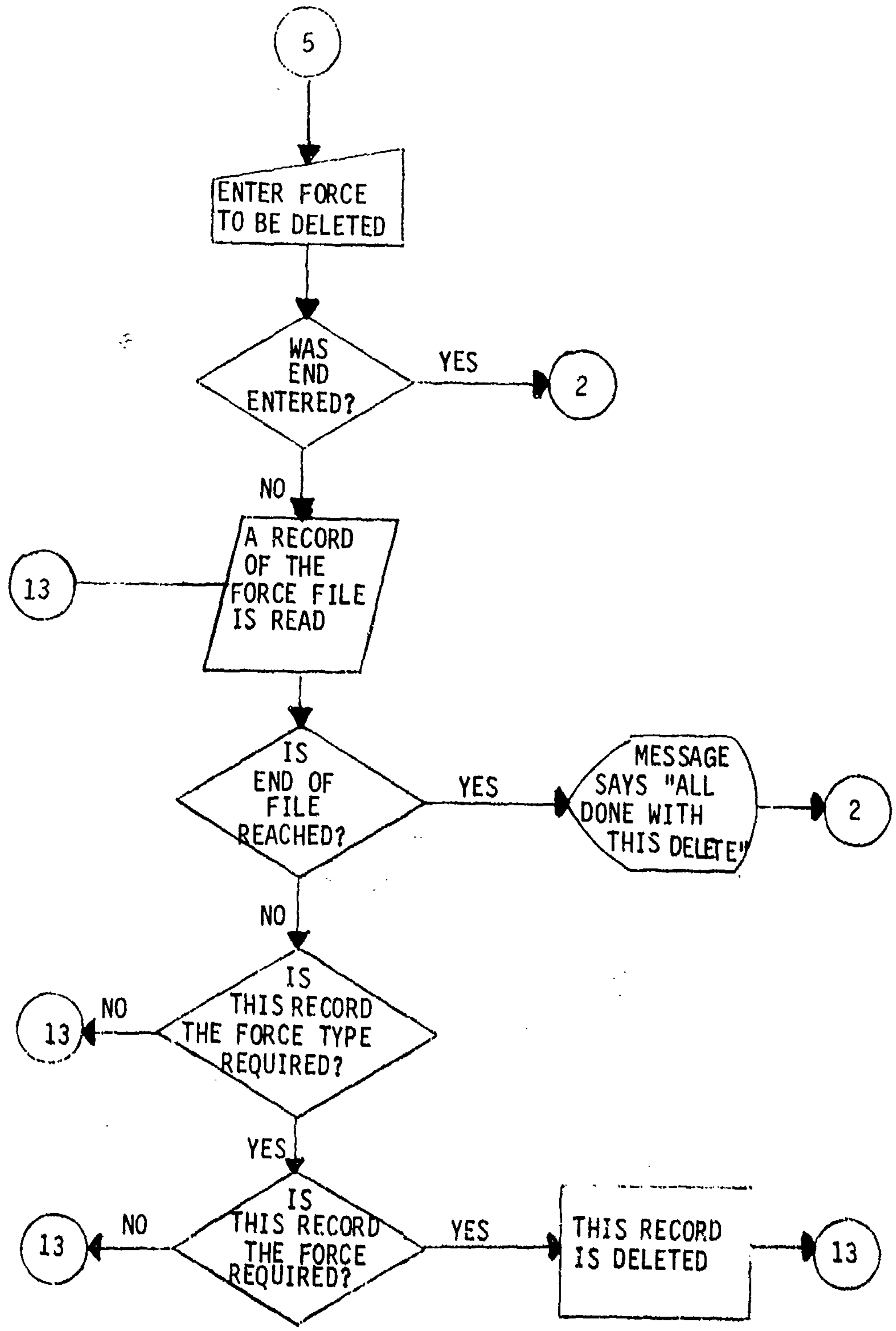

Figure 5. FORCE program logic flow diagram (continued).

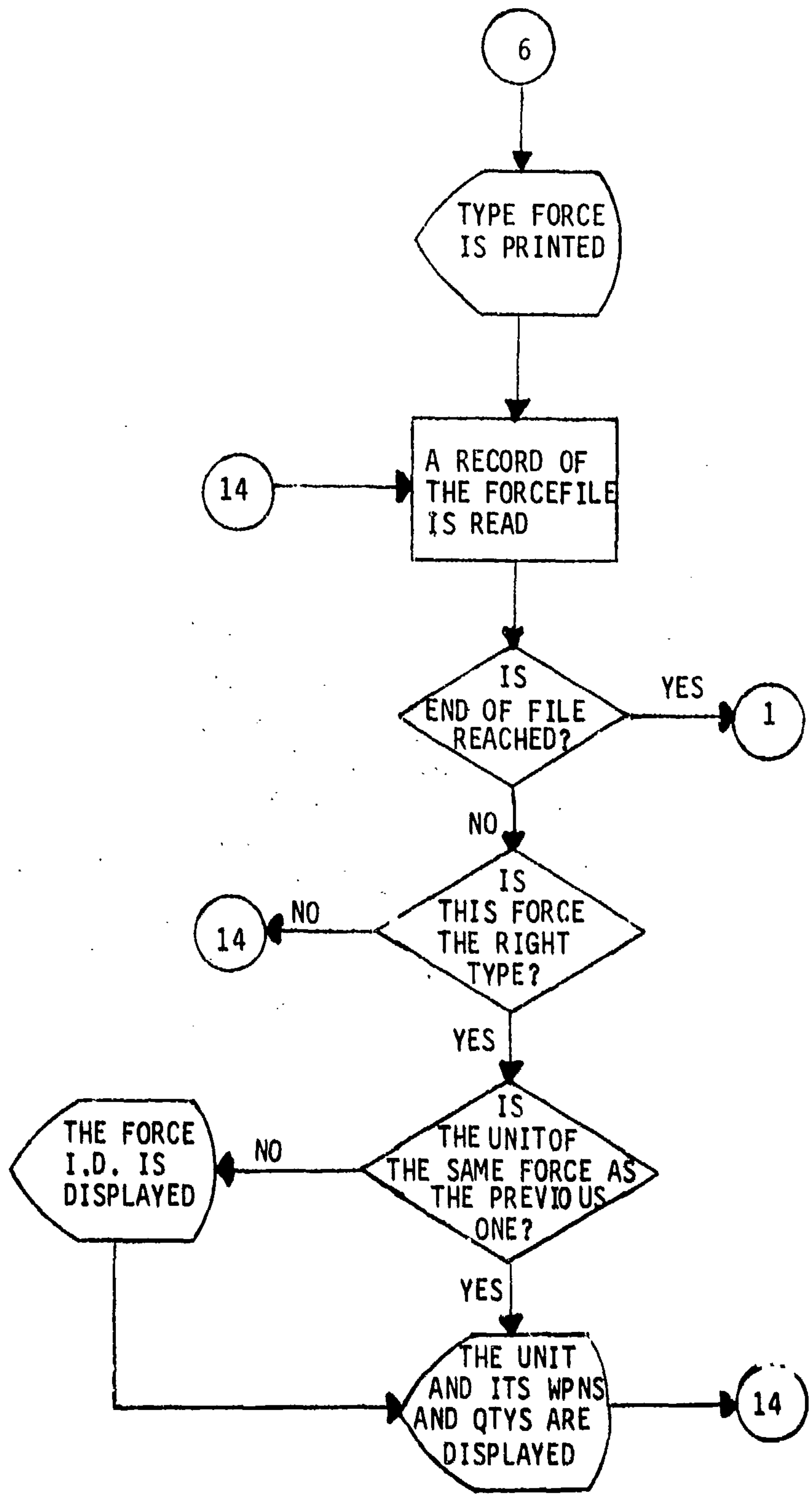

Figure 5. FORCE program logic flow diagram (concluded).

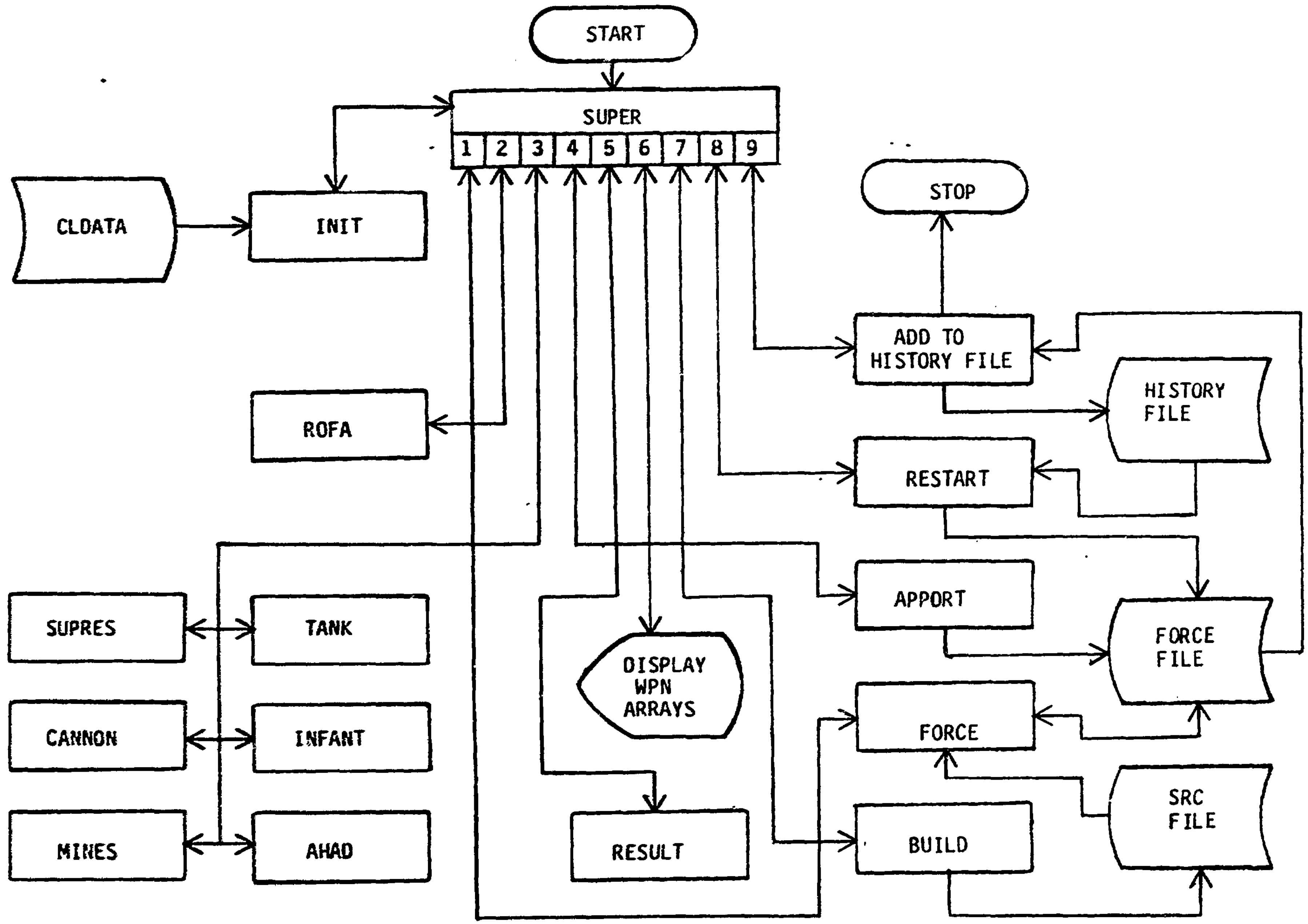

Figure 6. Jiffy game functional flow diagram.

Table 1. Control point gamer decisions.

Number	Description of Decision
1	Load forces into a sector
2	Calculate rate of advance
3	Assess combat
4	Apportion combat losses to units
5	Output battle statistics
6	Display weapon arrays
7	Add SRCs to TOE file
8	Restart at a previously gamed CI
9	Update history file - end game

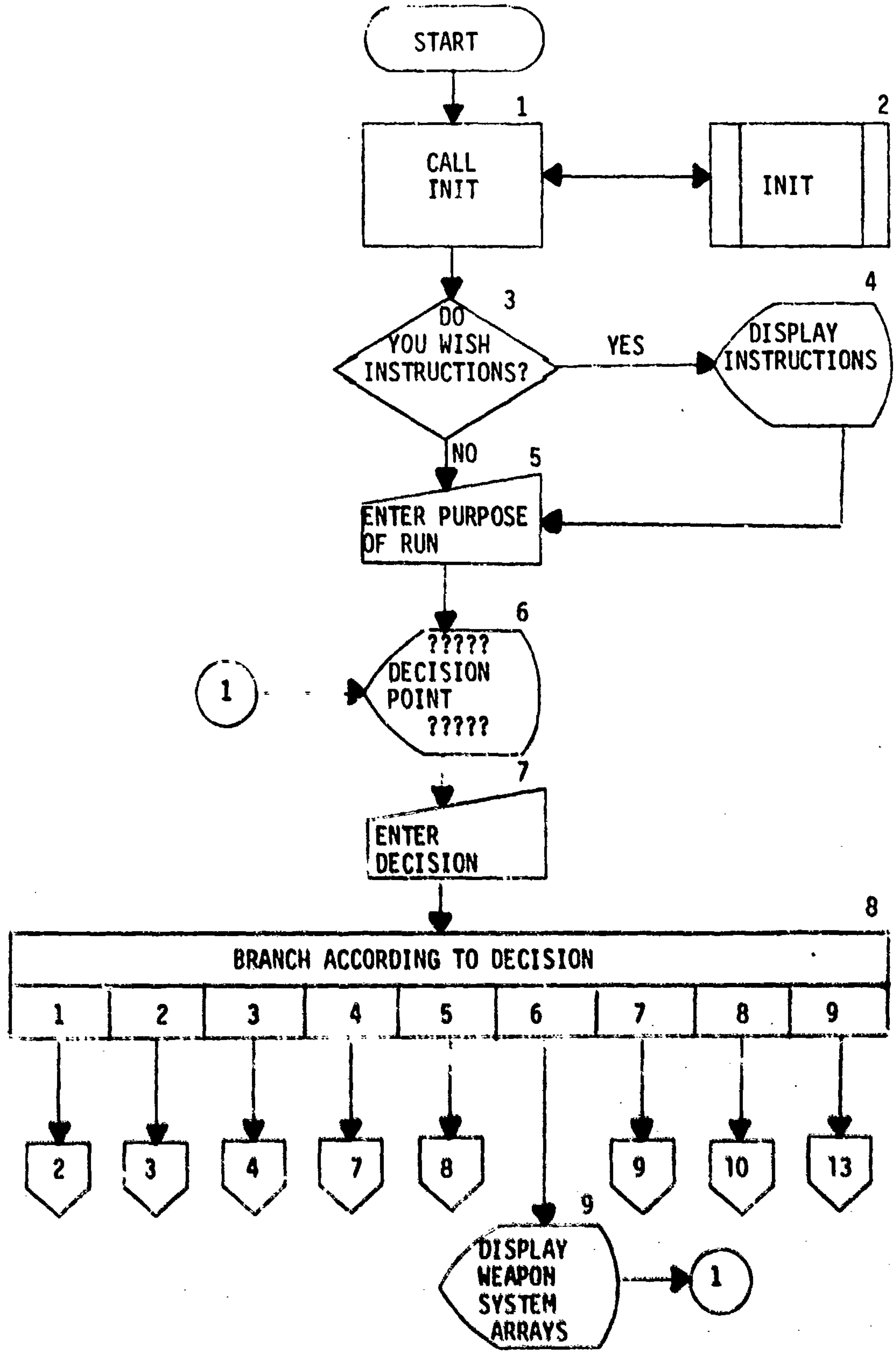

Figure 7. SUPER flow diagram.
(continued next page)

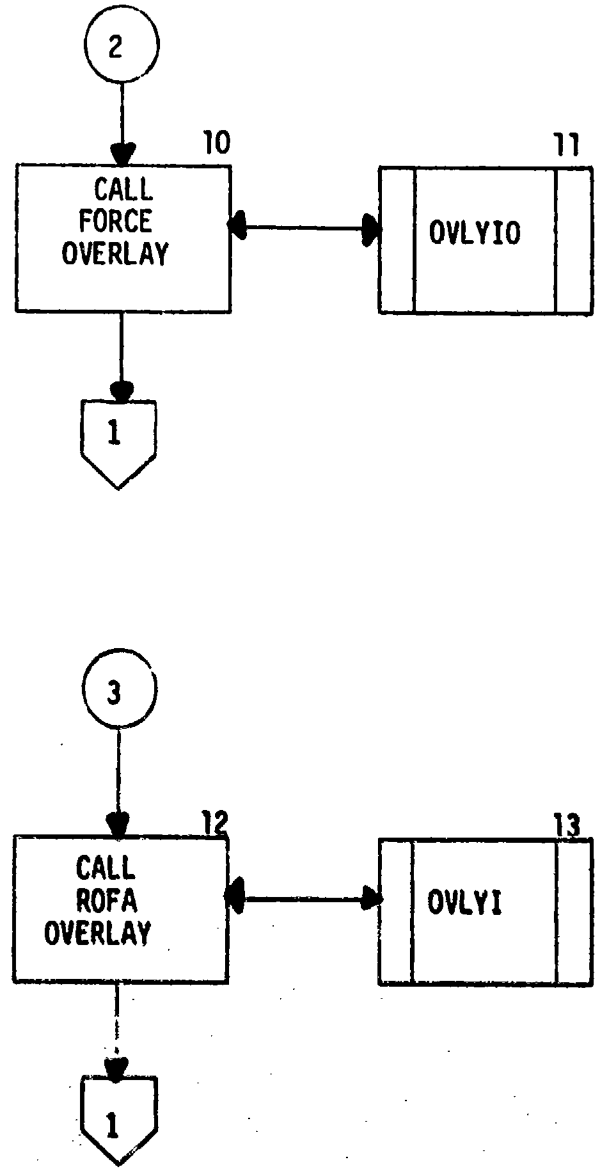

Figure 7. SUPER flow diagram (continued).

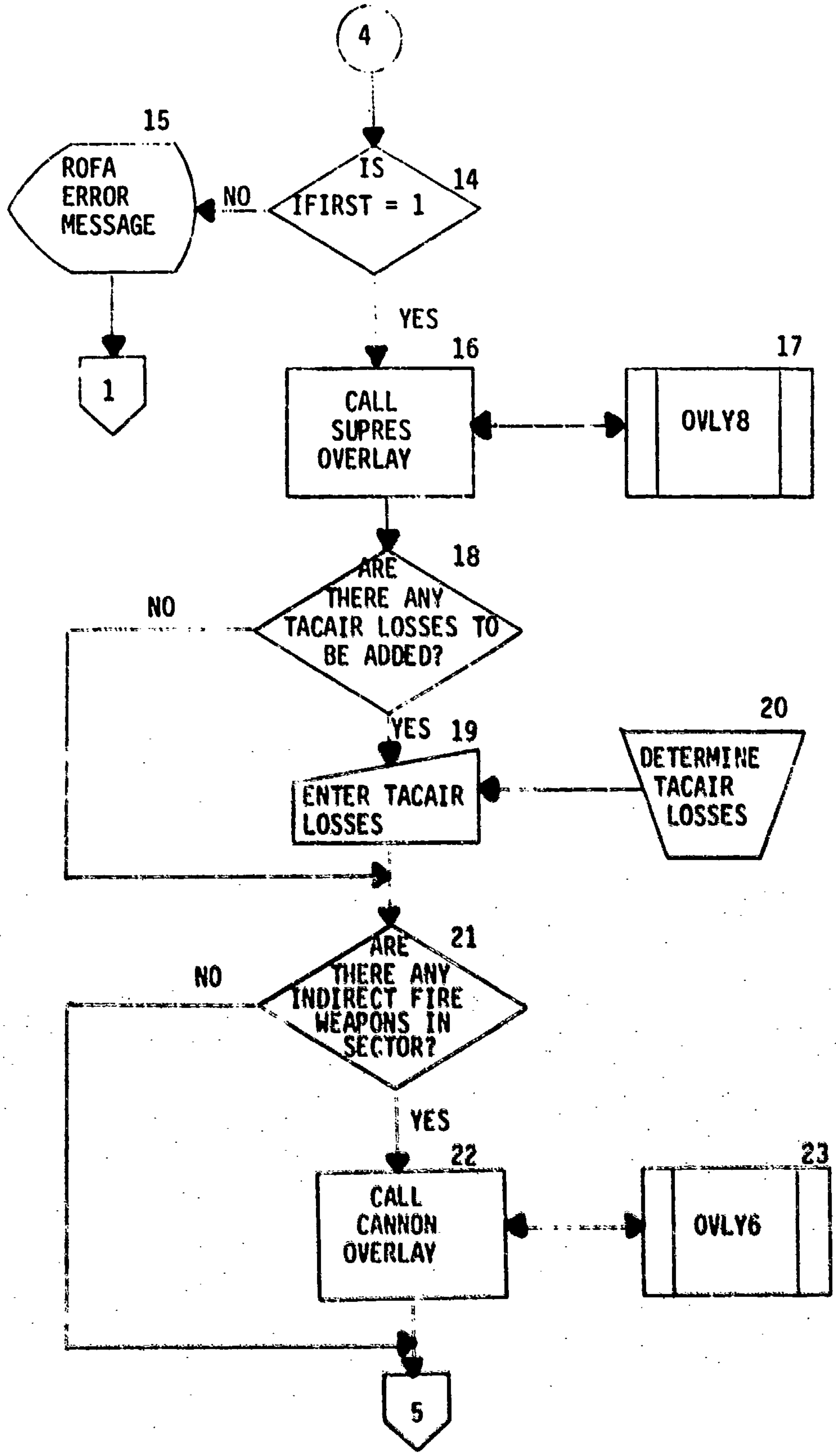

Figure 7. SUPER flow diagram (continued).

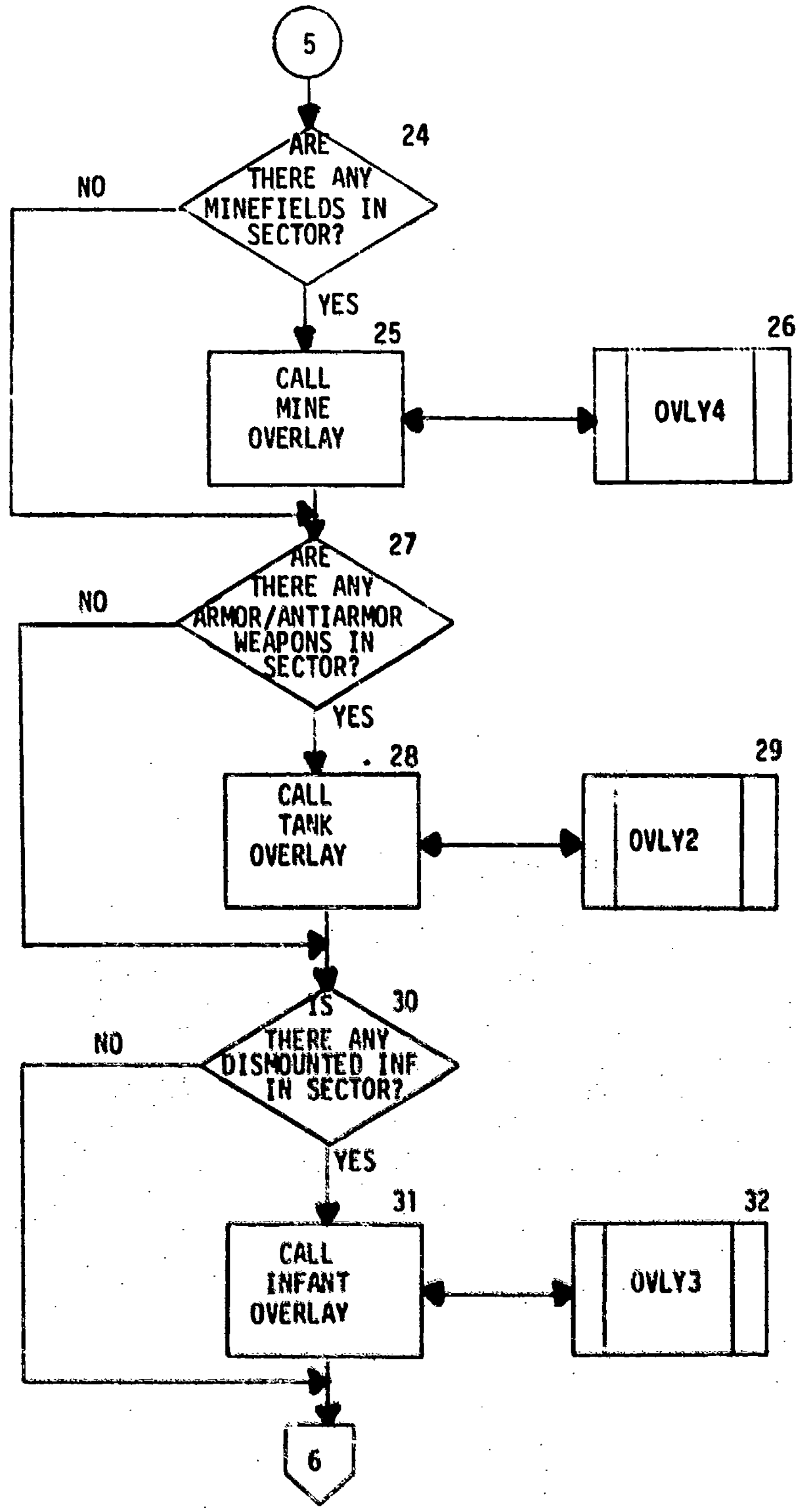

Figure 7. SUPER flow diagram (continued).

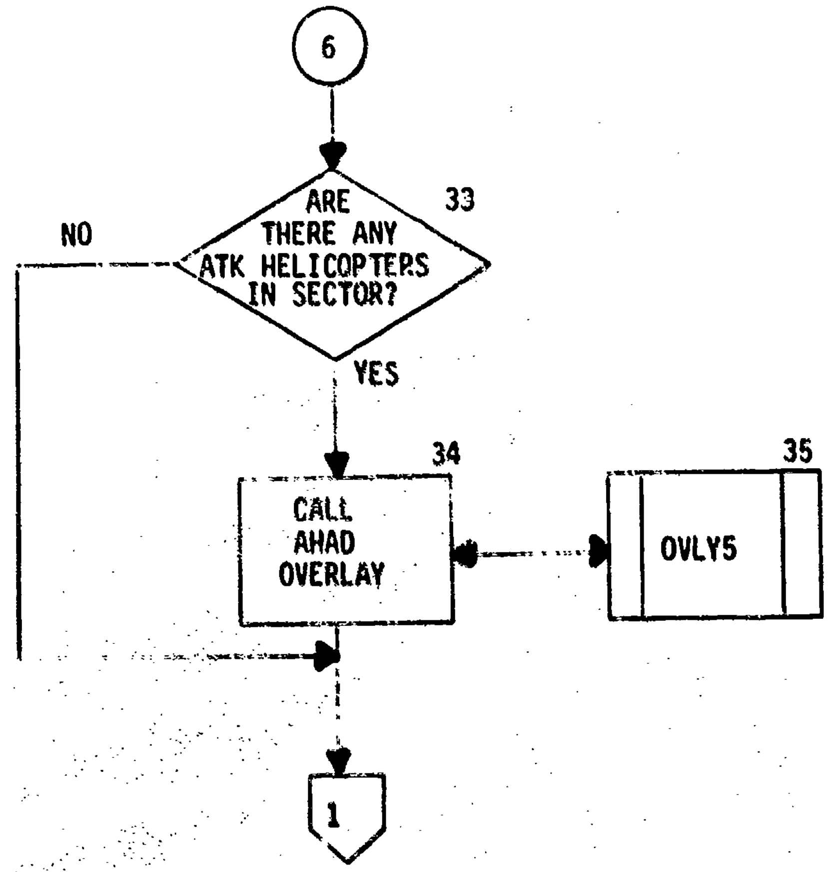

Figure 7. SUPER flow diagram (continued).

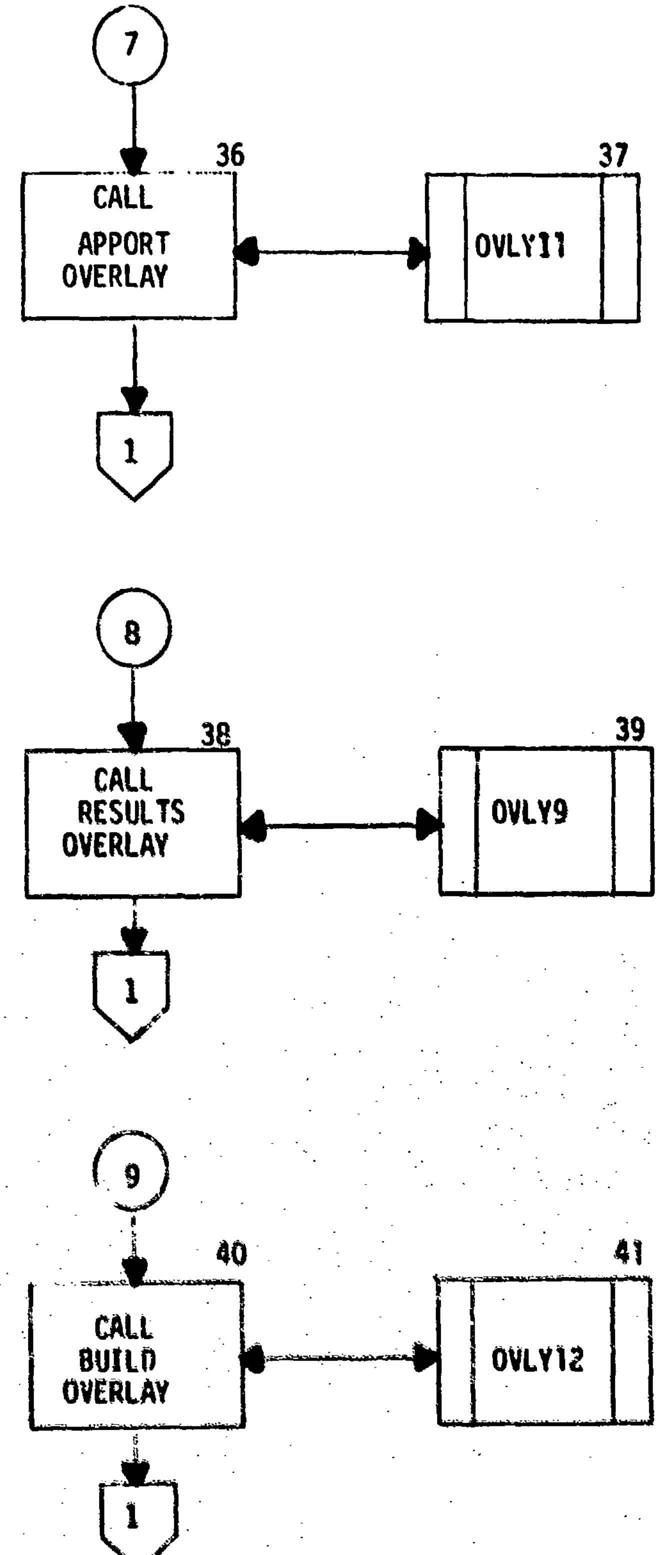

Figure 7. SUPER flow diagram (continued).

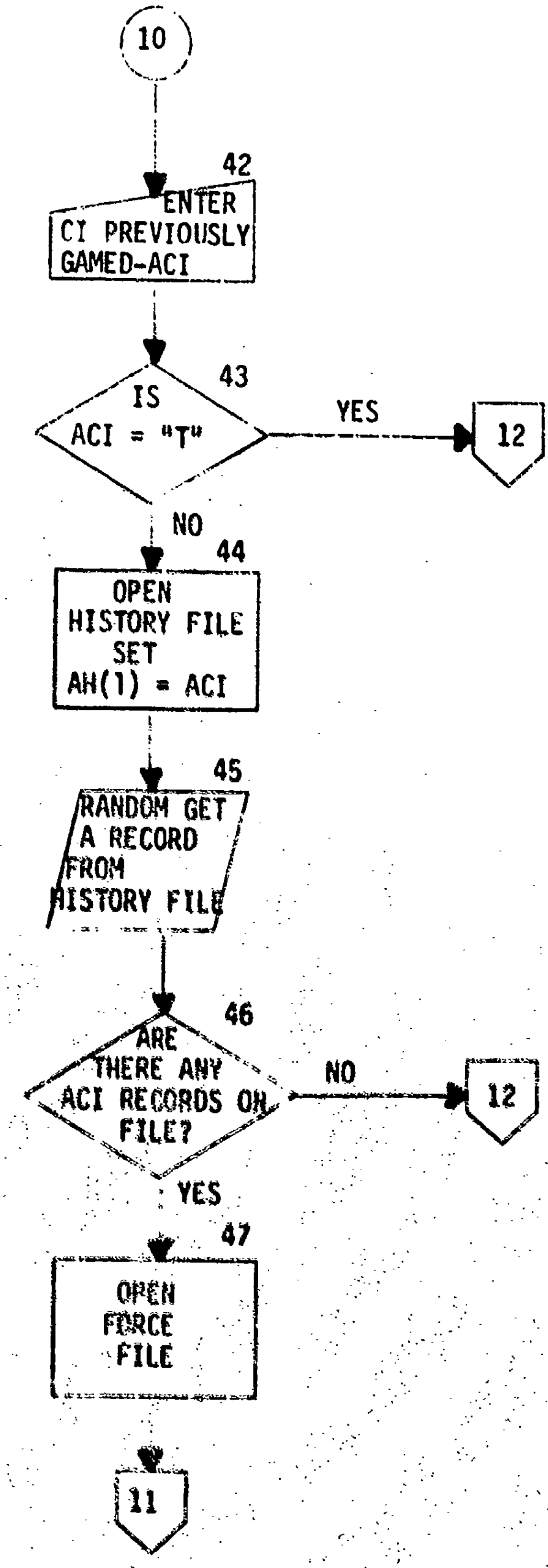

Figure 7. SUPER flow diagram (continued).

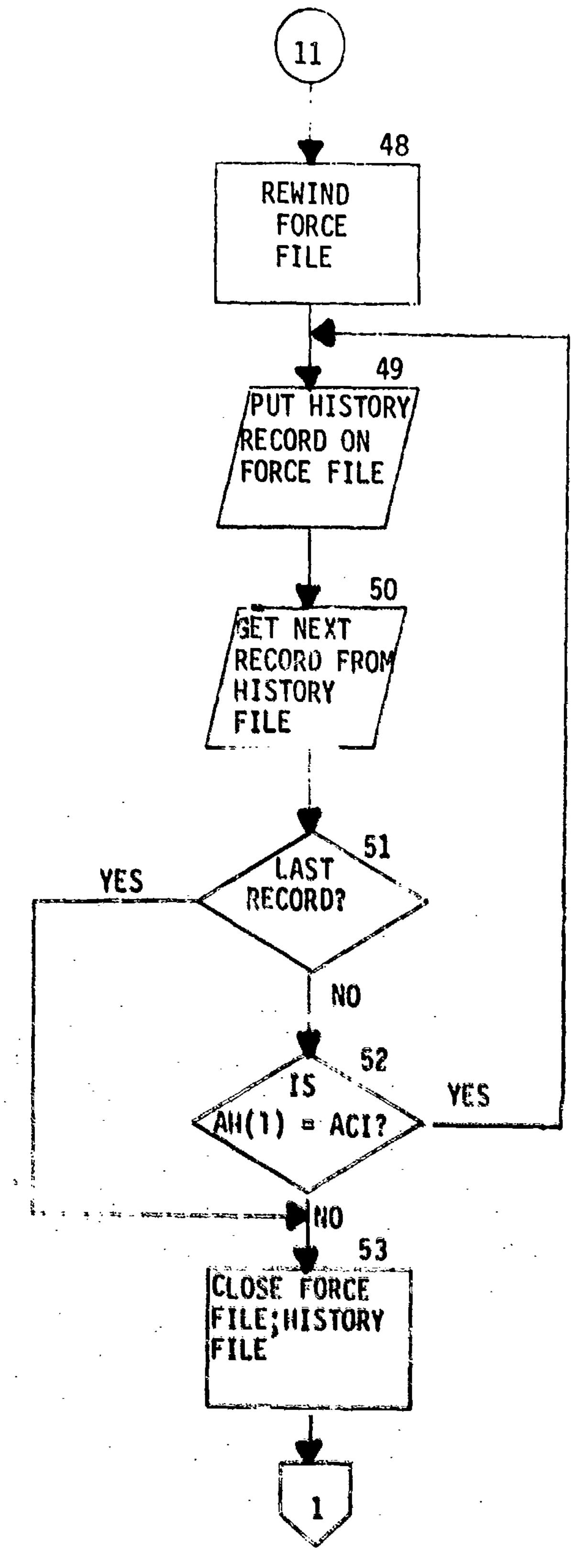

Figure 7. SUPER flow diagram (continued).

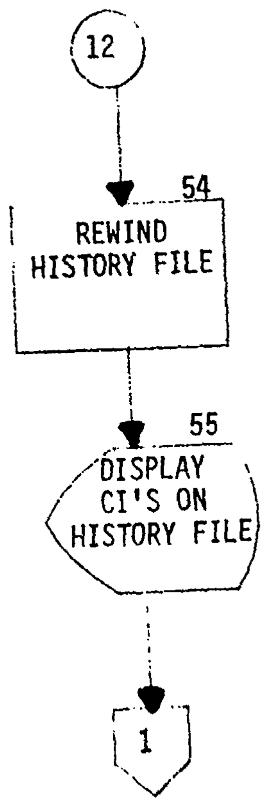

Figure 7. SUPER flow diagram (continued).

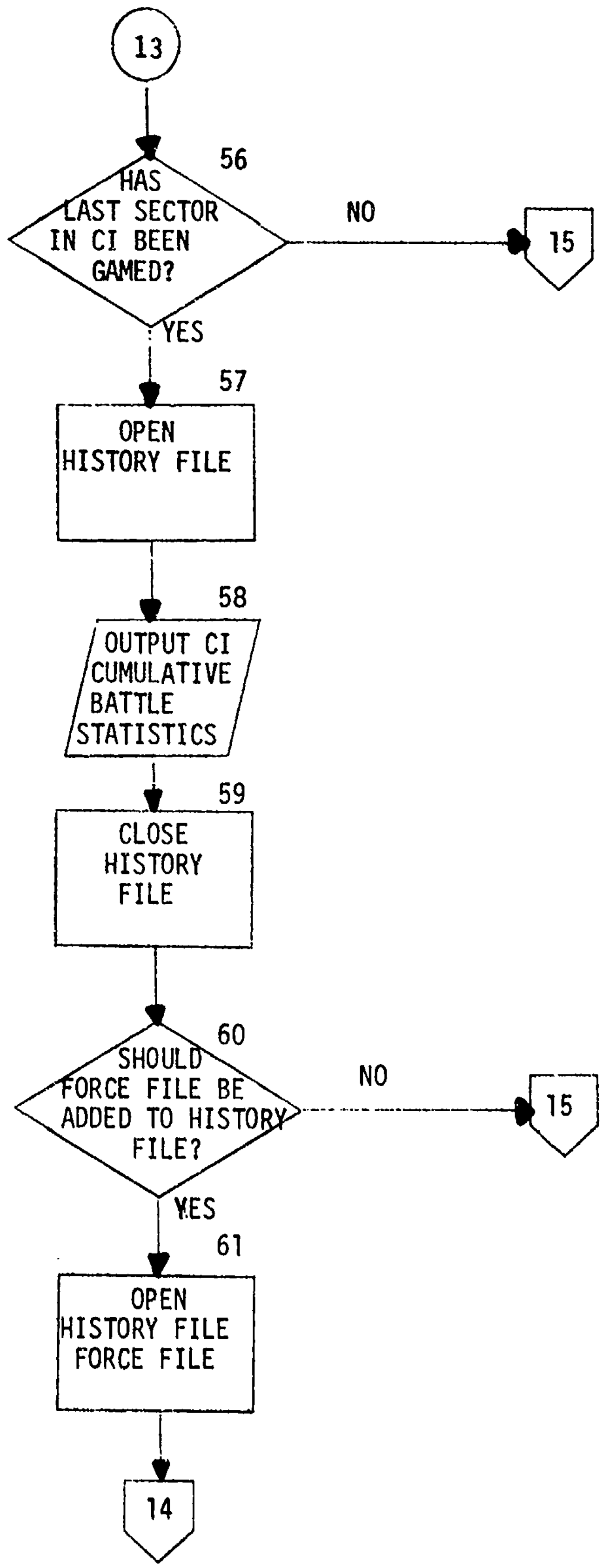

Figure 7. SUPER flow diagram (continued).

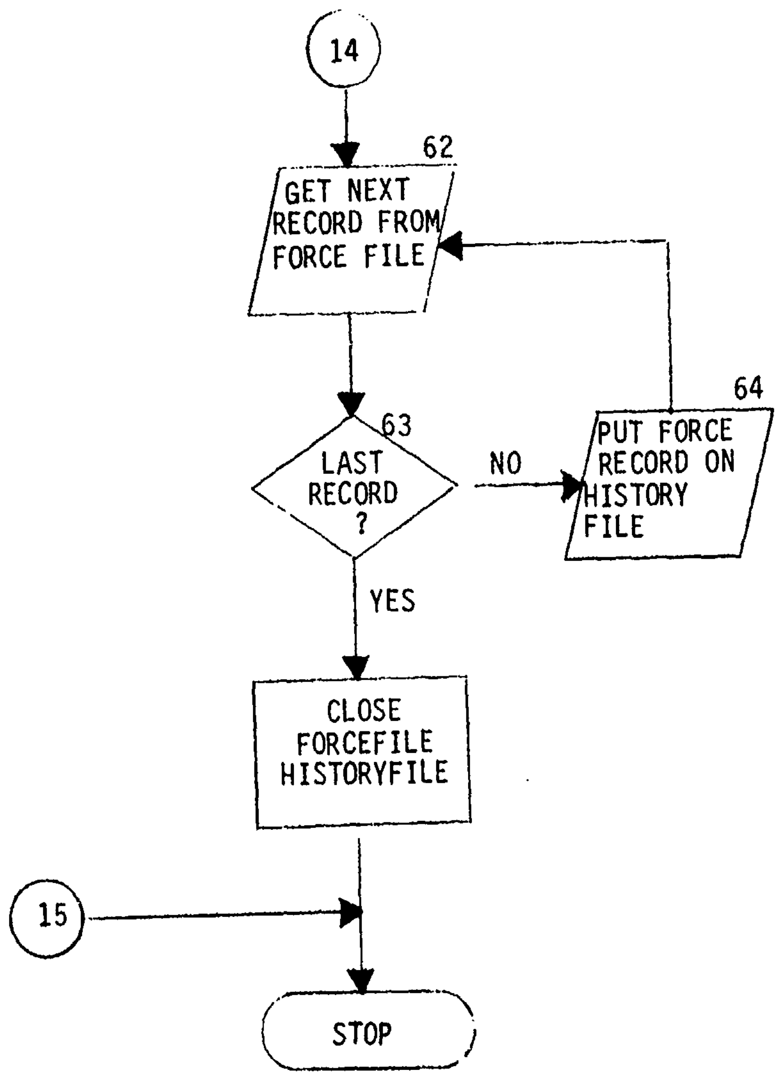

Figure 7. SUPER flow diagram (concluded).

3. inspects the types of weapon systems being played and determines the appropriate combat overlays to which to branch (blocks 14-35),

4. accepts input of TACAIR losses, which are determined external to the Jiffy Game (blocks 18-20),

5. records the forces remaining at the end of a critical incident on the HISTORY file (blocks 60-64),

6. outputs the cumulative battle statistics at the end of a critical incident (blocks 56-59, and

7. provides the gamers with the capability to reinitialize the forces at some previously gamed critical incident on the HISTORY file (blocks 42-55). The FORTRAN source code for SUPER is provided in figure F-1, and a list of the program variables is given in table F-1.

(b) INIT. The logic flow diagram for INIT is presented in figure 8. This routine initializes the arrays in /DATA/ common. Note that the firepower score array (FPS) is initialized from the classified data array (CLDATA). In addition, INIT zeros the SHOTS array and initializes the word packing array variables (PACK). The source code for INIT is provided in figure F-2. All the program variables used in INIT are common variables, and they are defined in table F-1.

(c) INDEX5. This routine is a subfunction that calculates a one-dimensional subscript from a five-dimensional variable. The flow diagram for INDEX5 is given in figure 9. The FORTRAN source listing is contained in figure F-3. A list of the program variables used in INDEX5 is provided in table F-3.

(d) LOSS. This subroutine is used to subtract weapon systems lost in the combat assessment routines from the weapon system arrays for both forces (ELMT). The LOSS flow diagram is presented in figure 10. If the gamer decides not to subtract the losses from the weapon system array, the losses are removed from the loss array (ALOSS). This allows the gamer to replay the combat, if the original assessment is for the same reason invalid. A list of the LOSS program variables is contained in table F-4, and a FORTRAN source code listing may be found in figure F-4.

(e) DISPLAY. This subroutine is called during gaming to display the status of specified units and parent units. The logic flow diagram for the DISPLAY subroutine is given in figure 11. The gamer has the option to display a particular unit or all units within a specified parent unit. The unit status parameters displayed include the unit effectiveness of the parent and subordinate unit(s) and the quantity and type of weapon systems remaining in each unit. The FORTRAN source code for DISPLAY is presented in figure F-5. A list of the program variables is contained in table F-5.

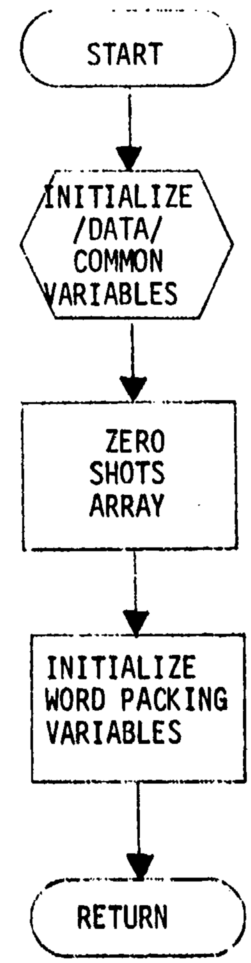

Figure 8. INIT flow diagram.

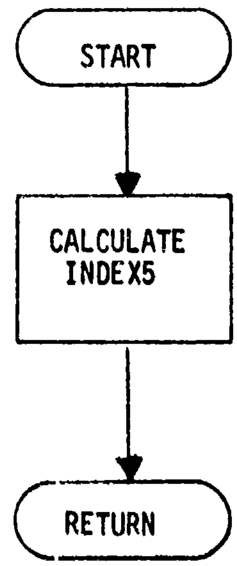

Figure 9. INDEX5 flow diagram.

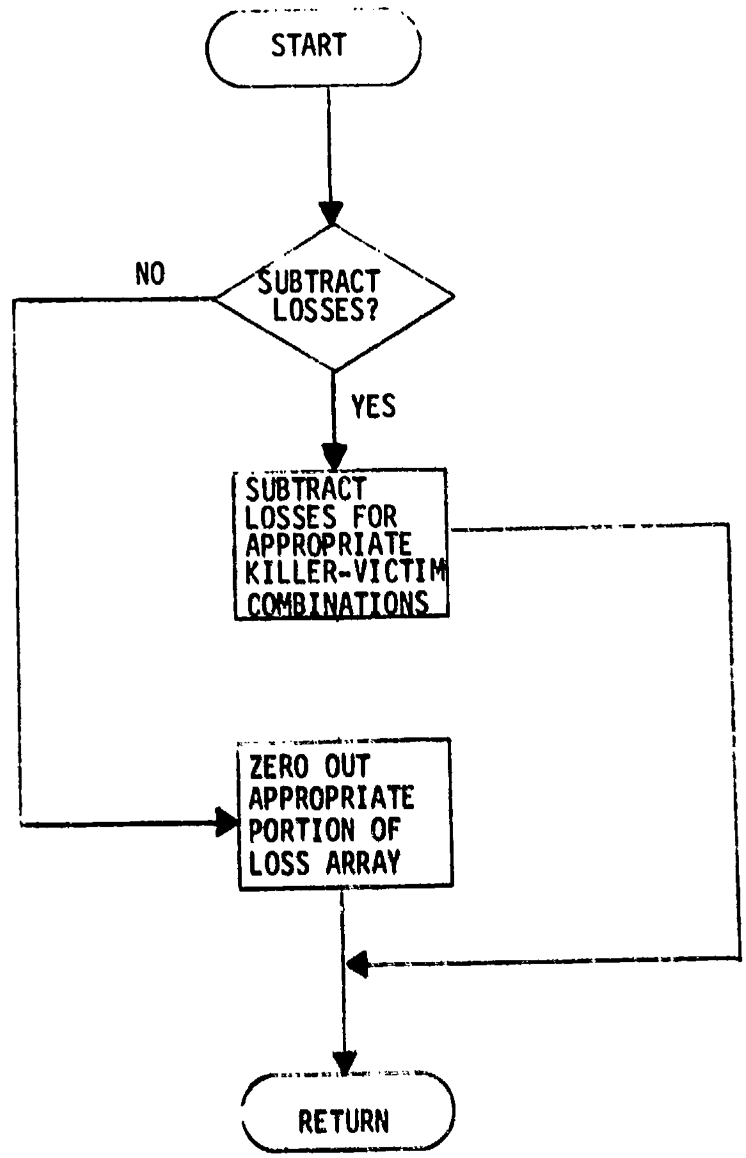

Figure 10. LOSS flow diagram.

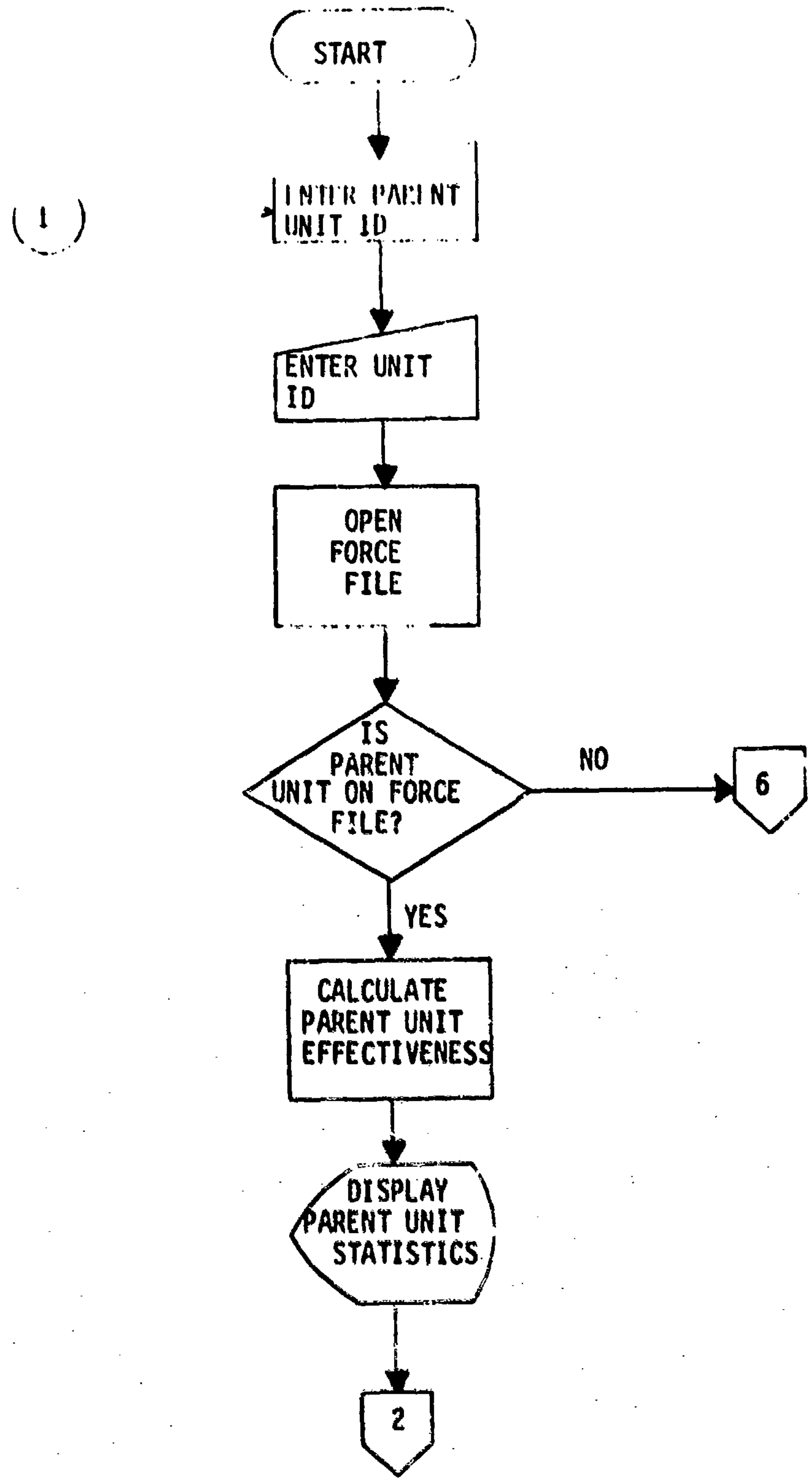

Figure 11. DISPLAY logic diagram.
(Continued next page)

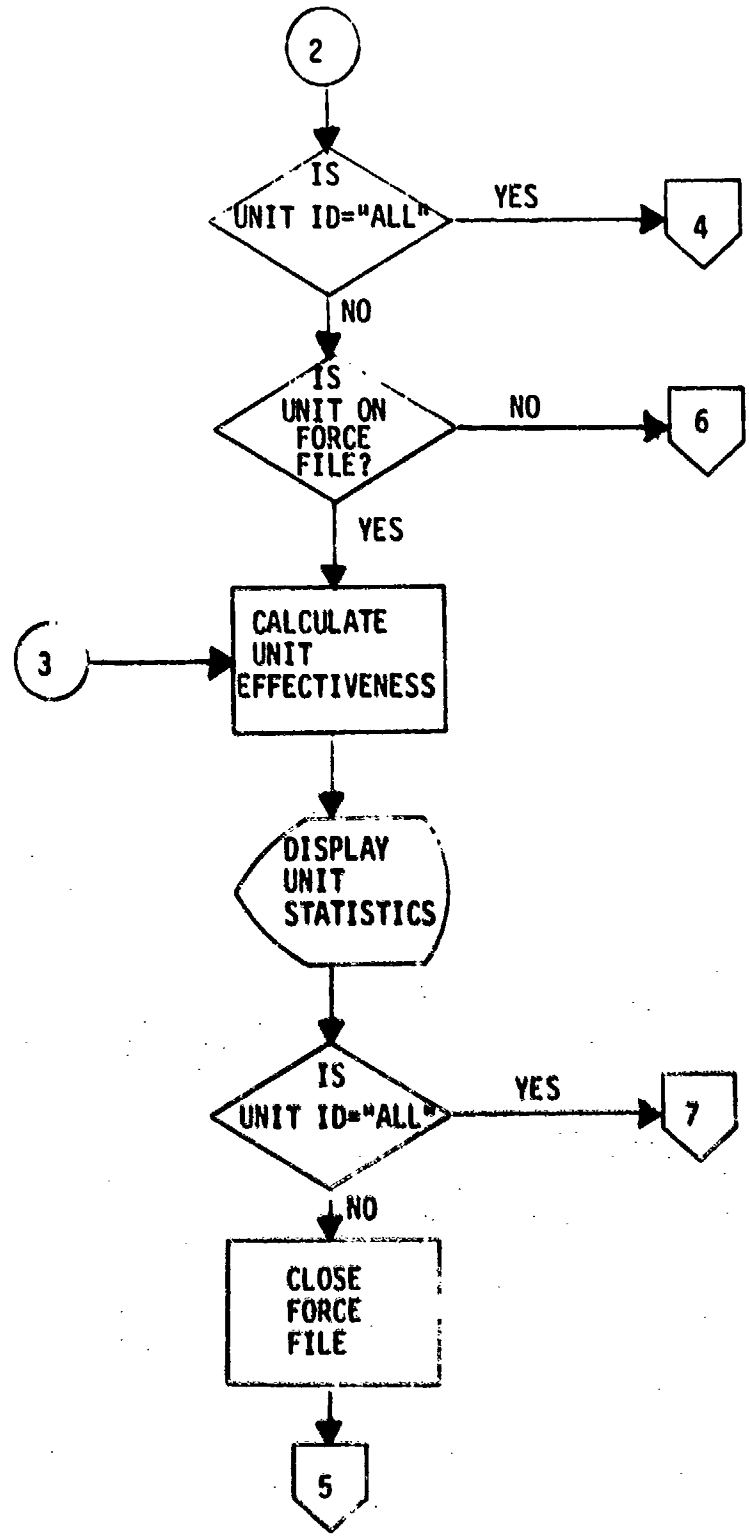

Figure 11. DISPLAY logic diagram (continued).

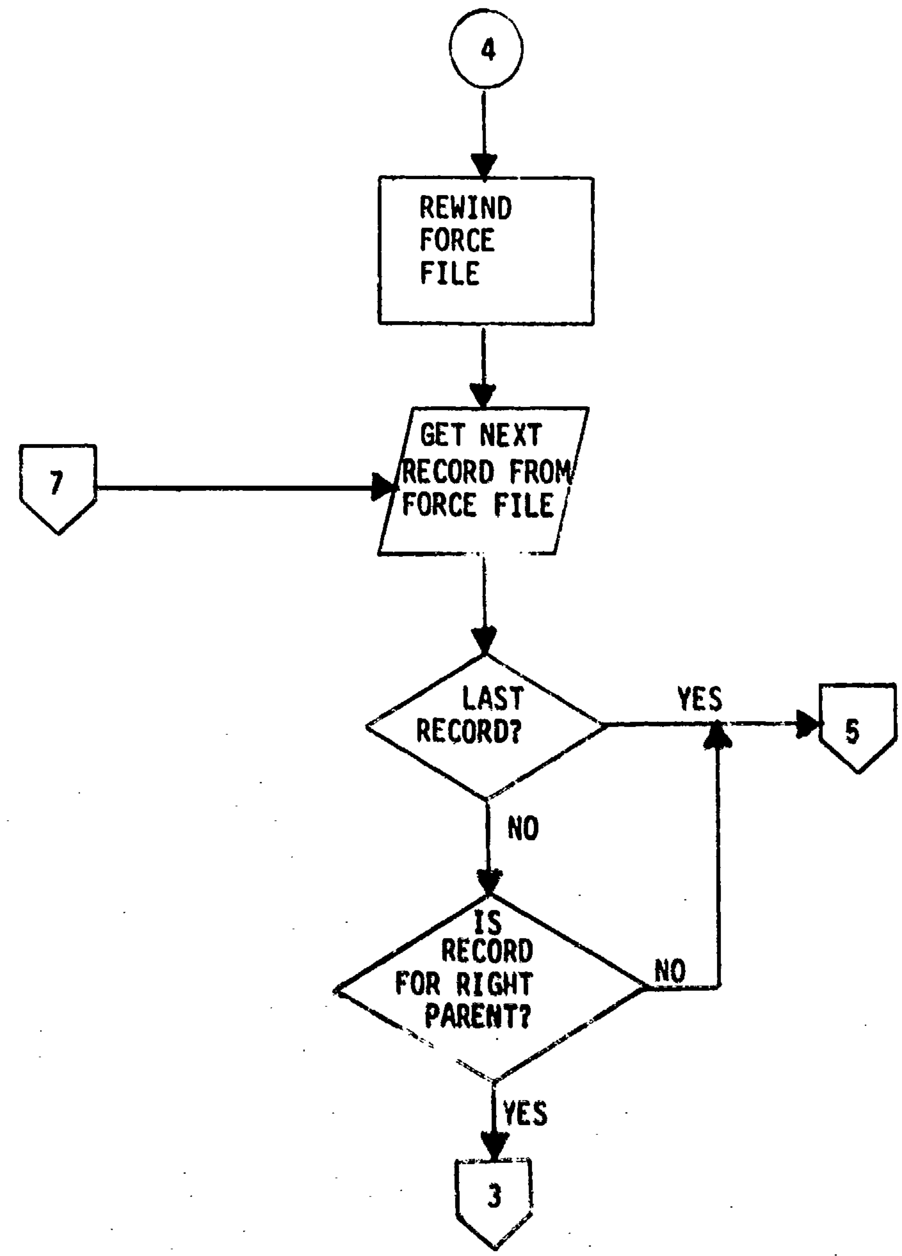

Figure 11. DISPLAY logic diagram (continued).

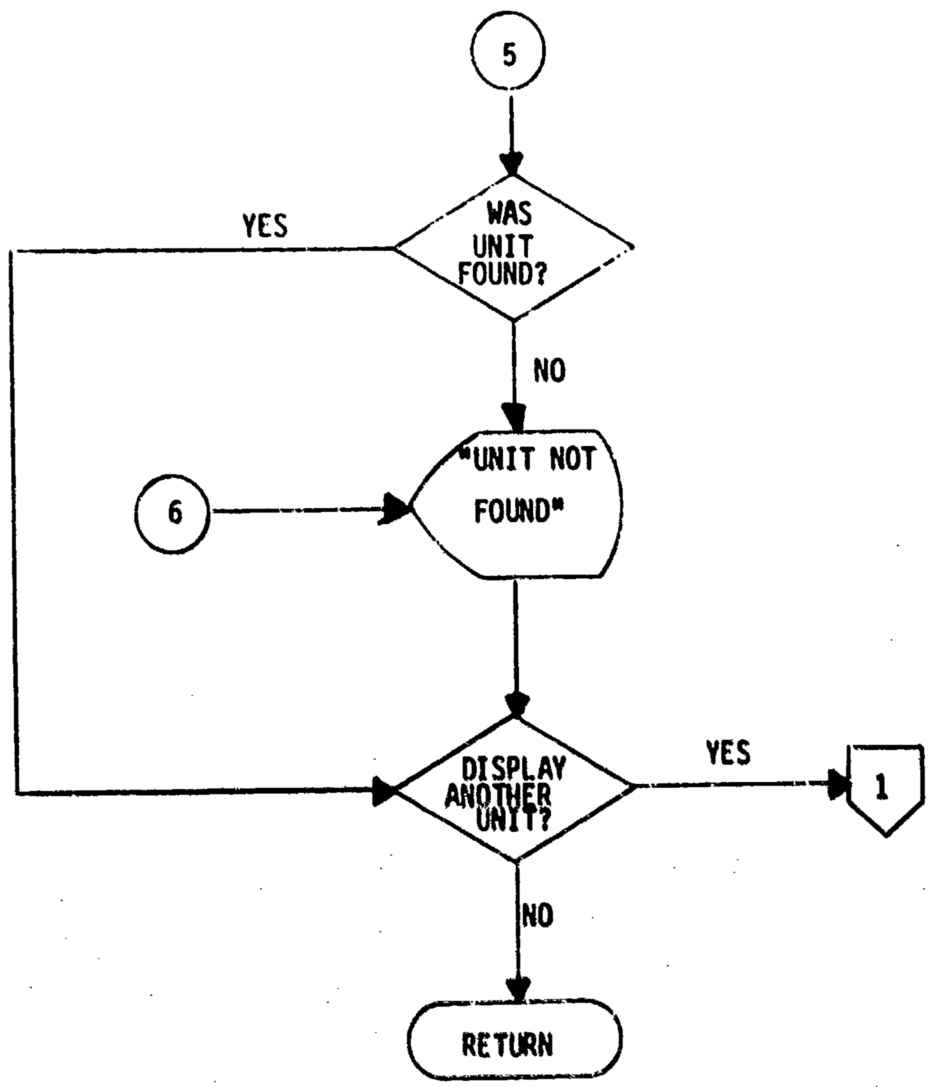

Figure 11. DISPLAY logic diagram (concluded).

(2) OVERLAY 1. The ROFA overlay (OVL 1) is accessed from the main Jiffy Game program at DECISION POINT number 2 (see table 1). The primary function of this routine is to determine and display, for the sector being gamed, the rate of advance of the attacking force; the length of the battle; the total distance covered by the attacker; the maneuver, fire support, and total firepower scores for each force; and the corresponding attacker:defender firepower ratios. To accomplish this, a number of parameters representing environmental and tactical military conditions that influence the nature of the conflict must be input interactively. Since these same factors also influence the other combat assessments, they are initialized here as variables in the blank COMMON area; thus, none of the combat assessment overlays can be accessed until this routine has been executed. The logic flow diagram for OVLY 1 is given in figure 12. There are no subroutines contained in this overlay although the INDEX5 function (see paragraph 4b(1)(c)) from OVLY 0 is utilized for extracting rate of advance values from the data array. The FORTRAN source code for ROFA is given in figure G-1, and the program variables are listed in table G-1.

(3) OVERLAY 2. Program OVLY 2 (TANK) is the third of the combat assessment routines called in the main Jiffy Game program (OVLY 0) from DECISION POINT number 3 (see table 1 and figure 7). In this overlay, the losses due to combat involving tanks, other armored combat vehicles, and antitank weapons are calculated and displayed. The overlay contains no subroutines but does call the INDEX5 function (see paragraph 4b(1)(c)) when extracting single shot kill probabilities (SSKPs) for assessments and also the LOSS subroutine (see paragraph 4b(1)(d)) after the losses have been assessed. The SSKP data used in this routine reside on the classified random access file (CLDATA); other data are either contained in the common areas or initialized in the problem itself. The flow diagram for OVLY 2 is given in figure 13. The TANK routine cycles through a series of nested DO loops in assessing losses for each possible combination of targets and firers for both forces. The gamer inputs a range band index, which initiates the assessment logic cycle. At the end of each assessment cycle, the gamer either inputs another range band index to continue with another cycle or signals that the assessments are completed. When the assessments are finished, the overall results are displayed, the LOSS subroutine is called, and control is returned to the SUPER overlay. The FORTRAN source code for OVLY 2 is given in figure H-1, and the program variables are listed in table H-1.

(4) OVERLAY 3. Program OVLY 3 (INFANT) is the fourth combat assessment routine accessed by SUPER (the main Jiffy Game program) from DECISION POINT number 3 (see table 1 and figure 7) and is called whenever both forces contain infantry personnel in the weapon system (ELMT) array. The function of this overlay is to compute and display the losses incurred as a result of dismounted infantry combat for the sector being gamed. There are no subroutines included within this overlay; the LOSS subroutine of OVLY 0 (see paragraph 4b(1)(c)) is called at the end of the assessments. Figure 14 contains the logic flow diagram for OVLY 3. The routine requires a number of interactive gamer inputs, which set the parameters necessary to carry out a

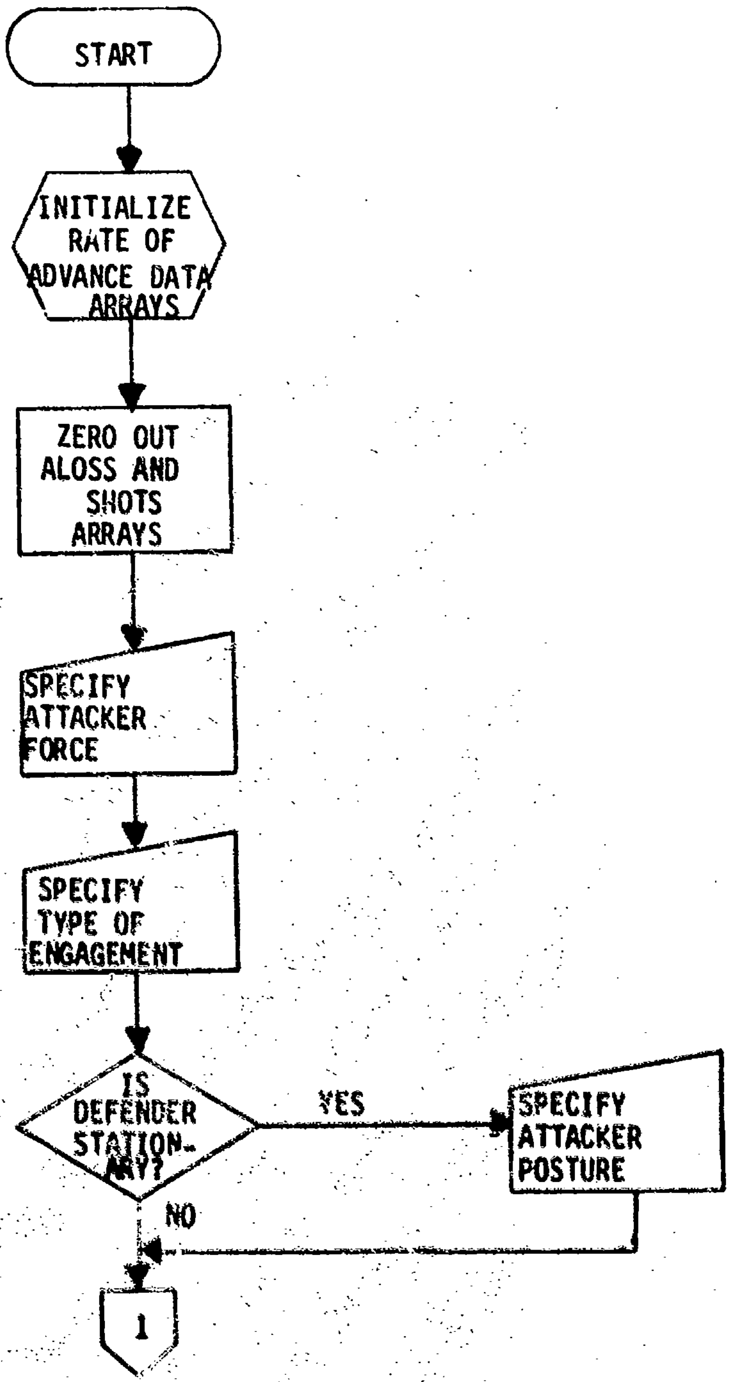

Figure 12. ROFA (OVLY 1) flow diagram.
(Continued next page)

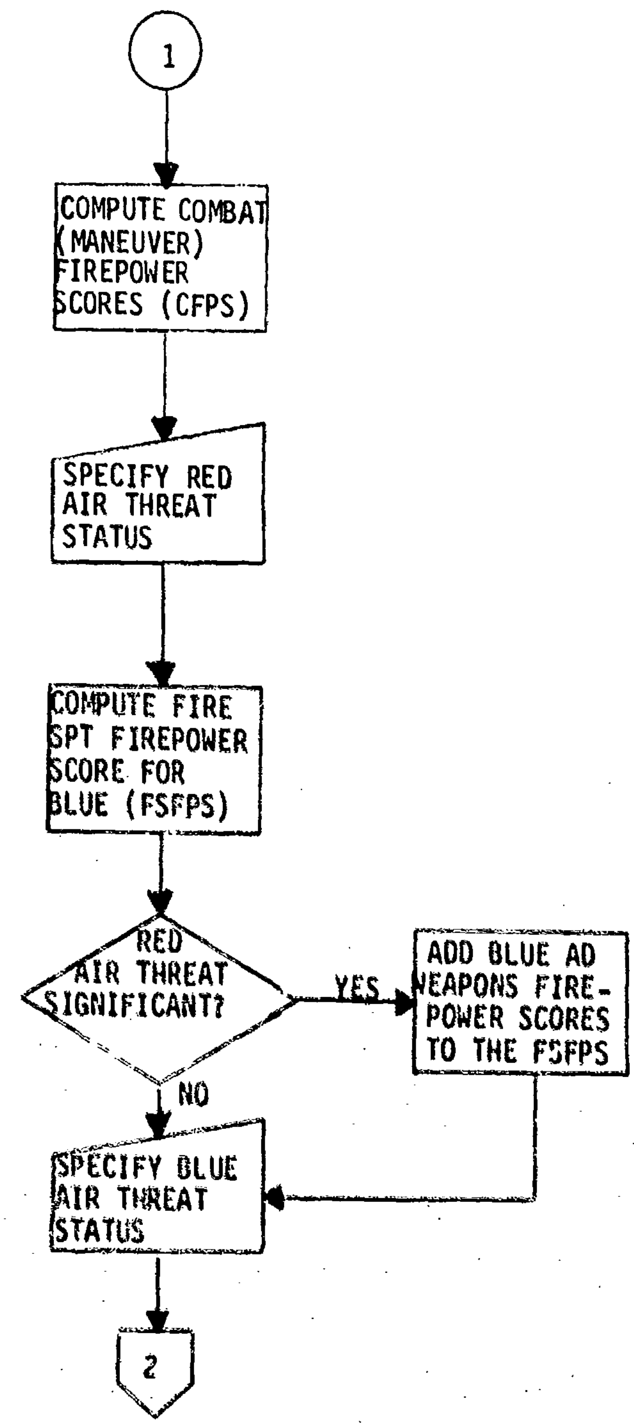

Figure 12. ROFA (OVLY 1) flow diagram (continued).

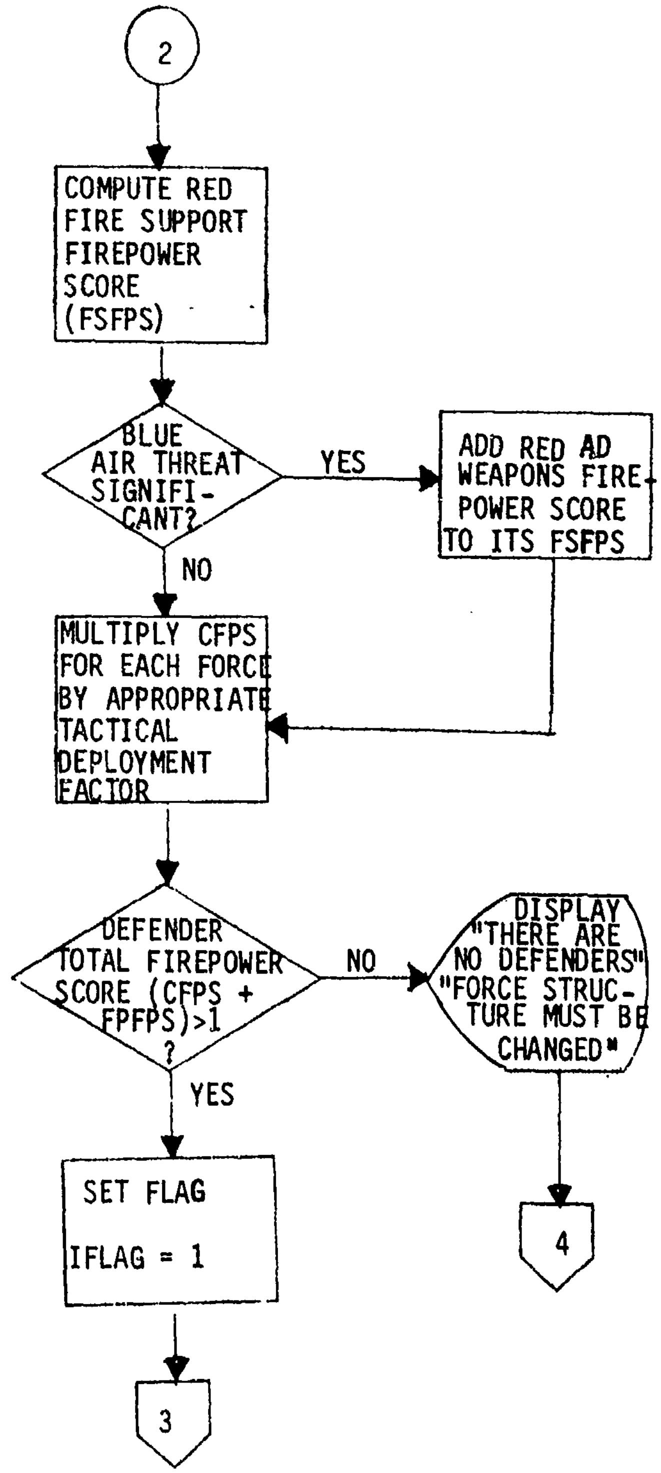

Figure 12. ROFA (OVLY 1) flow diagram (continued).

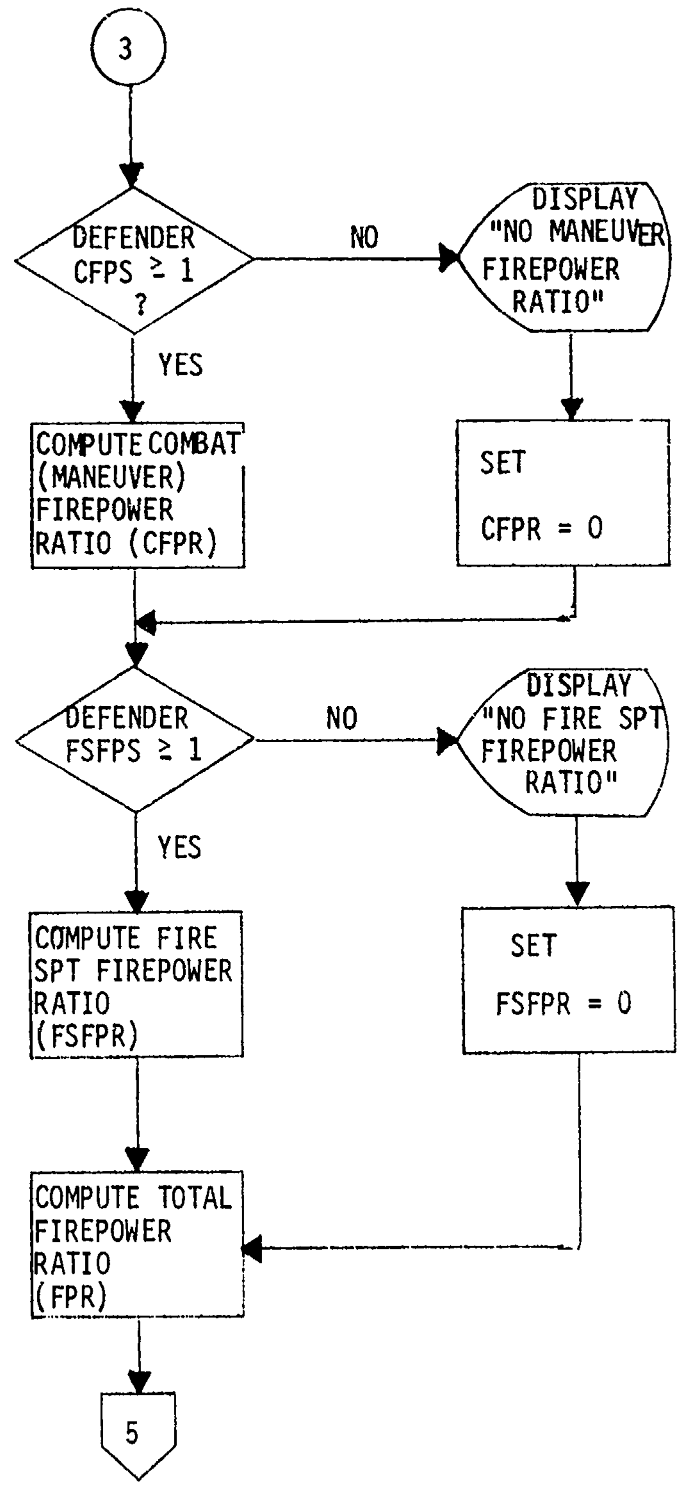

Figure 12. ROFA (OVLY 1) flow diagram (continued).

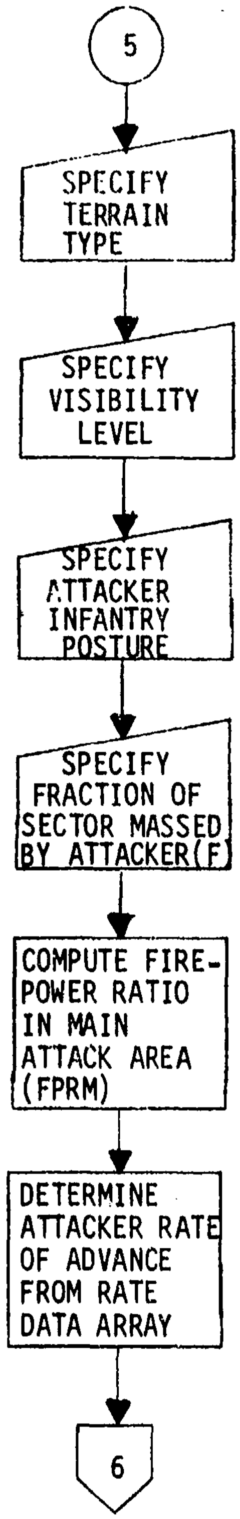

Figure 12. ROFA (OVLY 1) flow diagram (continued).

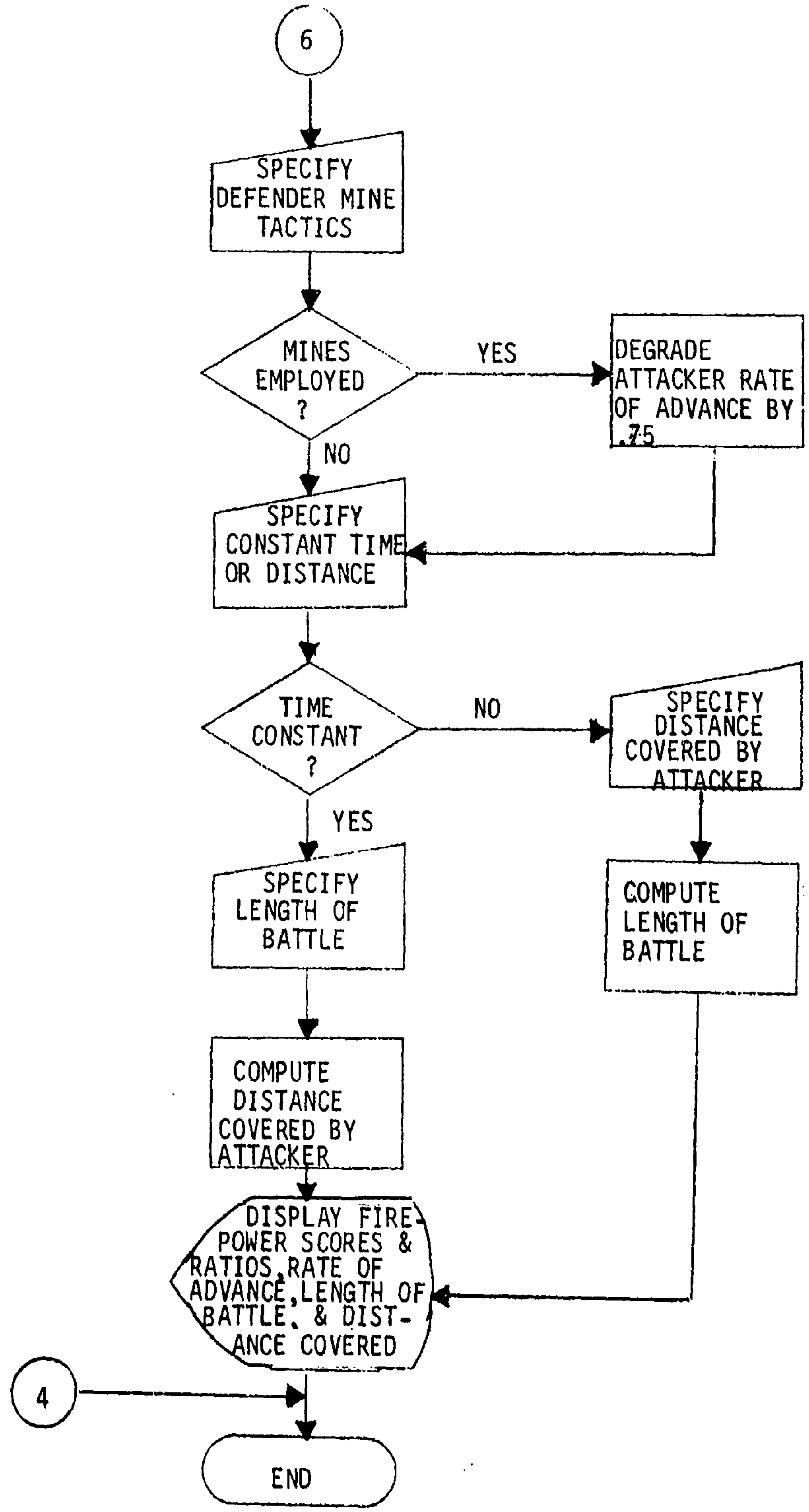

Figure 12. ROFA (OVLY 1) flow diagram (concluded).

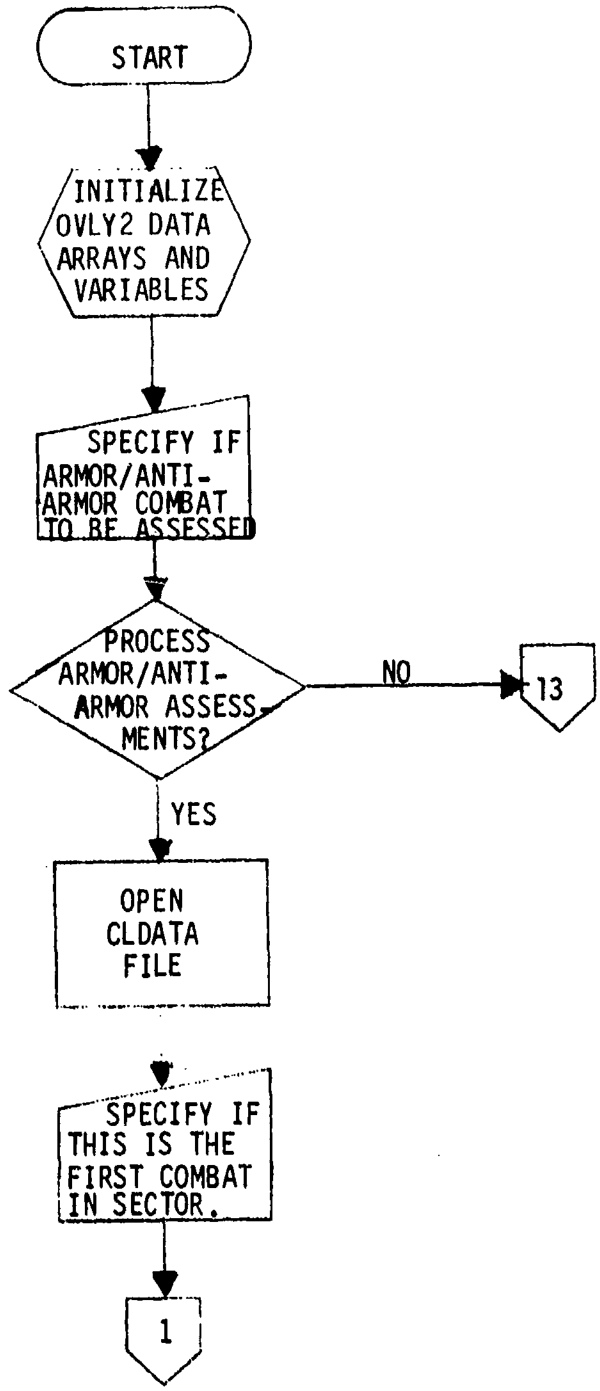

Figure 13. TANK (OVLY 2) flow diagram.
(Continued next page)

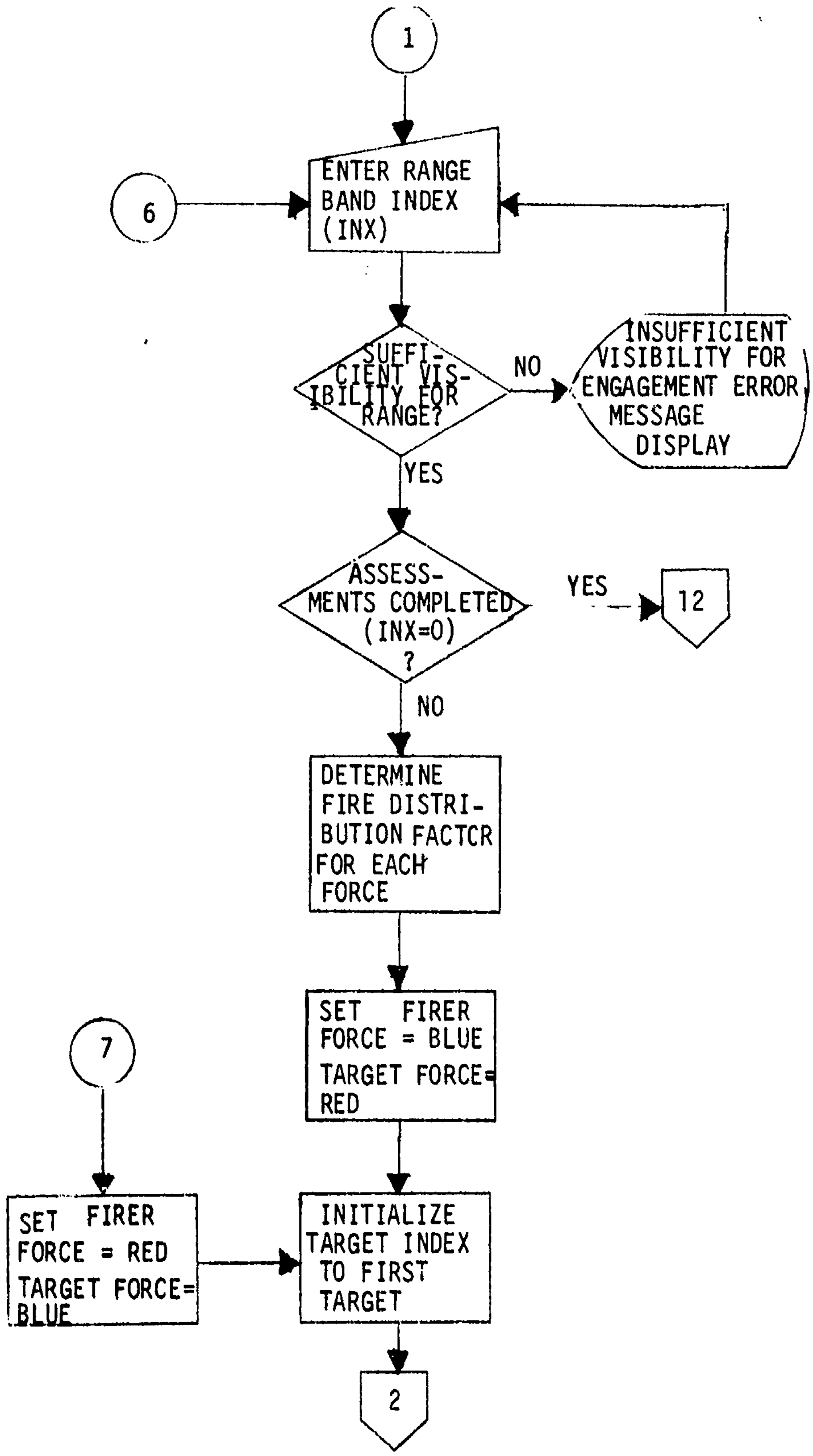

Figure 13. TANK (OVLY 2) flow diagram (continued).

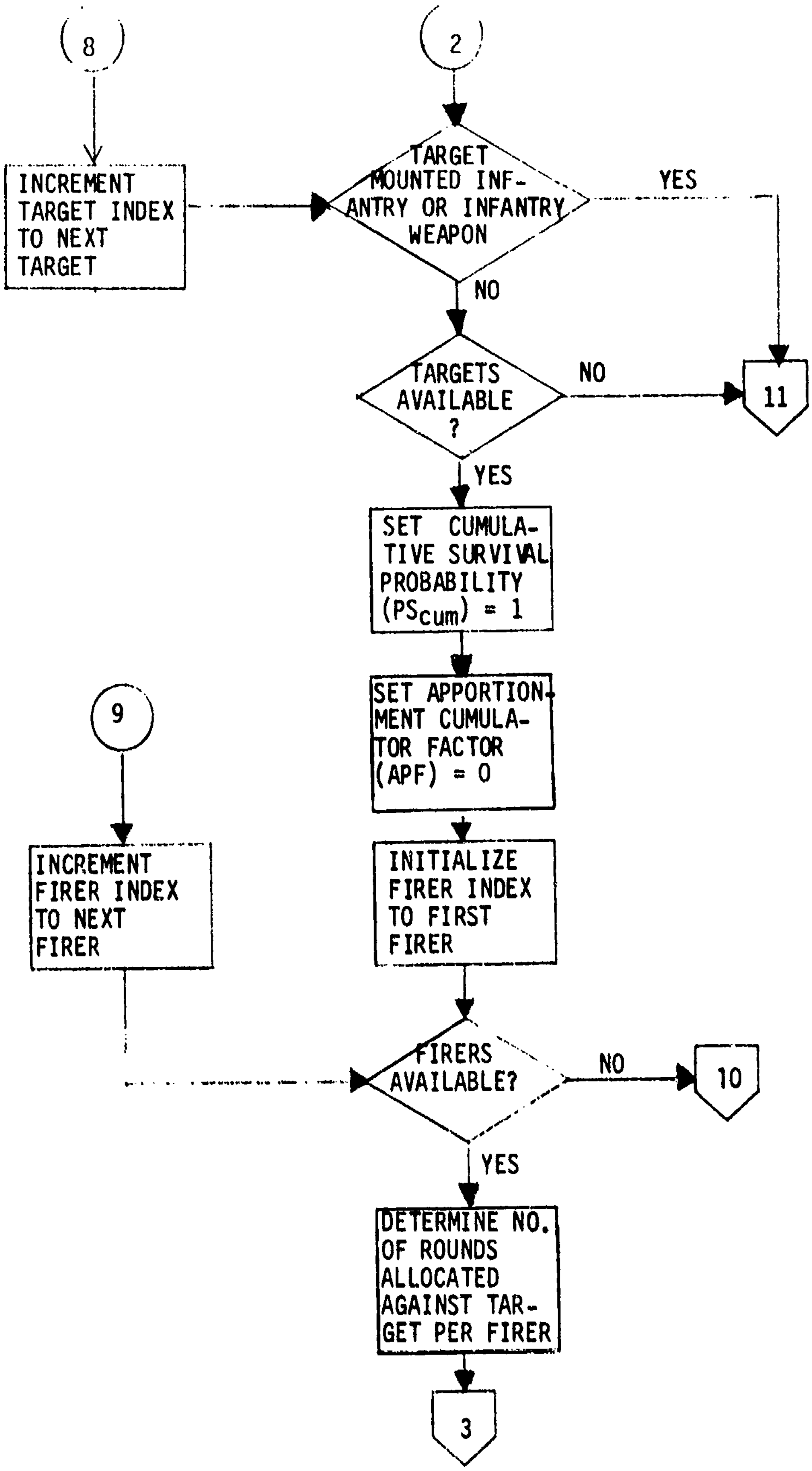

Figure 13. TANK (OVLY 2) flow diagram (continued).

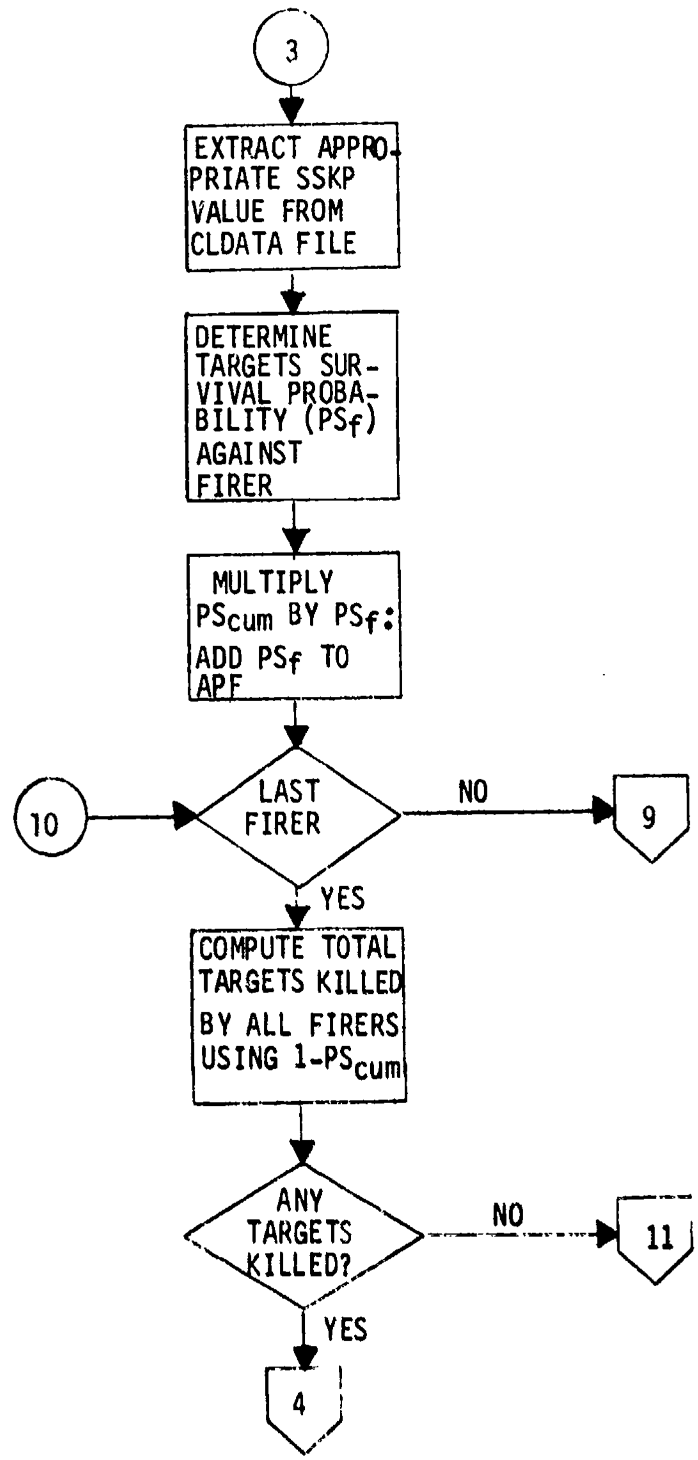

Figure 13. TANK (OVLY 2) flow diagram (continued).

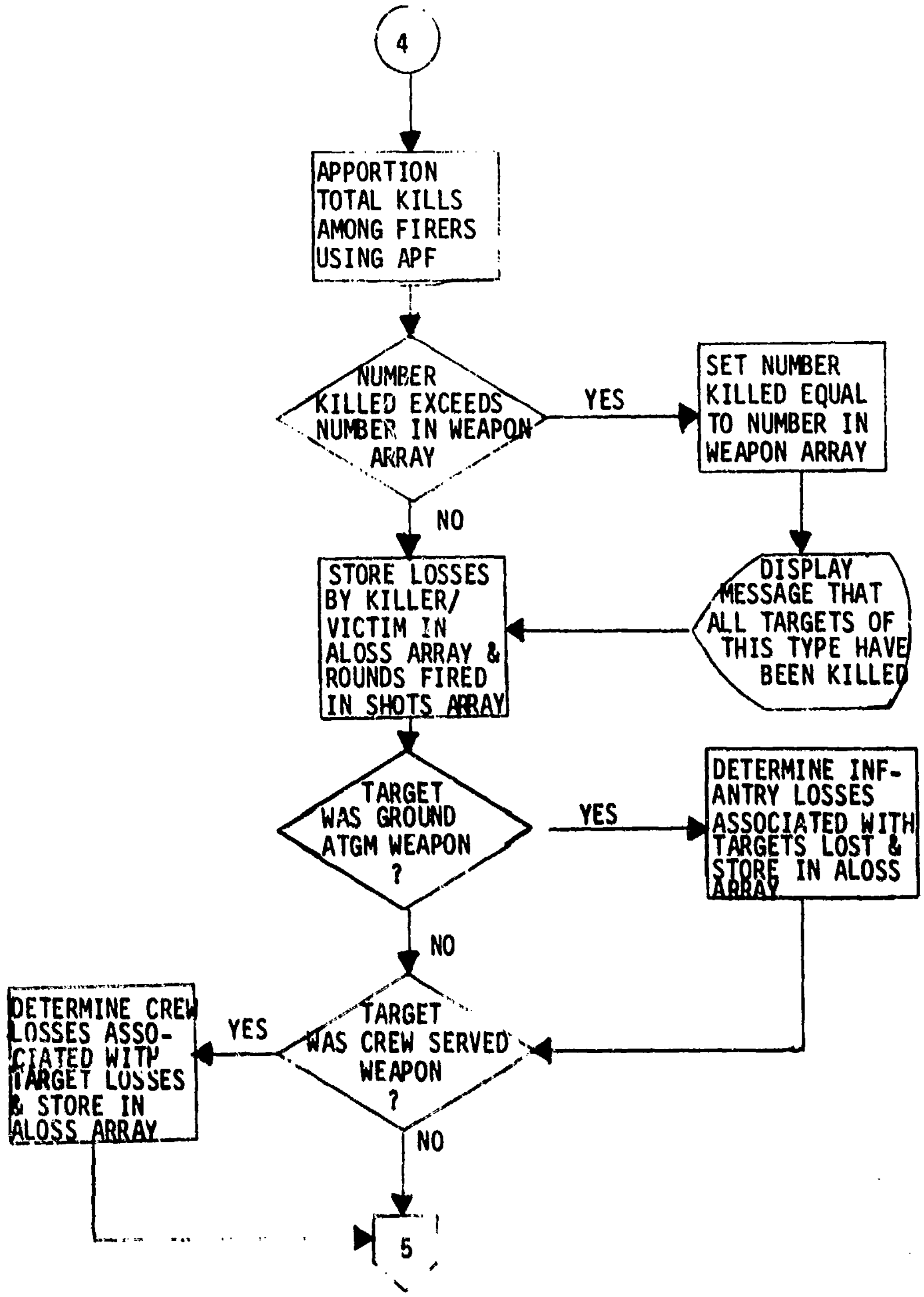

Figure 13. TANK (OVLY 2) flow diagram (continued).

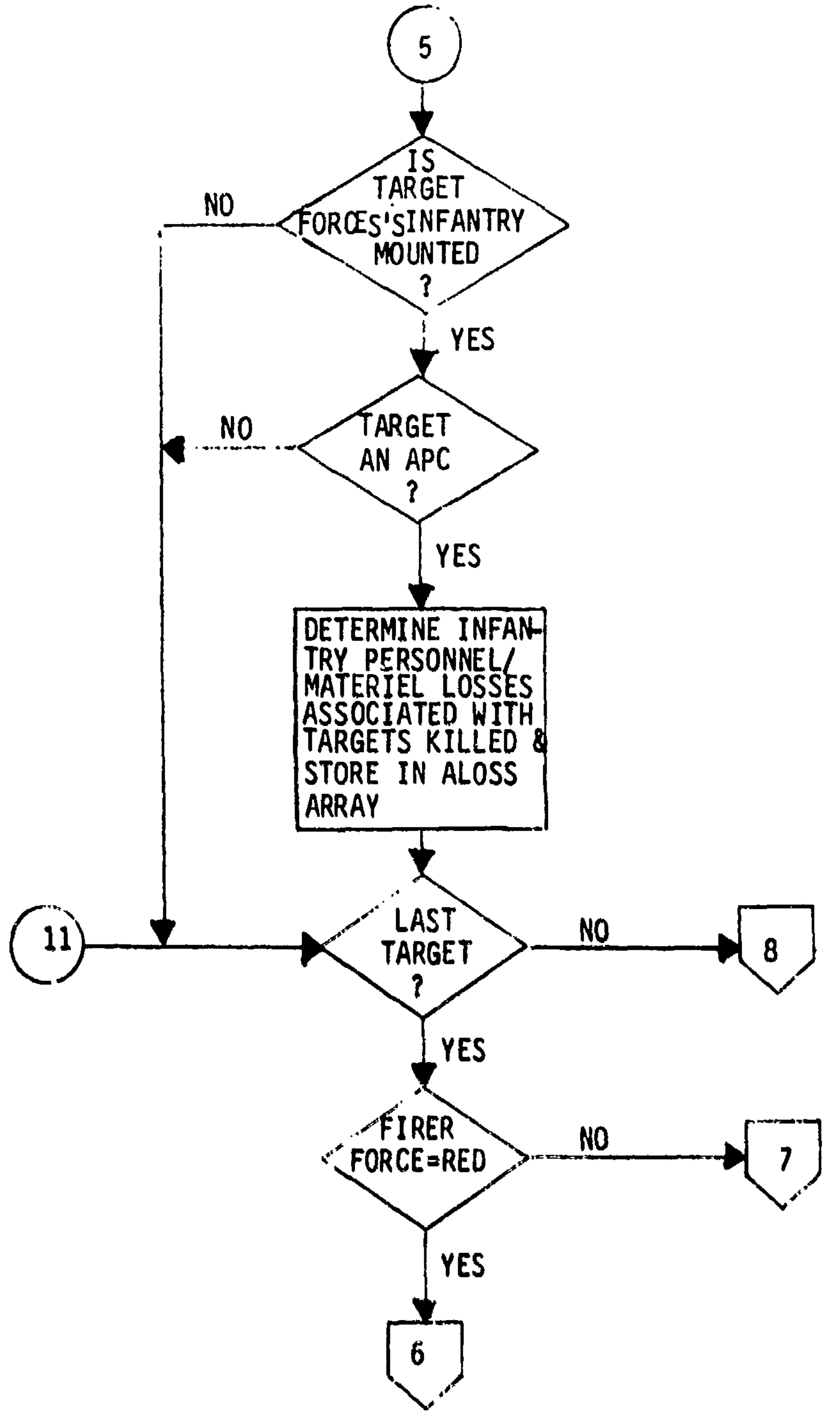

Figure 13. TANK (OVLY 2) flow diagram (continued).

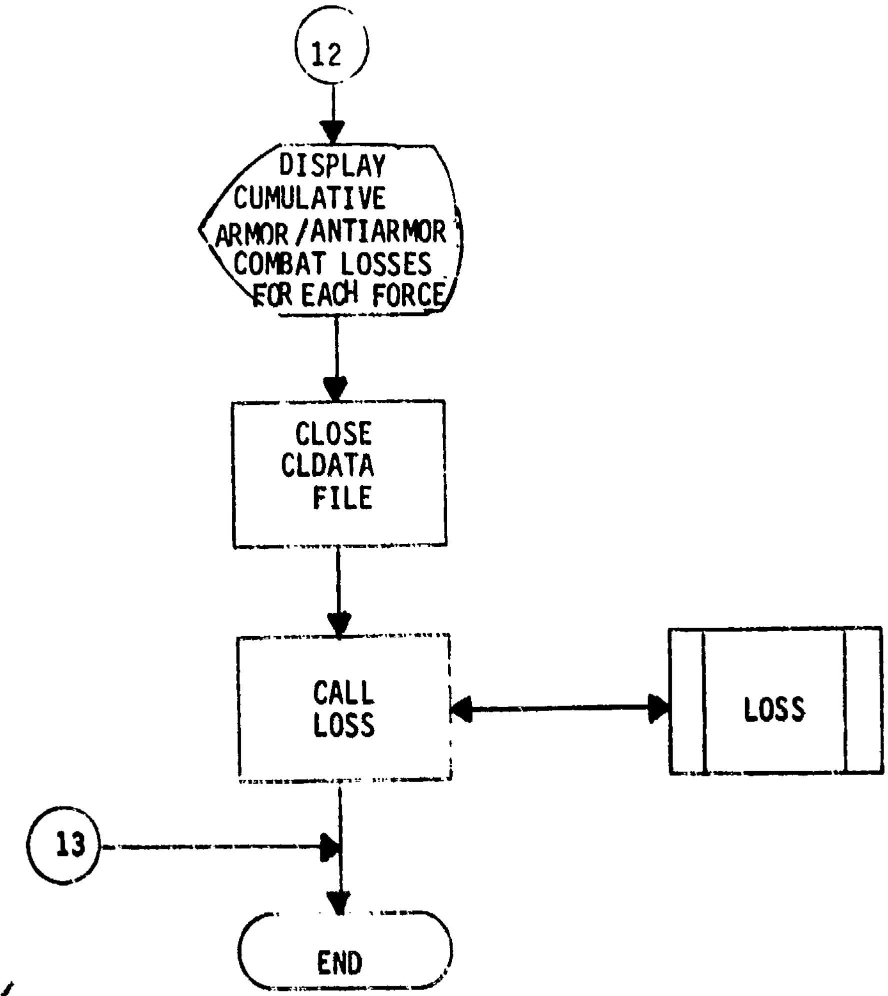

Figure 13. TANK (OVLY 2) flow diagram (concluded).

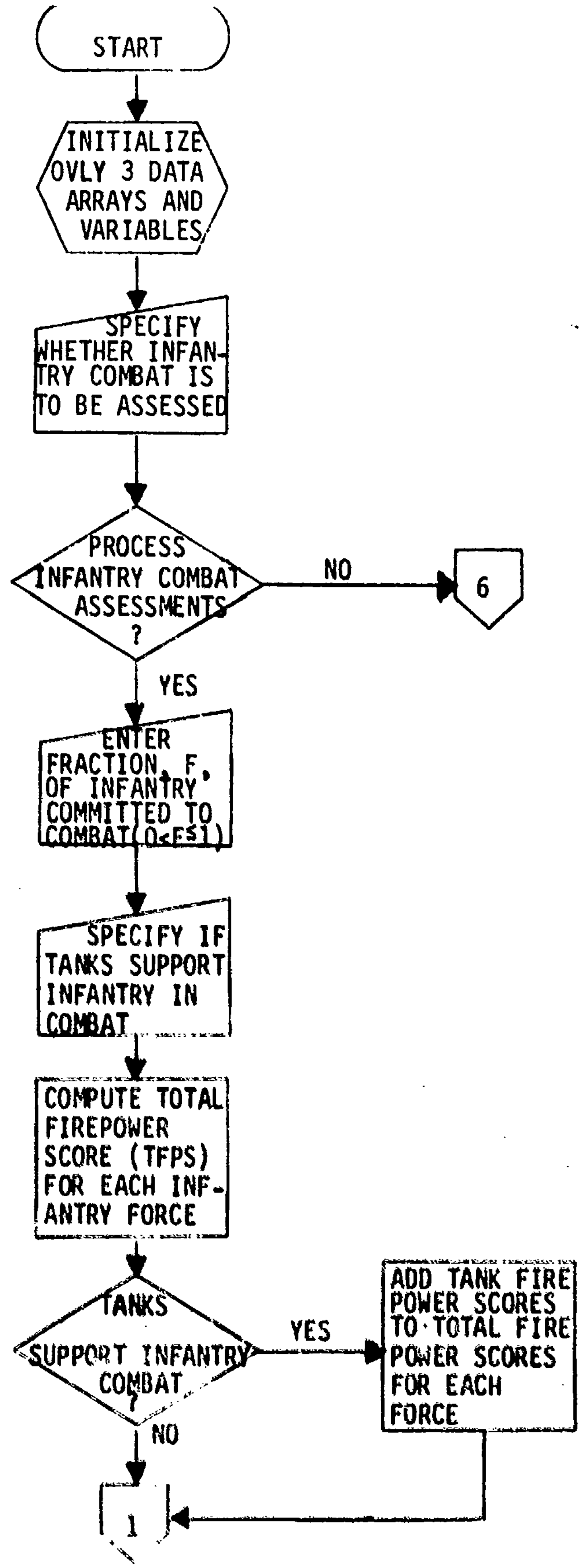

Figure 14. INFANT (OVLY 3) flow diagram.
(Continued next page)

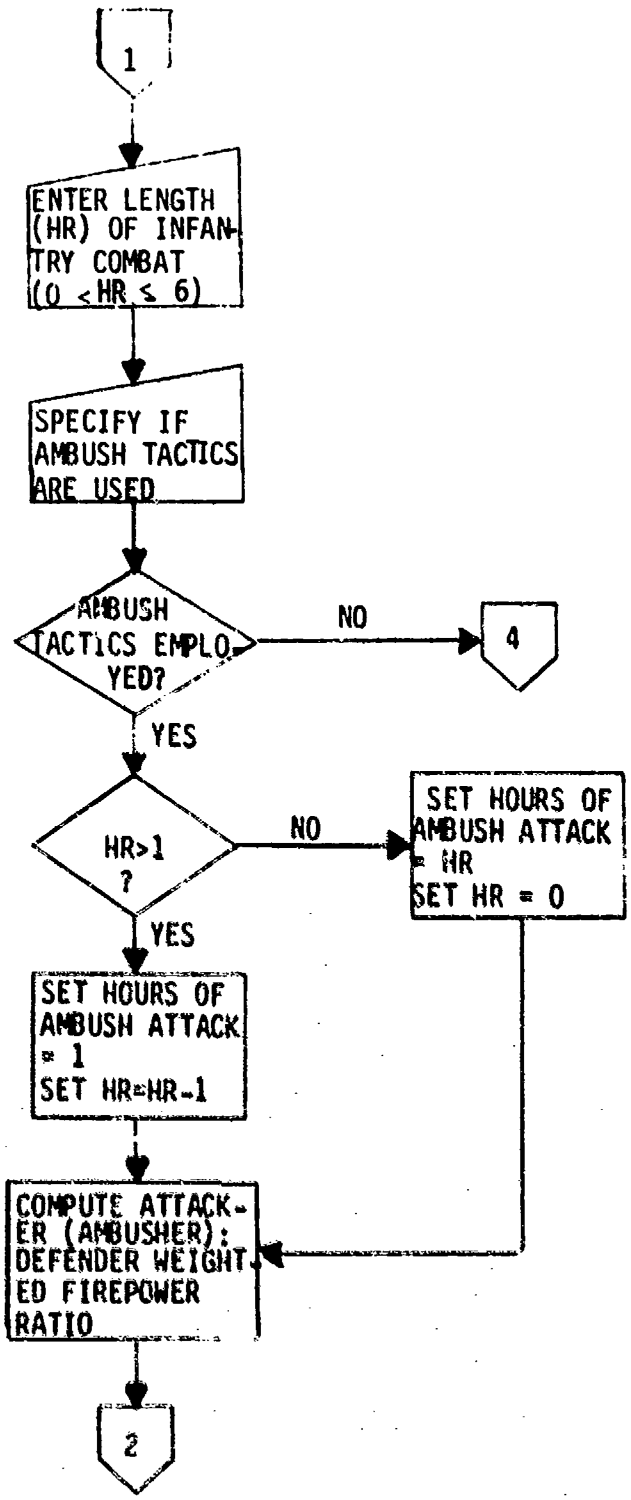

Figure 14. INFANT (OVLY 3) flow diagram (continued).

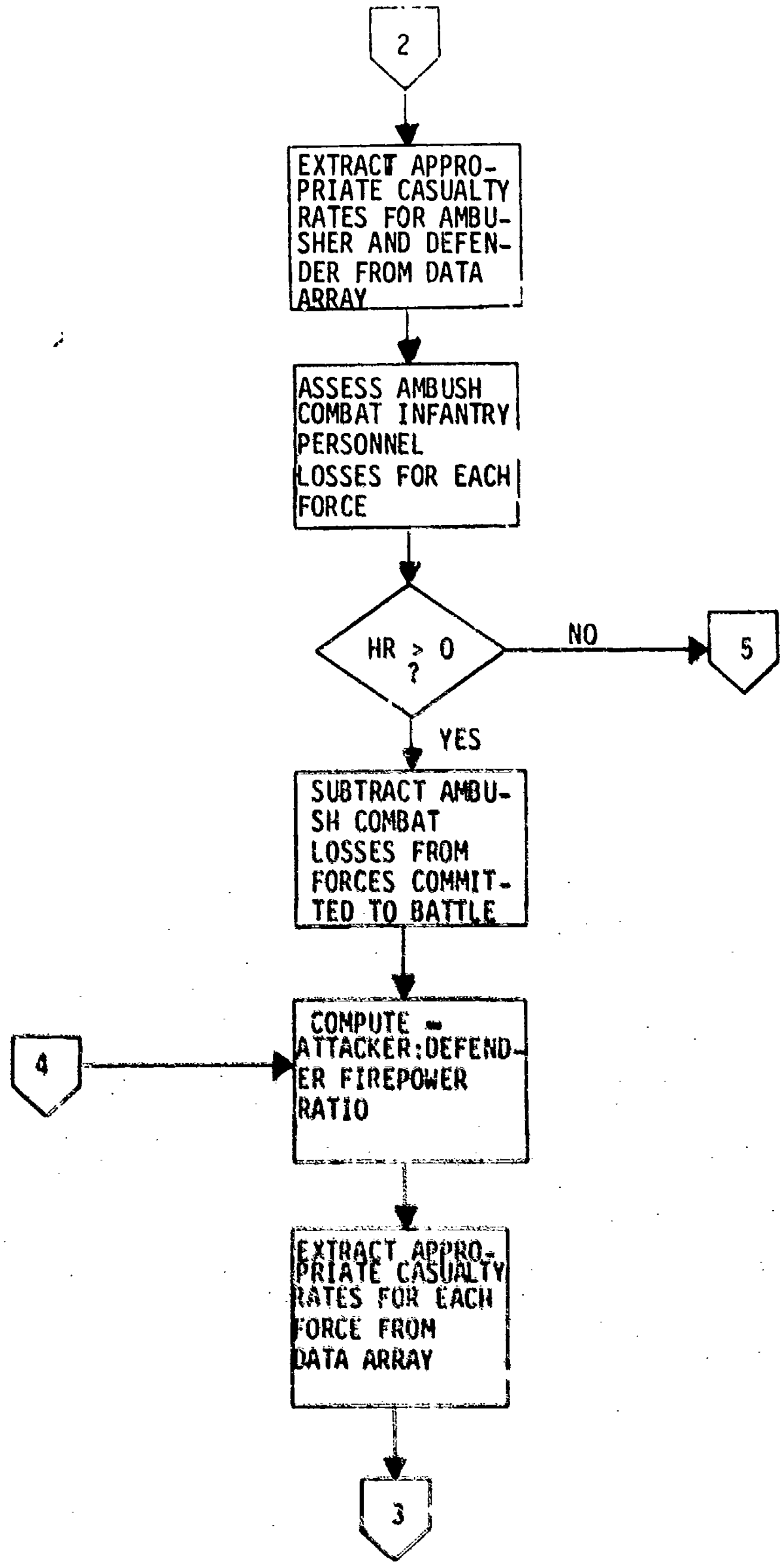

Figure 14. INFANT (OVLY 3) flow diagram (continued).

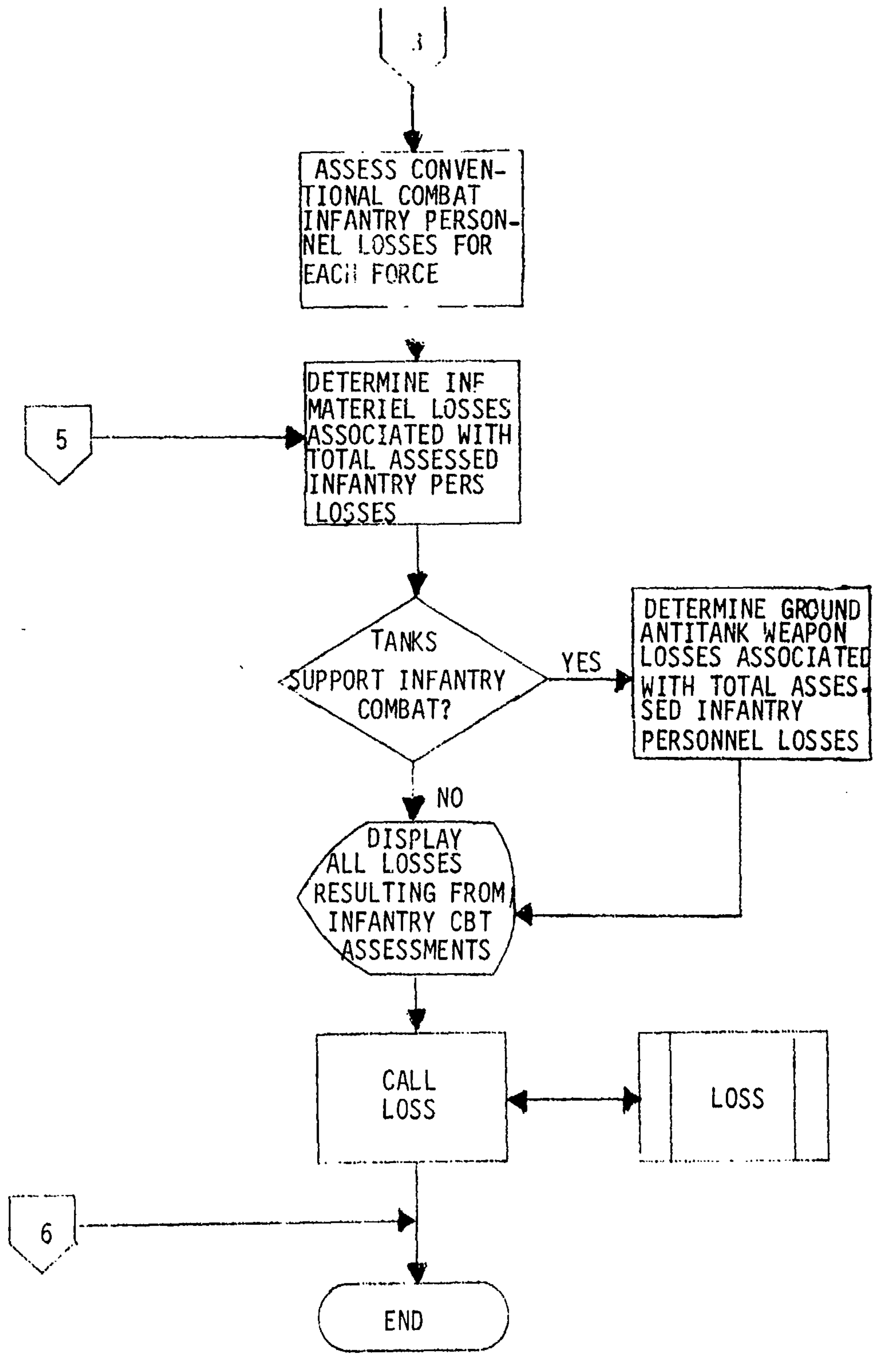

Figure 14. INFANT (OVLY 3) flow diagram (concluded).

one-time assessment of ambush and/or conventional dismounted infantry combat casualties suffered by each force. Following the display of the losses and processing of the LOSS subroutine, control is returned to SUPER. Program variables for OVLY 3 are listed in table I-1; the FORTRAN source code is contained in figure I-1.

(5) OVERLAY 4. Program OVLY 4 is the second combat assessment routine accessed by the supervisory Jiffy Game program from DECISION POINT number 3 (see table 1 and figure 7). This overlay consists of a main program (MINE) and a subroutine (FASCAM), which contain the assessment logic for attrition due to minefields. The LOSS subroutine (see paragraph 4b(1)(d)) is also called from the MINE program when all minefield assessments have been processed. Variable lists and FORTRAN source code listings for OVLY 4 are contained in appendix J.

(a) MINE. The primary function of the MINE program is to assess and display the losses suffered by the attacking force to minefields emplaced manually or mechanically (i.e., conventional minefields). MINE also contains the control point at which the type of minefield employed is specified interactively by the gamer. At the end of any minefield assessment, the program returns to this control point; thus, several assessments can be processed employing the same or different types of minefields before control is returned to the supervisory program. The logic flow diagram for MINE is given in figure 15. Only a minimal amount of data is needed to assess minefield losses; most of the necessary parameters are set interactively by gamer inputs. The processing of assessments is terminated from the control point, after which the LOSS subroutine is called and the overlay exited. The FORTRAN source code is given in figure J-1, and the program variables are listed in table J-1.

(b) FASCAM. This subroutine of OVLY 4 contains the logic used to assess losses to minefields composed of scatterable mines (FASCAM). The subroutine is called from the main overlay program (MINE) whenever the gamer specifies that a FASCAM minefield assessment is being processed. The logic flow diagram of FASCAM is given in figure 16. Although the assessment computation logic is essentially the same as for conventional minefields, the FASCAM minefields require a different set of inputs and casualty rate data. The FORTRAN source code for FASCAM is given in figure J-2, and the program variable list is given in table J-2.

(6) OVERLAY 5. Program OVLY 5 (AHAD) is the last of the combat assessment routines called from the supervisory program (SUPER) at DECISION POINT number 3 (see table 1 and figure 7). The purpose of this program is to determine and display losses resulting from combat involving attack helicopters and air defense systems. The overlay contains no subroutines; the INDEX 5 function (see paragraph 4b(1)(c)) is utilized in extracting helicopter single shot kill probabilities, and subroutine LOSS (see paragraph 4b(1)(d)) is called after all assessments have been made. Both the helicopter and AD SSKP's are stored in the classified random access file (CLDATA); several unclassified

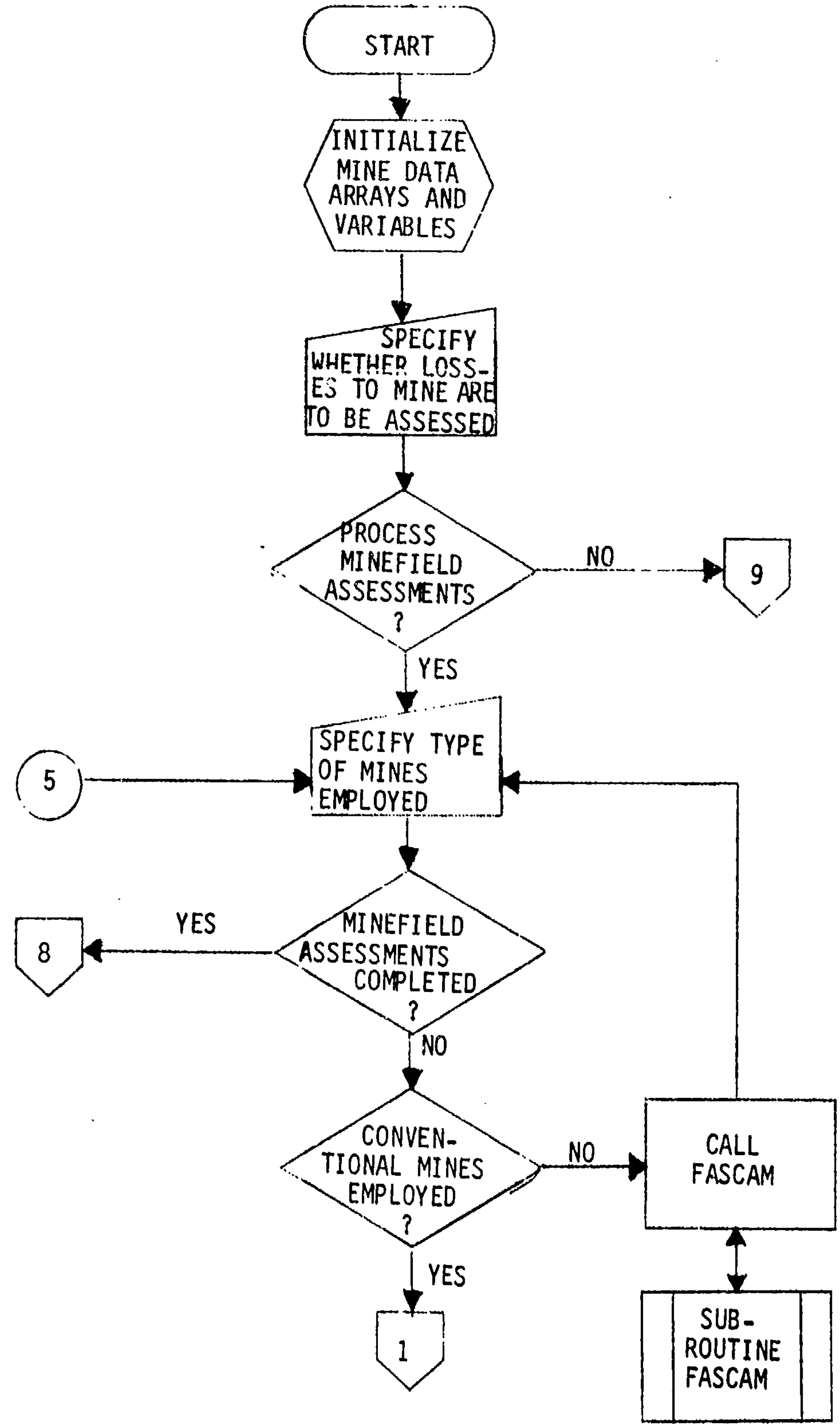

Figure 15. MINE flow diagram.
(Continued next page)

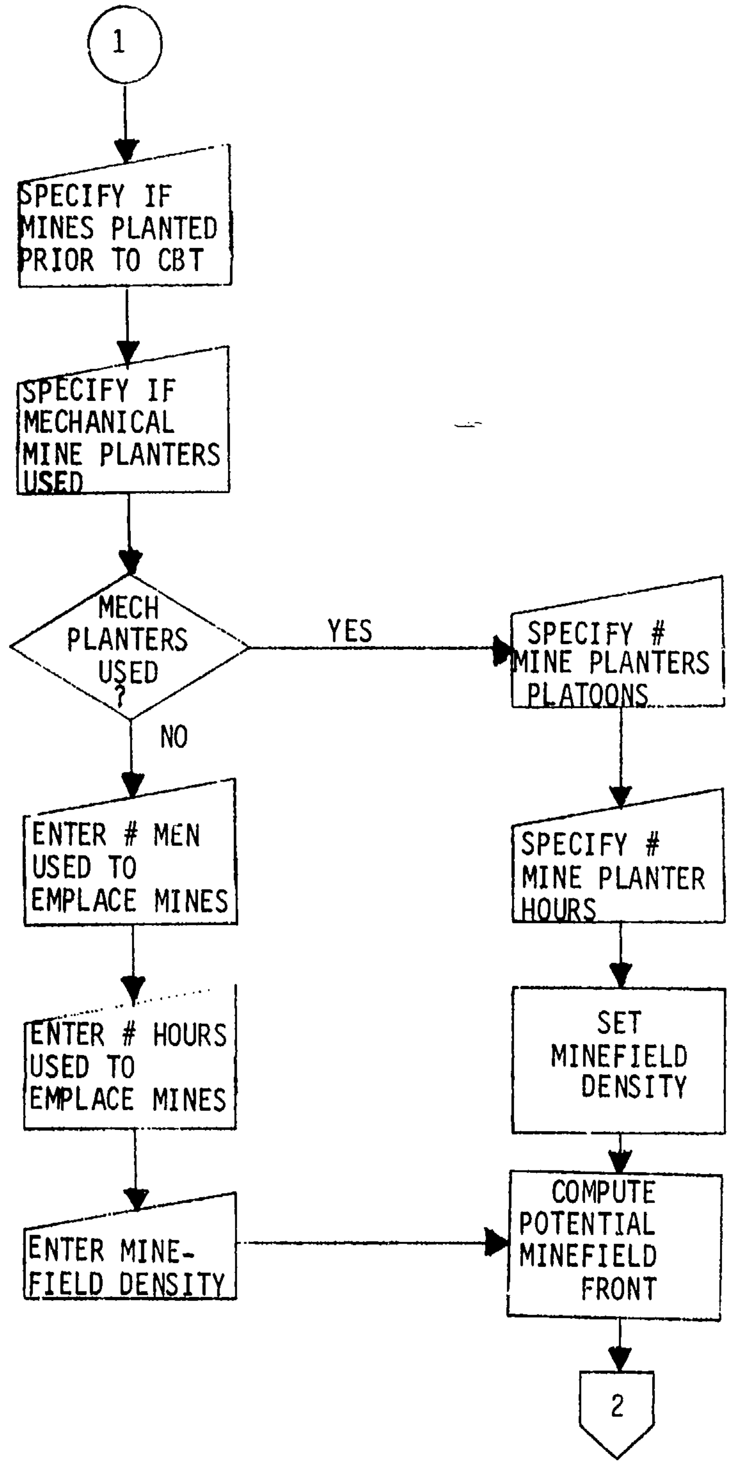

Figure 15. MINE flow diagram (continued).

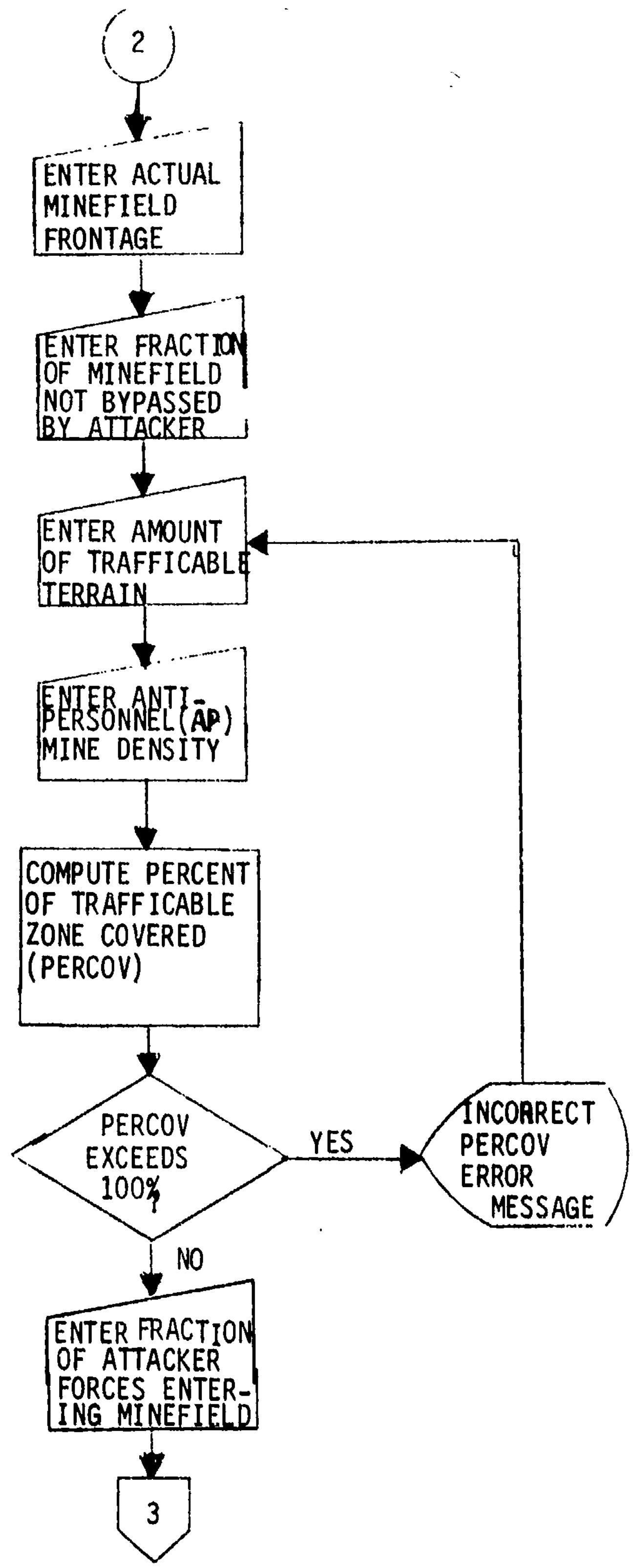

Figure 15. MINE flow diagram (continued).

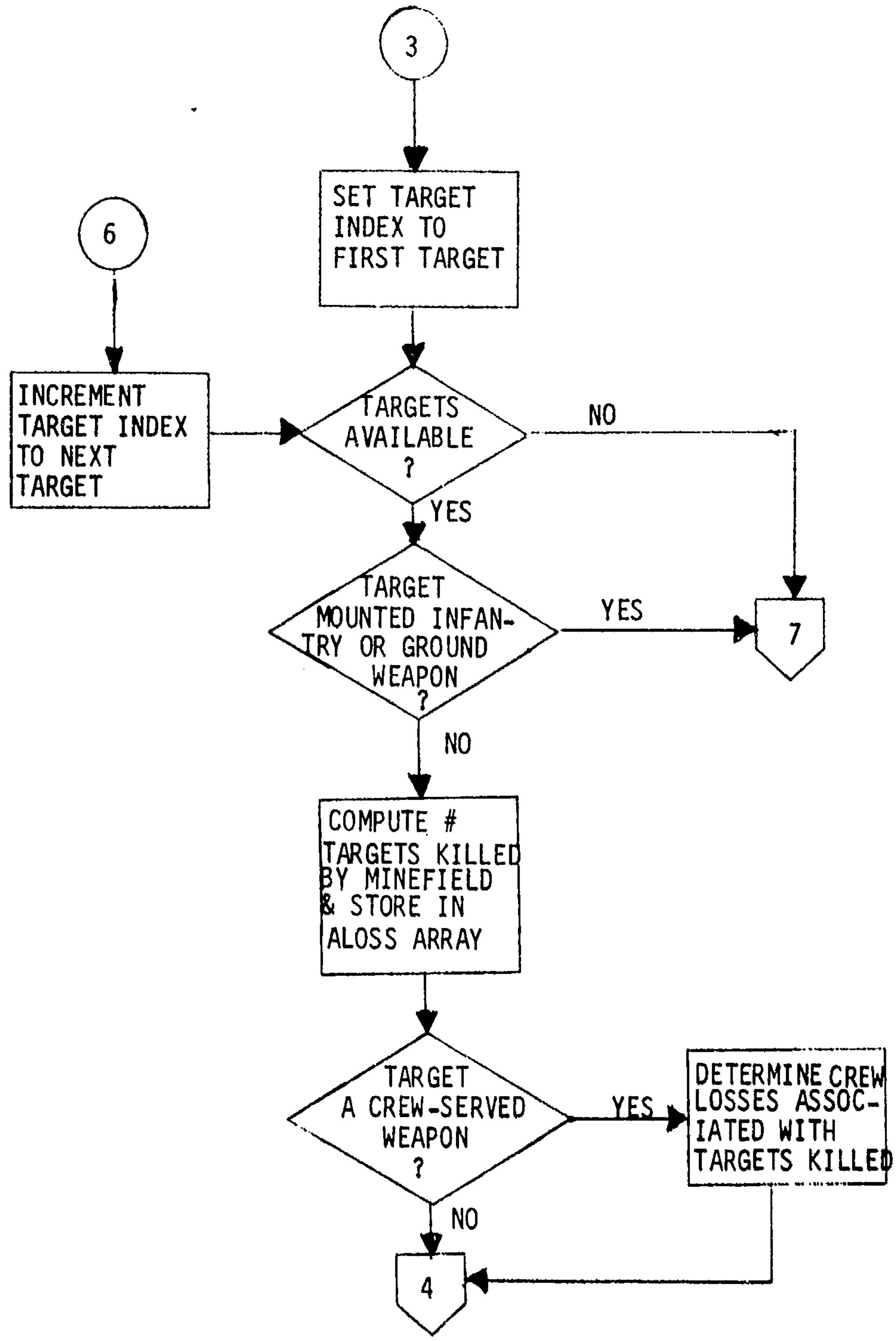

Figure 15. MINE flow diagram (continued).

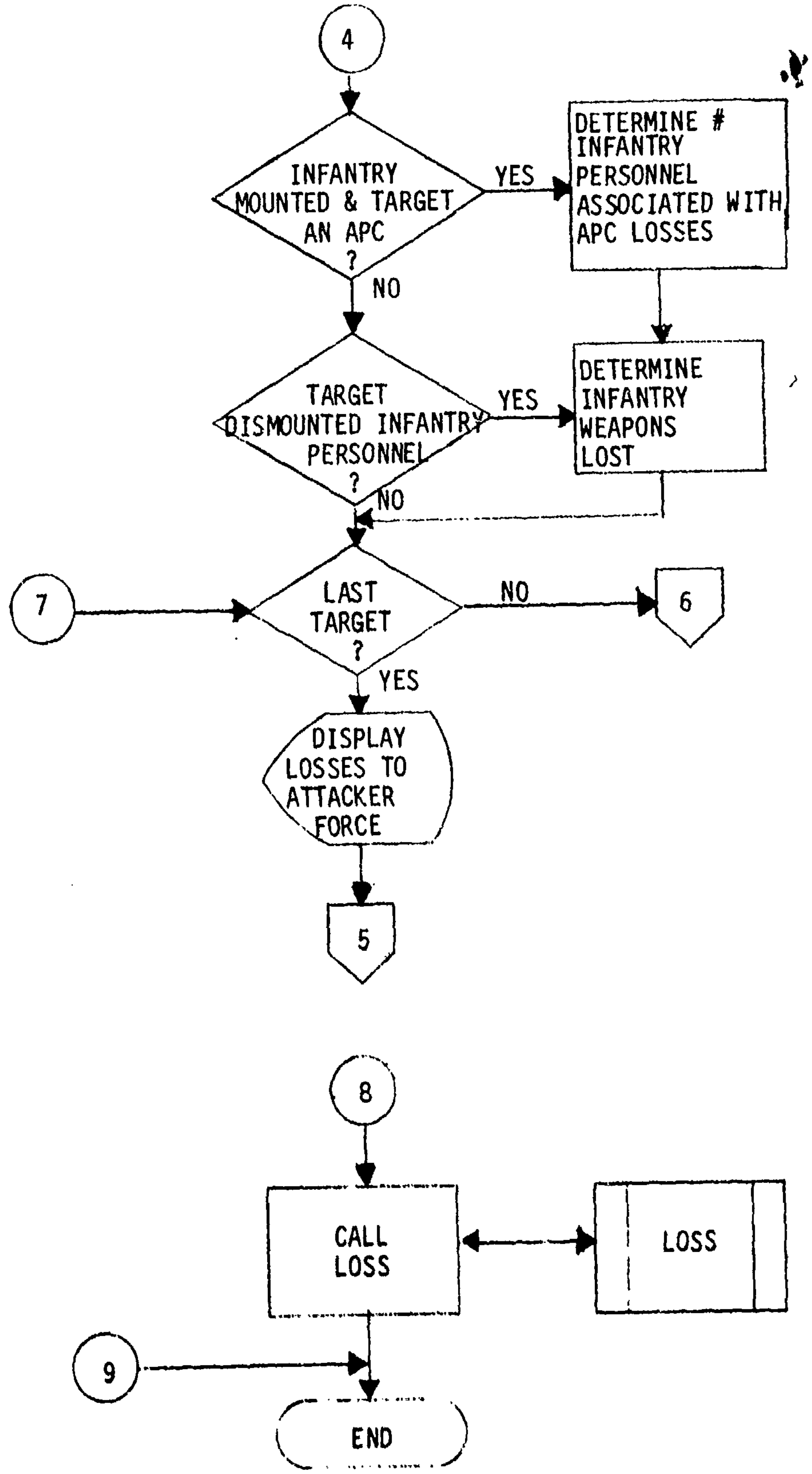

Figure 15. MINE flow diagram (concluded).

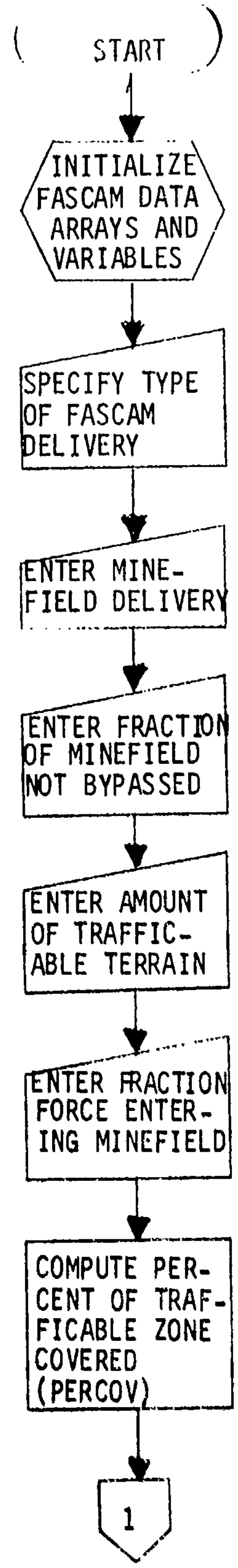

Figure 16. Subroutine FASCAM logic flow diagram.
(Continued next page)

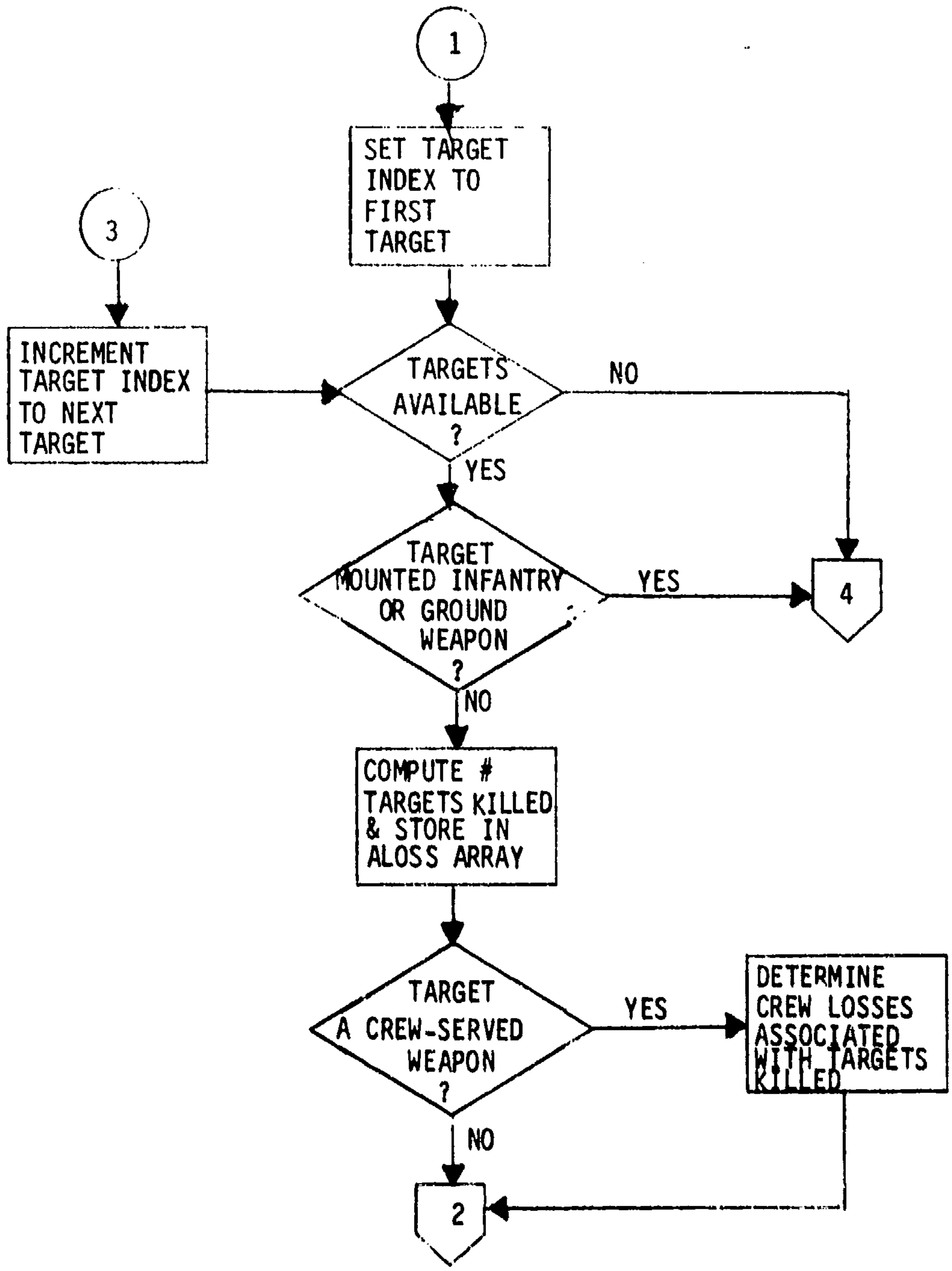

Figure 16. Subroutine FASCAM logic flow diagram (continued).

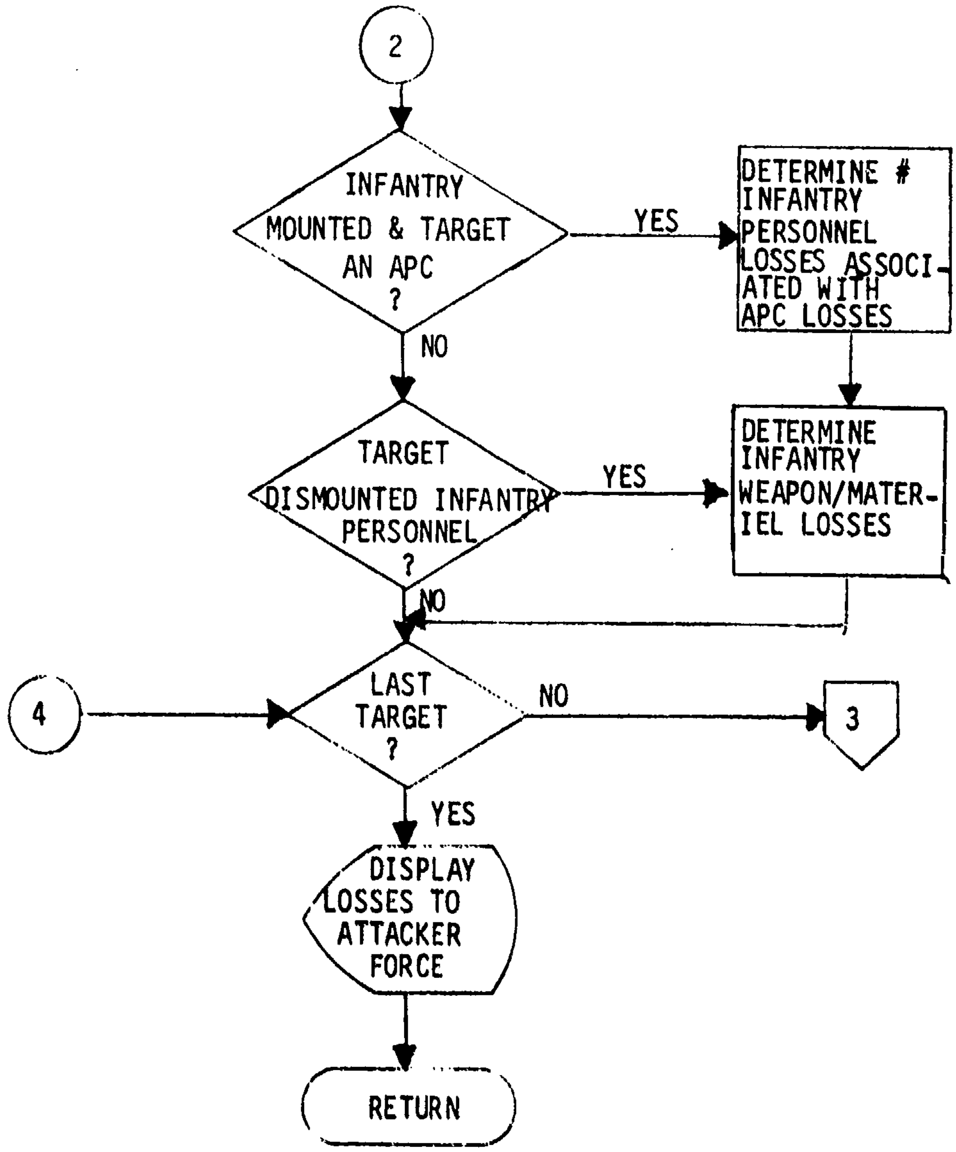

Figure 16. Subroutine FASCAM logic flow diagram (concluded).

data arrays are initiated in the program itself. The OVLY 5 logic flow diagram is given in figure 17. The program contains two sets of assessment logic, one for attack helicopter assessments against ground forces and another for air defense assessments against helicopters. The interactive definition of an attack helicopter mission initiates processing of both types of assessments, which are made for each pop-up of the helicopters in the attack cell and consequently may be cycled through several times for each mission. The number of helicopter missions to be assessed for each force is determined by the gamer; the Red helicopter/Blue air defense assessments are completed prior to beginning the Blue helicopter/Red air defense assessments. When all assessments have been completed, the cumulative losses are displayed for both forces, the LOSS subroutine is called, and the overlay exited. The OVLY 5 program variables are listed in table K-1, and the FORTRAN source code is given in figure K-1.

(7) OVERLAY 6. The overlay, OVLY 6, is the first combat assessment routine called by the supervisory program (SUPER) from DECISION POINT number 3 (see table 1 and figure 7). The overlay consists of the main program (CANNON) and one subroutine (CLGP); the function of OVLY 6 is to assess losses due to indirect fire weapon systems. The subroutine LOSS (see paragraph 4b(1)(d)) is also called when all assessments have been made. The routines require three data arrays from the classified random access file (CLDATA) in addition to the data initiated within the program itself. Appendix L contains FORTRAN source codes and program variable lists for OVLY 6.

(a) CANNON. The main program of overlay 6, CANNON, performs nearly all the assessments associated with mortar and field artillery fire and also displays the losses from all indirect fire missions. The logic flow diagram for CANNON is given in figure 18. The routine requires a number of gamer inputs to specify the types of indirect fire missions being assessed and to set parameters that are used in the actual assessment computations. The program cycles through several nested DO loops in making the loss calculations in order to assess all possible target/firer combinations; this is done for each force firing at the opposing force and for each phase of indirect fire combat being assessed. The only indirect fire assessment not included in the CANNON routine is for cannon-launched guided projectiles (CLGP). CLGP missions are available only to the Blue force and are assessed by calling the subroutine CLGP. The losses resulting from each of three major phases of indirect fire combat are displayed separately. When all assessments have been completed, the cumulative losses are displayed, the LOSS subroutine is called, and control is returned to the supervisory program. The FORTRAN source code for CANNON is given in figure L-1, and the program variables are listed in table L-1.

(b) CLGP. Subroutine CLGP is accessed from the indirect fire program to determine losses of Red weapons to Blue CLGP fire. The logic flow diagram for this subroutine is given in figure 19. The only gamer input required is the number of CLGP missions to be assessed; the computed losses

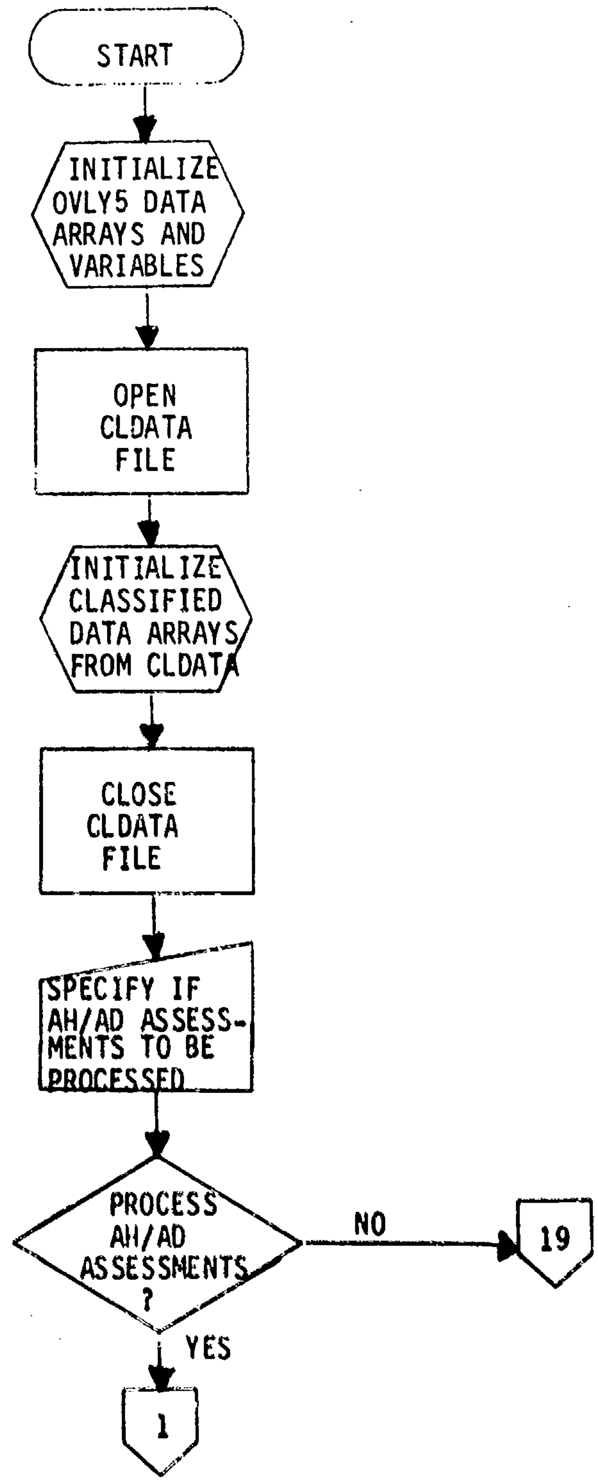

Figure 17. OVLY5 (AHAD) flow diagram.
(Continued next page)

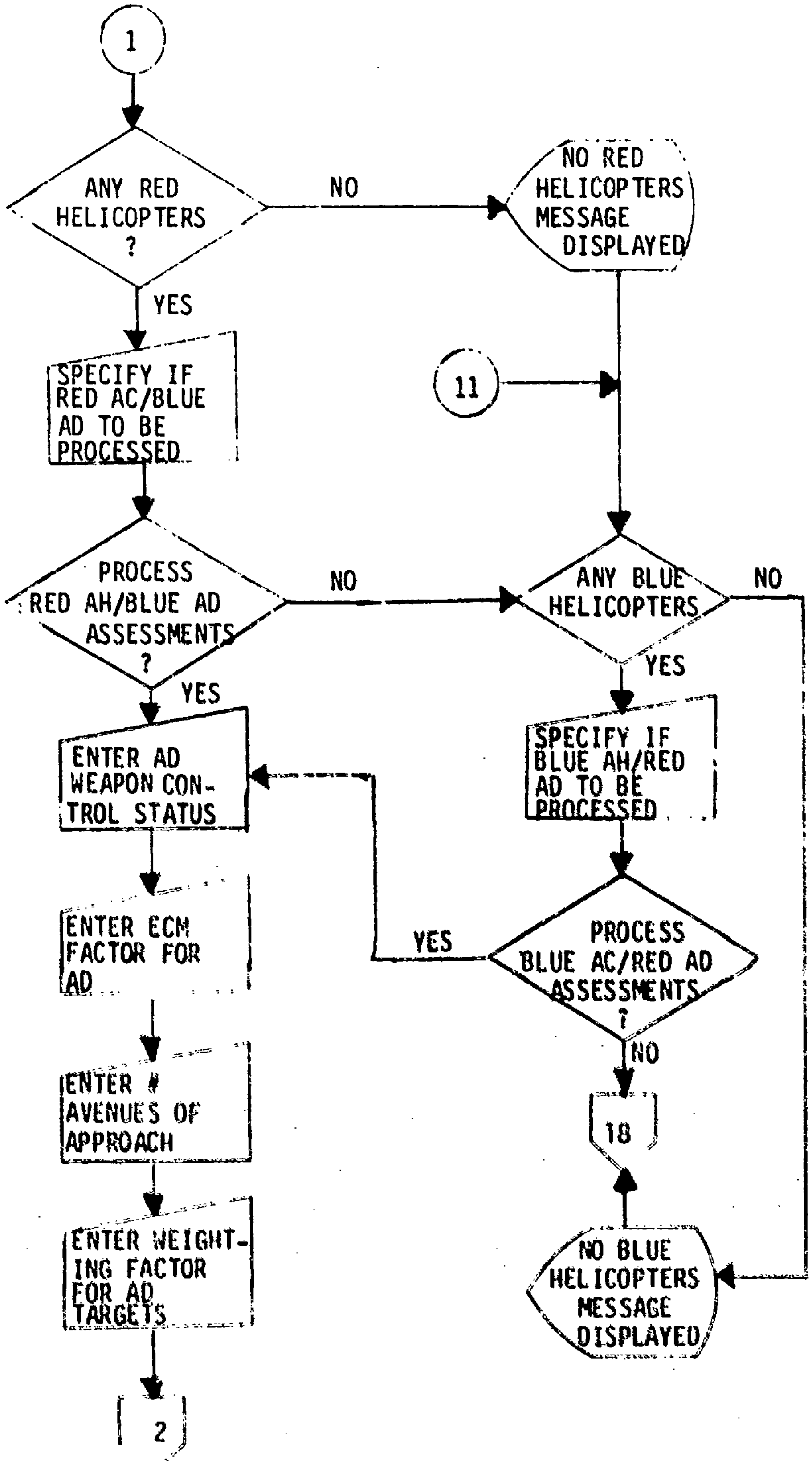

Figure 17. OVLY5 (AHAD) flow diagram (continued).

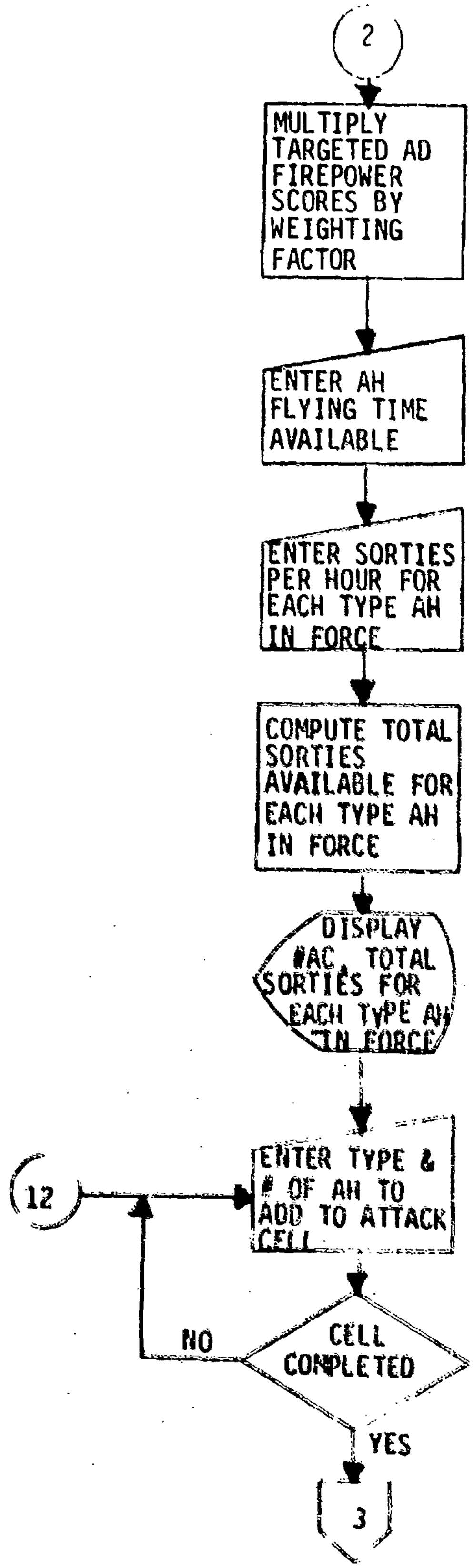

Figure 17. OVLY5 (AHAD) flow diagram (continued).

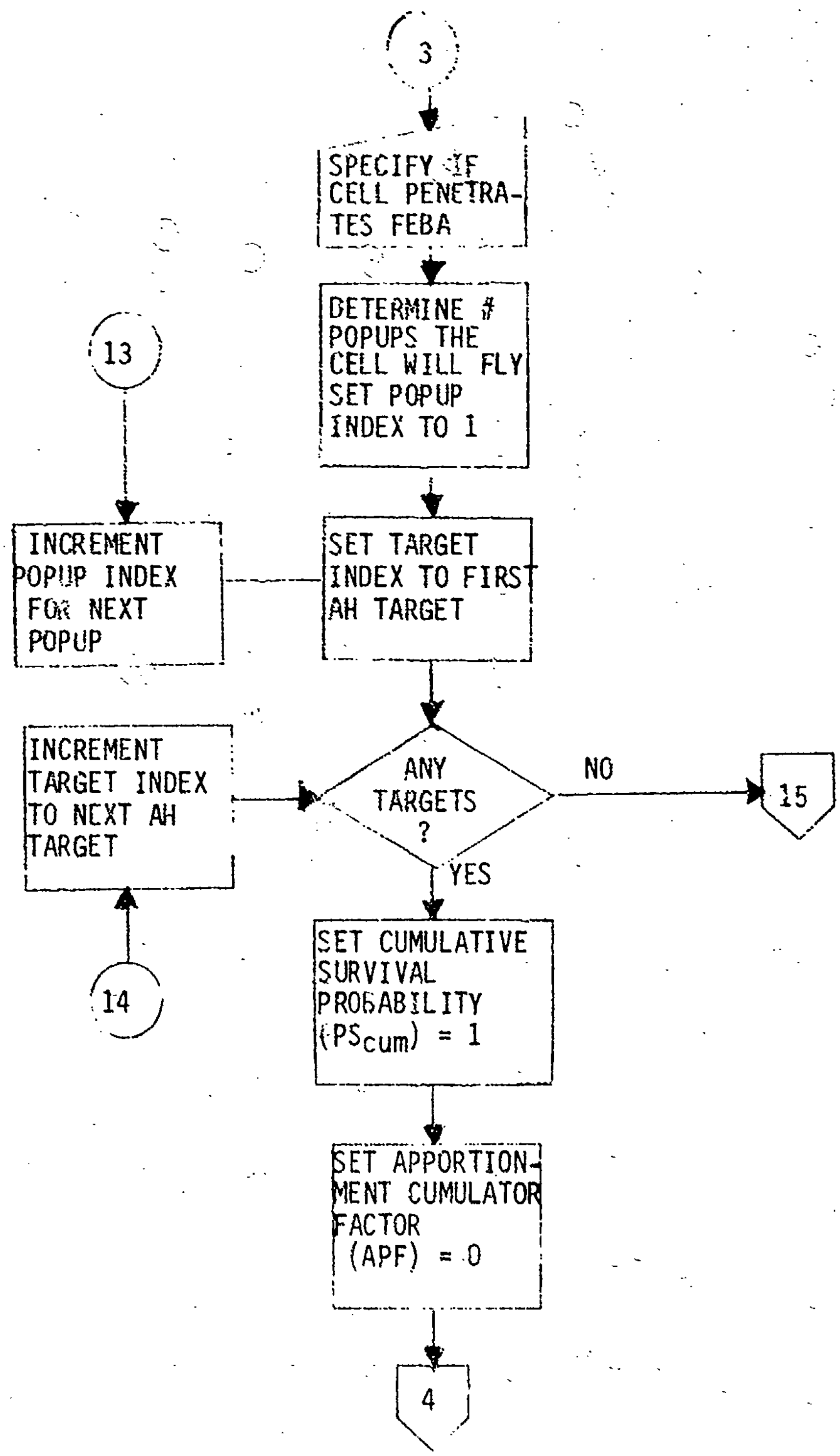

Figure 17. OVLY5 (AHAD) flow diagram (continued).

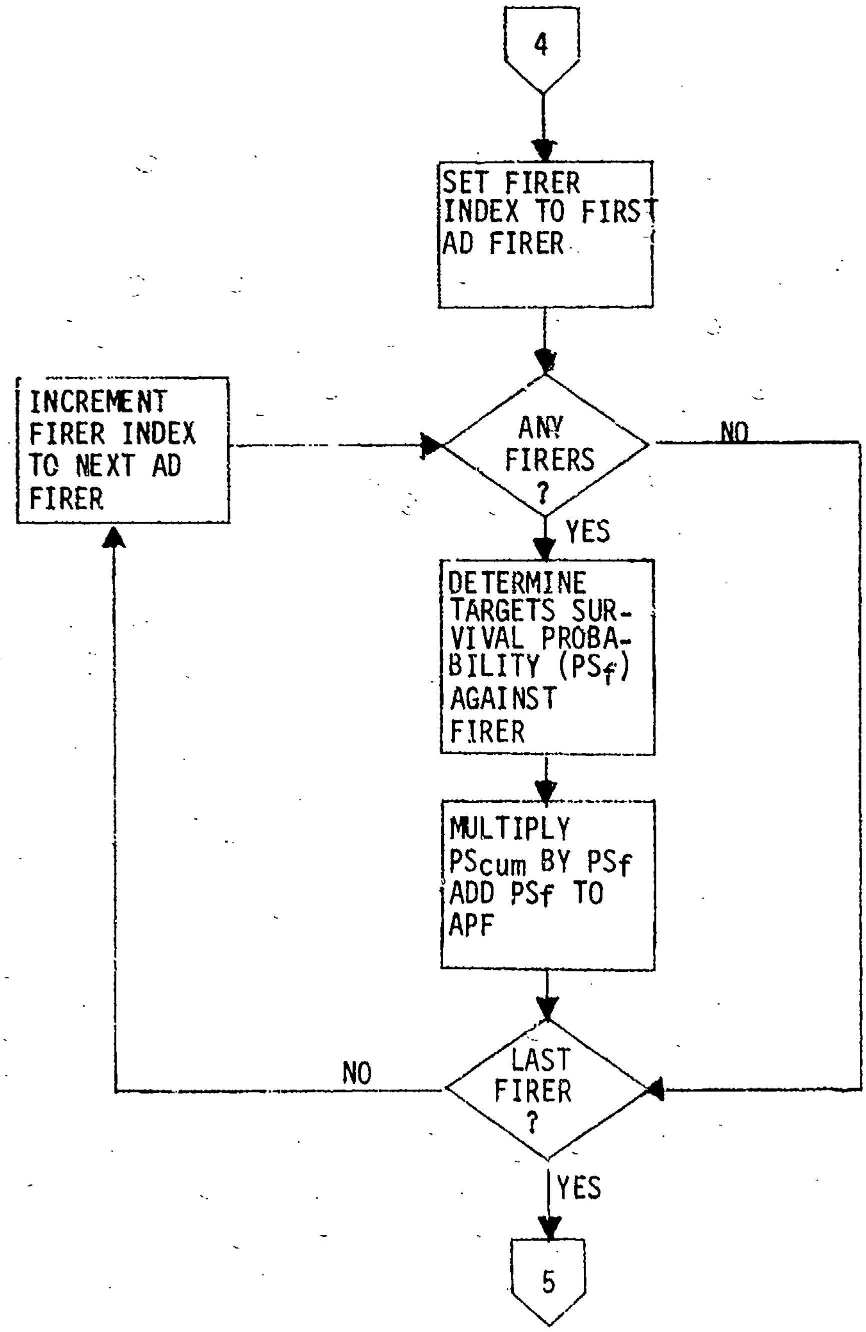

Figure 17. OVLY5 (AHAD) flow diagram (continued).

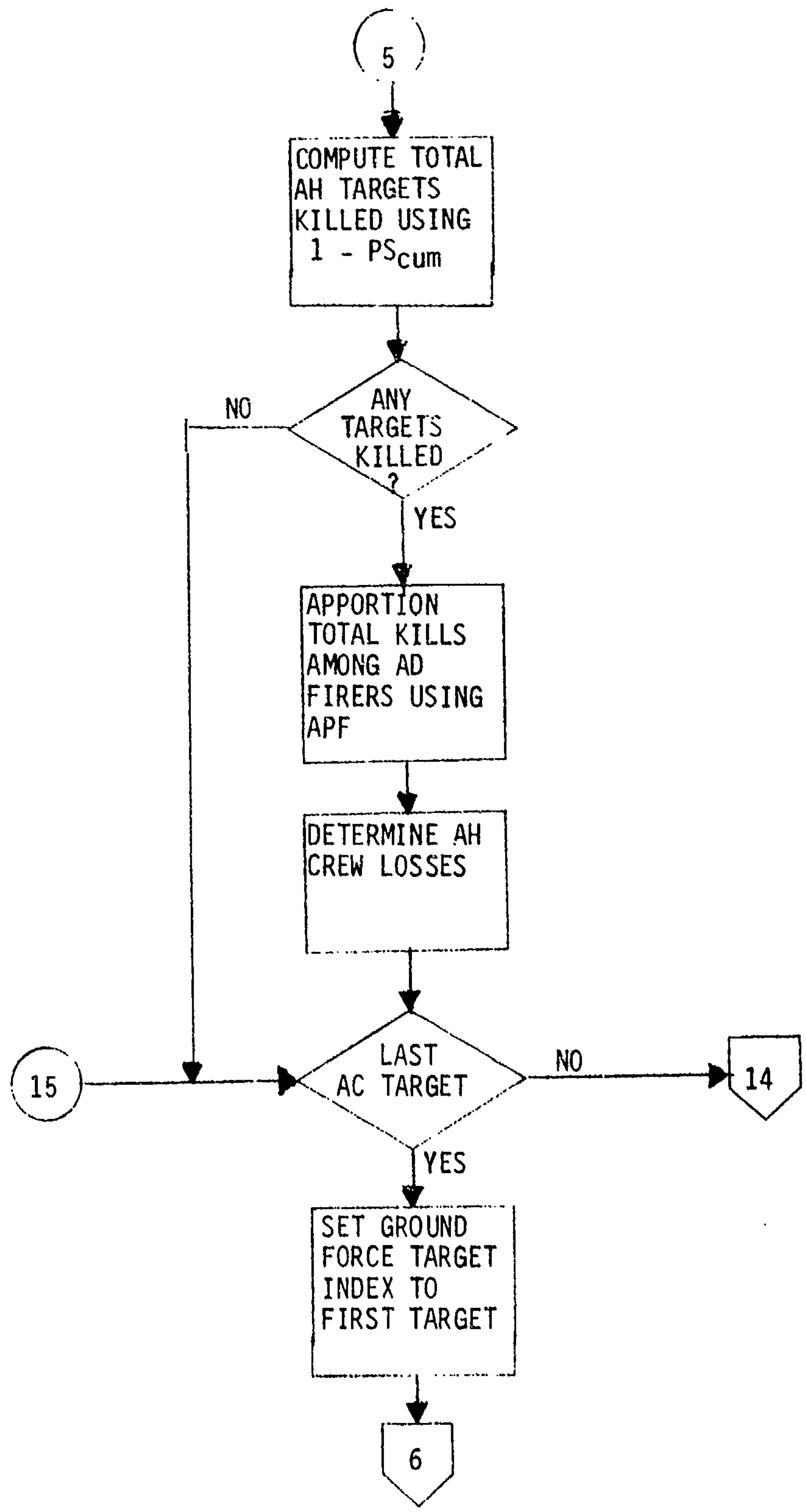

Figure 17. OVLY5 (AHAD) flow diagram (continued).

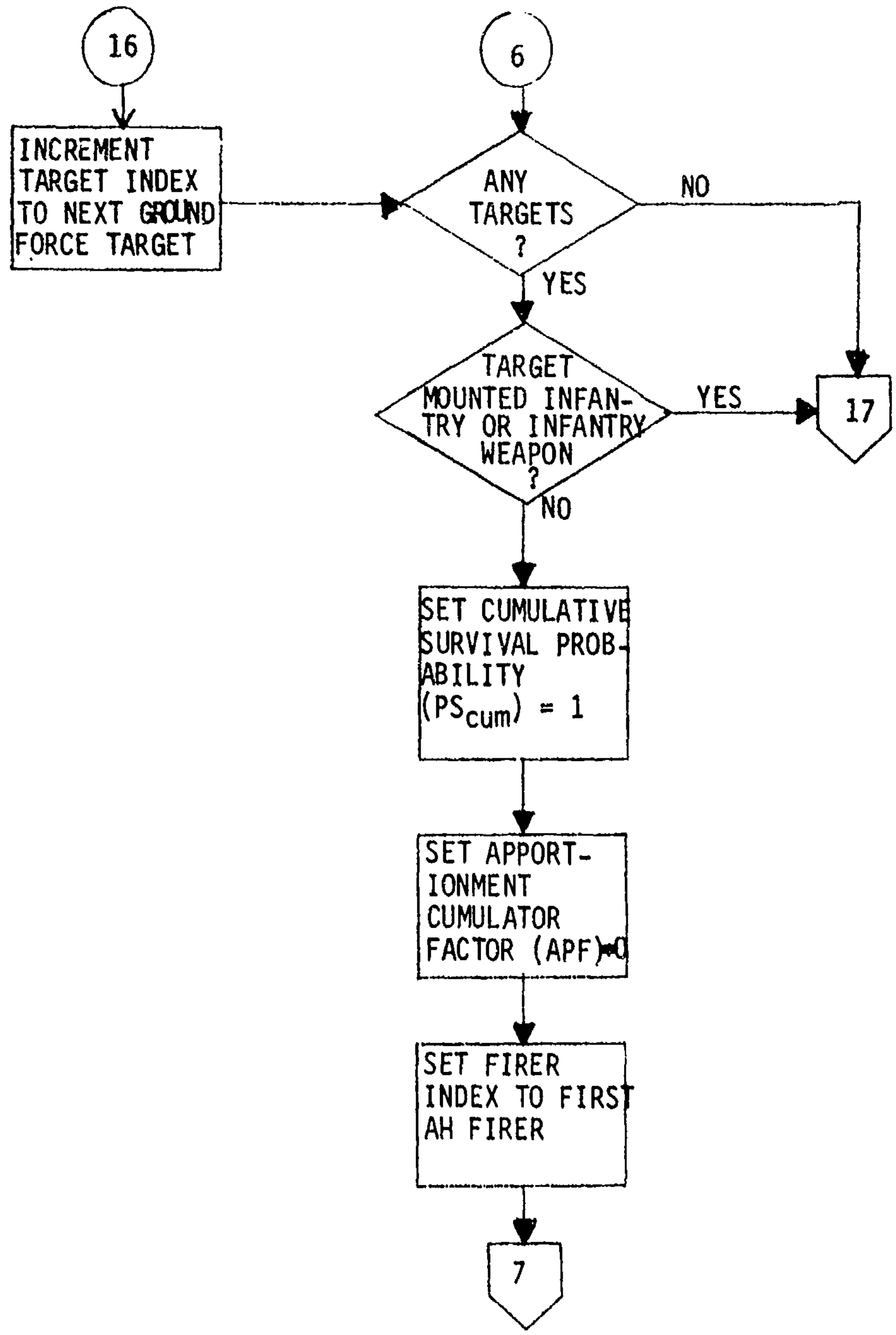

Figure 17. OVLY5 (AHAD) flow diagram (continued).

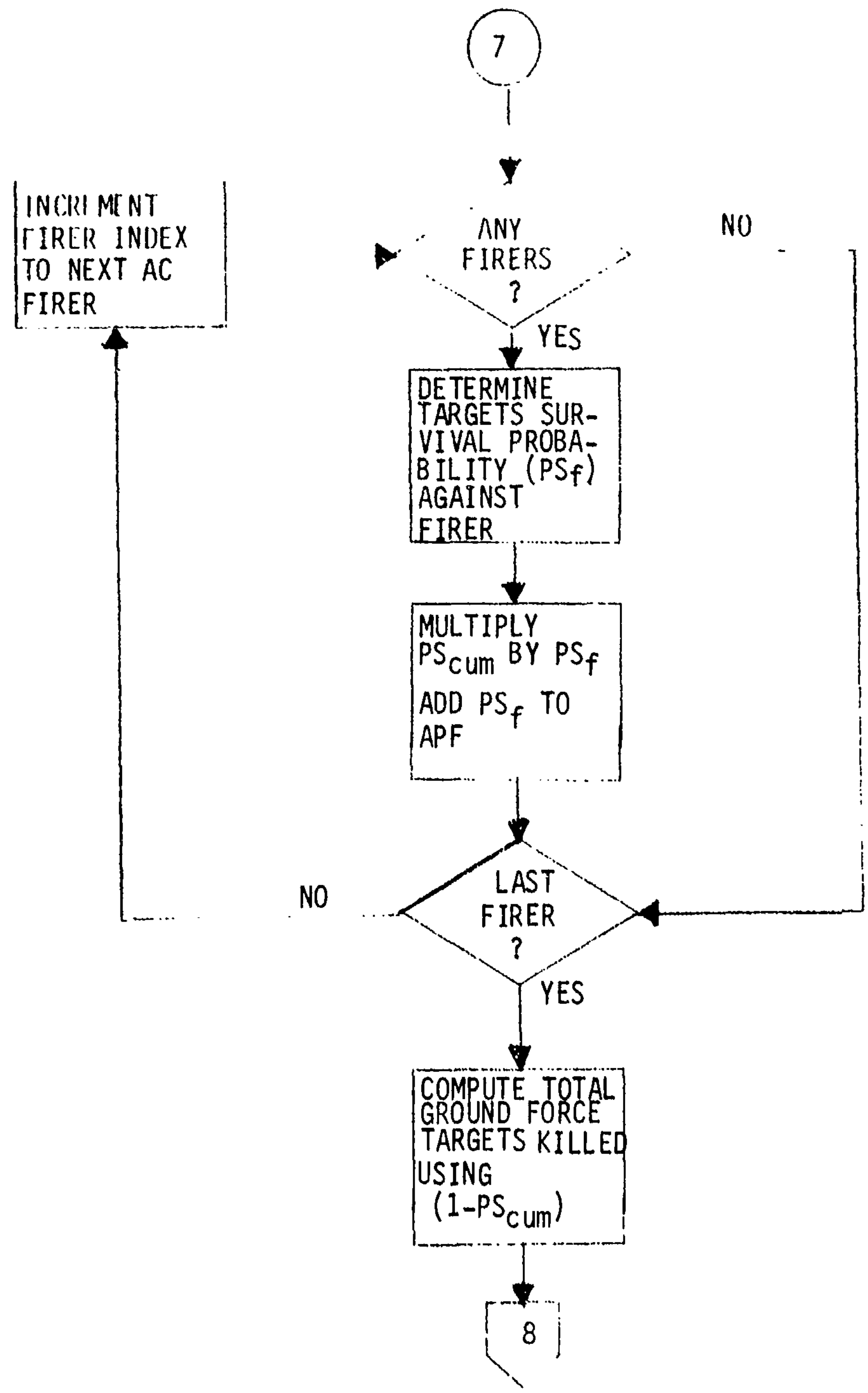

Figure 17. OVLY5 (AHAD) flow diagram (continued).

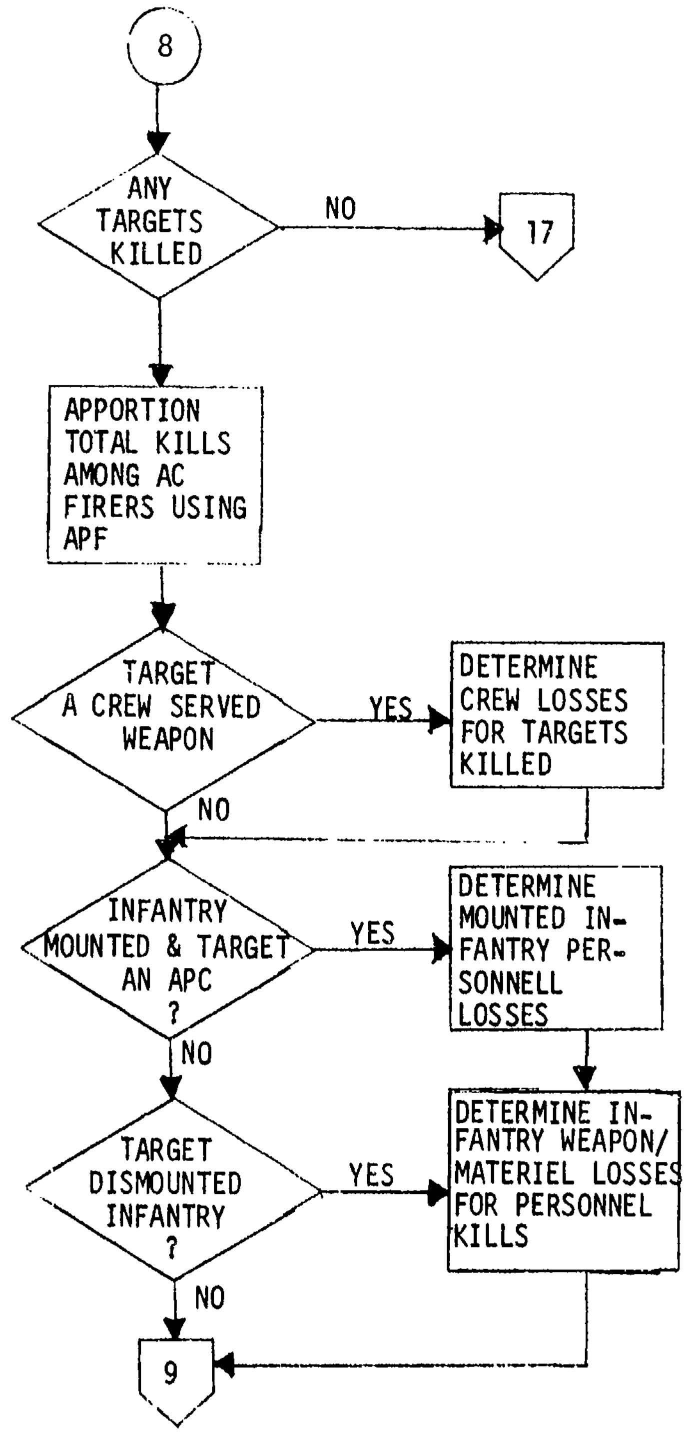

Figure 17. OVLY5 (AHAD) flow diagram (continued).

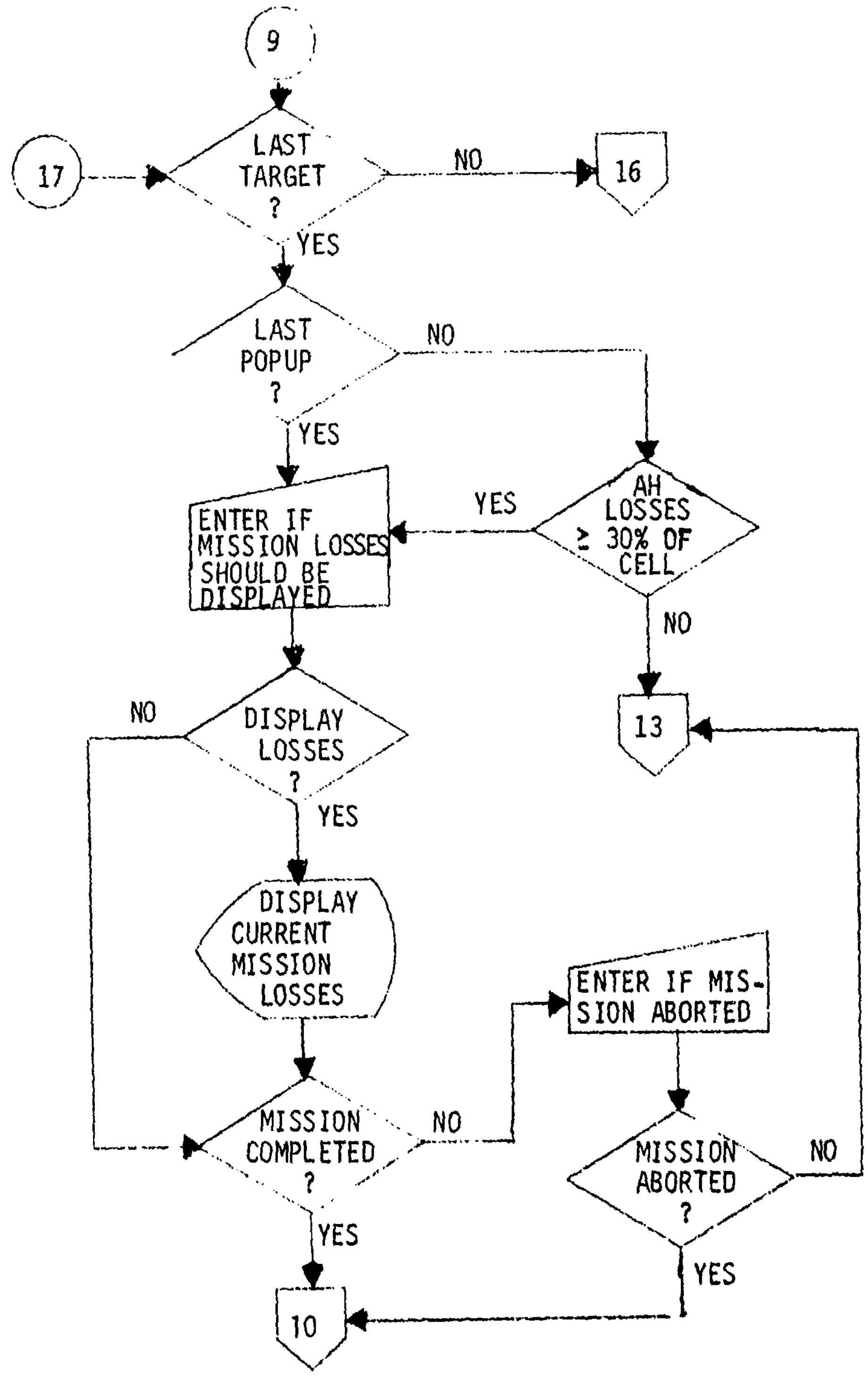

Figure 17. OVLY5 (AHAD) flow diagram (continued).

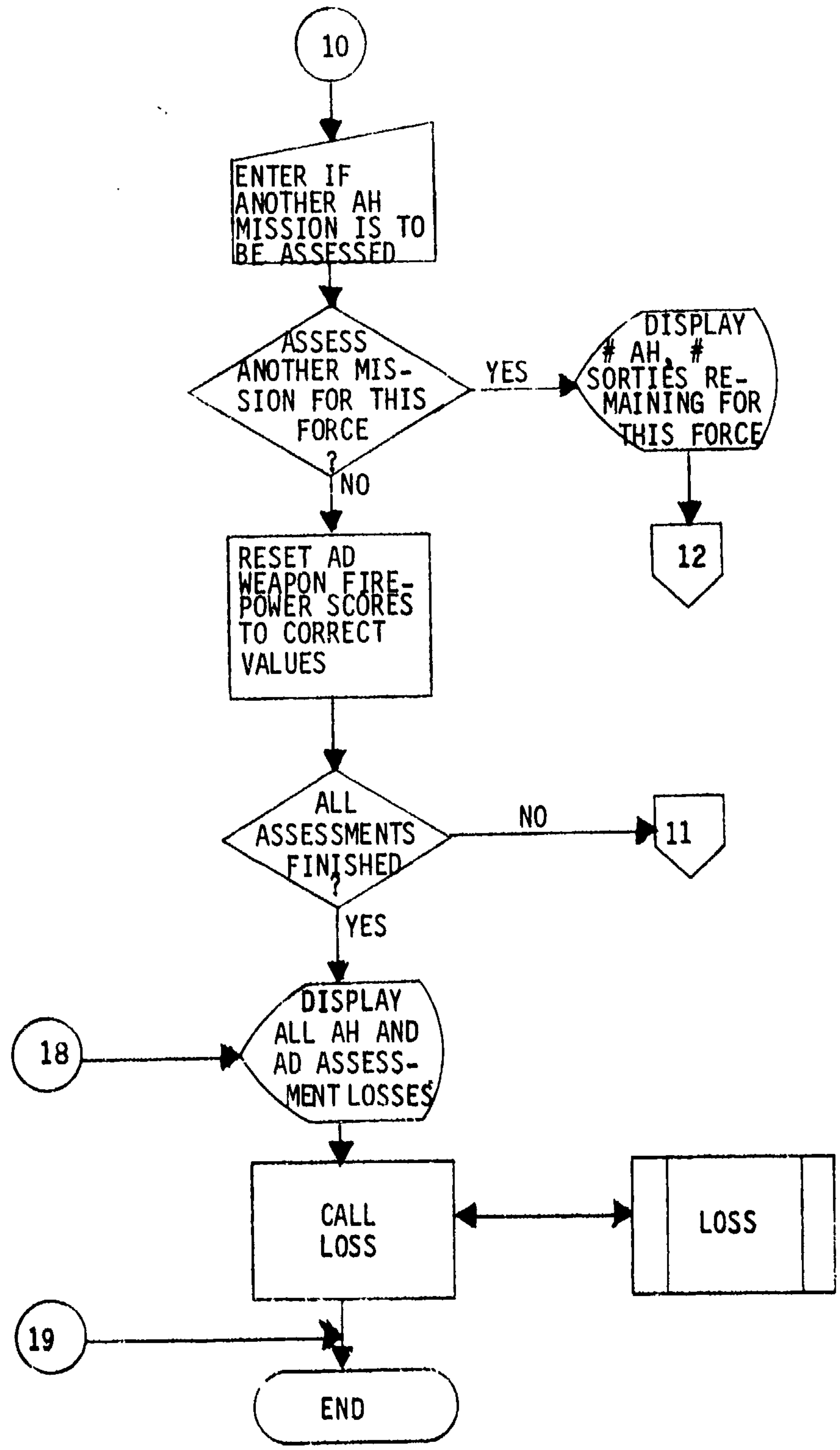

Figure 17. OVLY5 (AHAD) flow diagram (concluded).

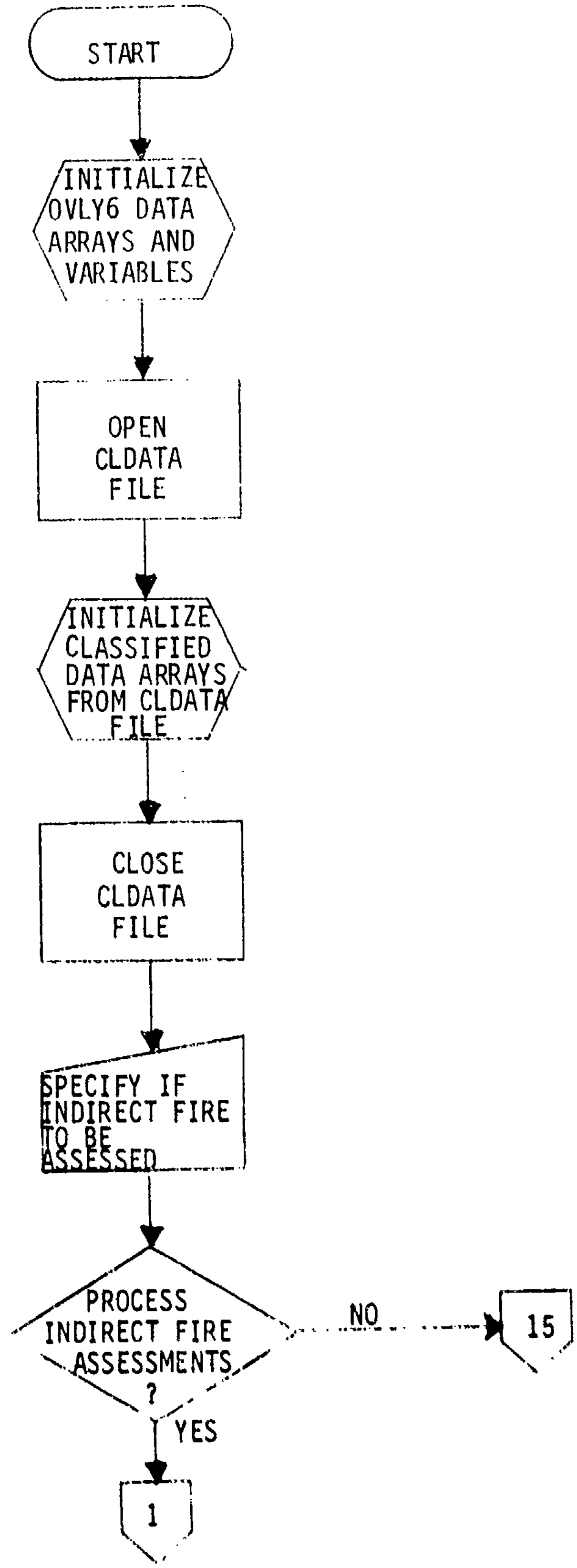

Figure 18. CANNON logic flow diagram.
(Continued next page)

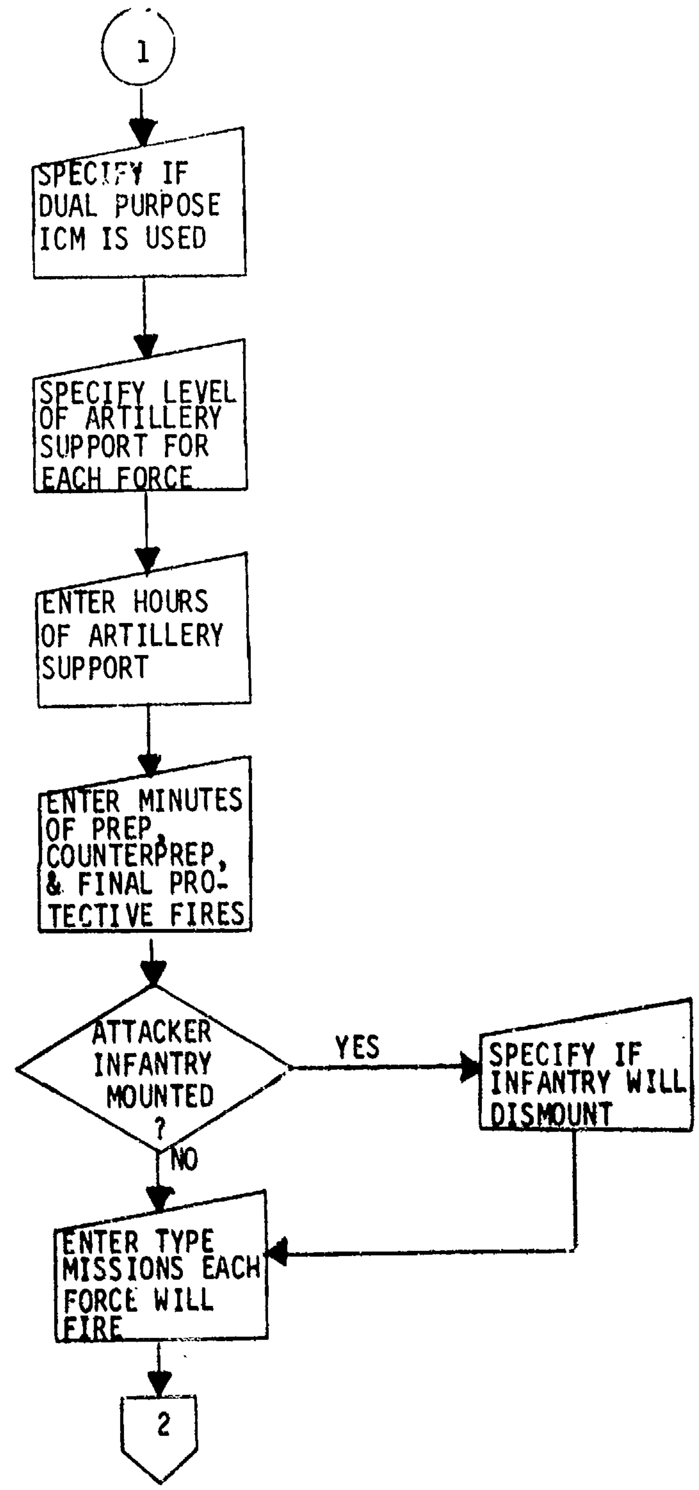

Figure 18. CANNON logic flow diagram (continued).

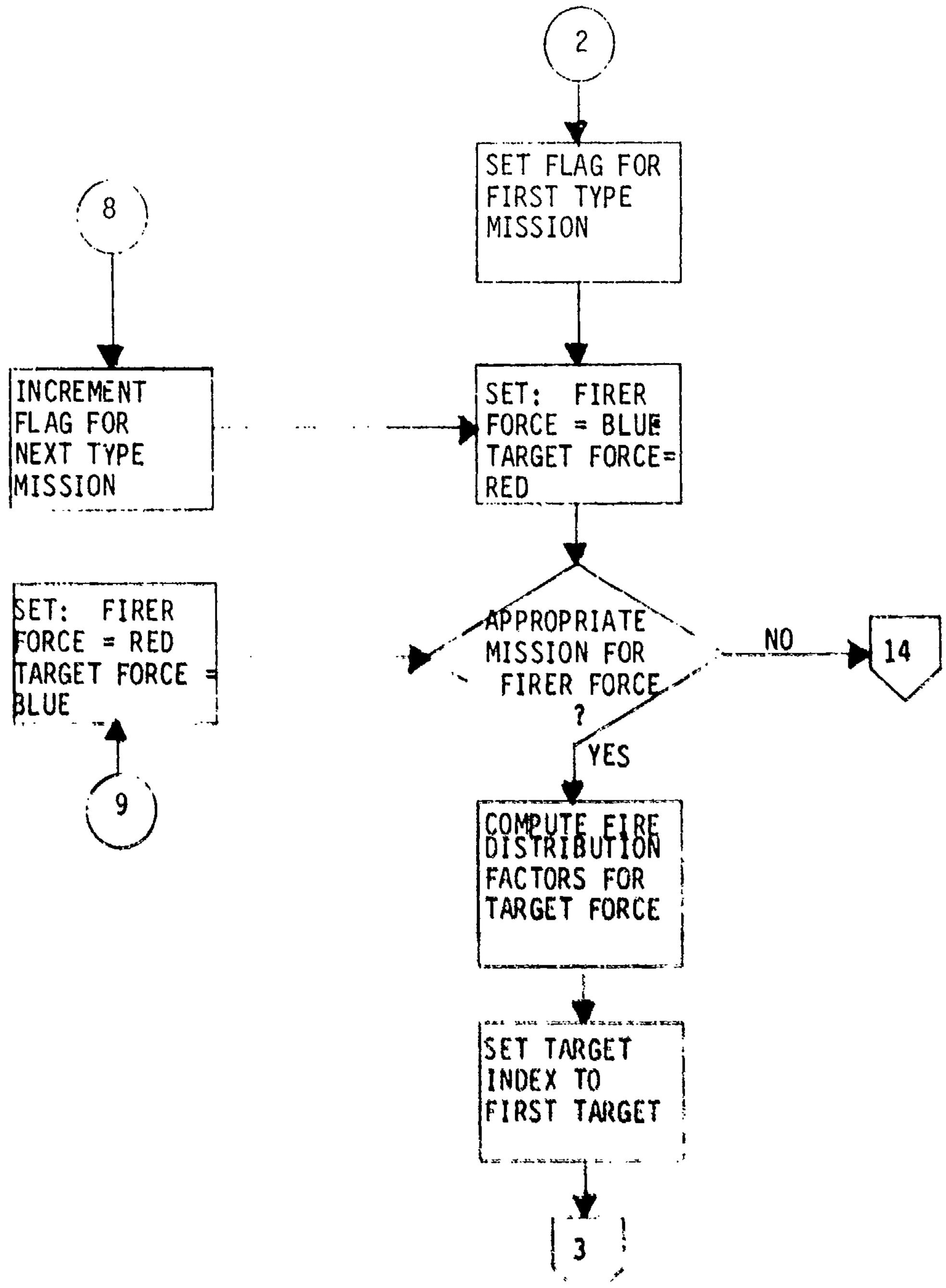

Figure 18. CANNON logic flow diagram (continued).

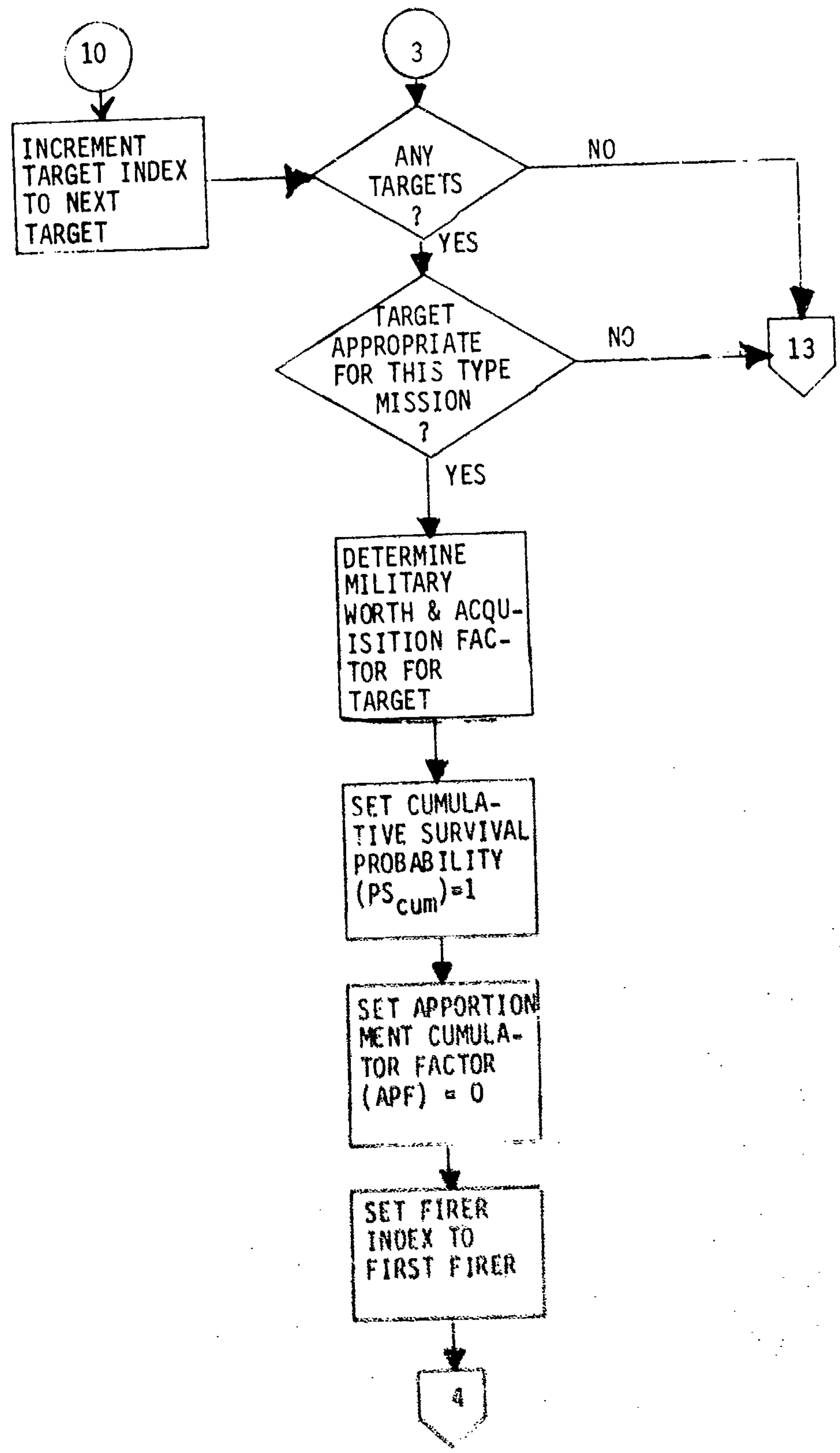

Figure 18. CANNON logic flow diagram (continued).

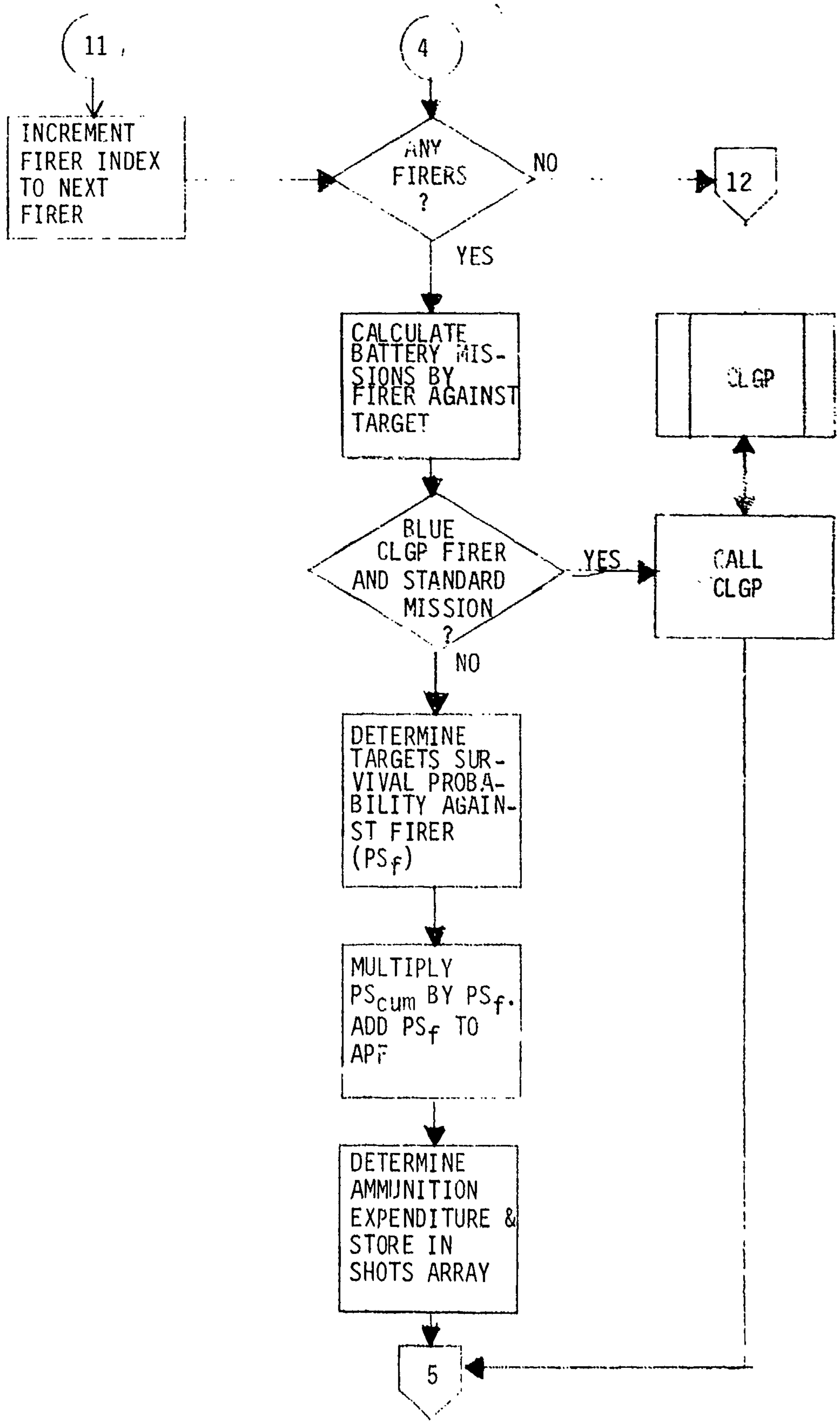

Figure 18. CANNON logic flow diagram (continued).

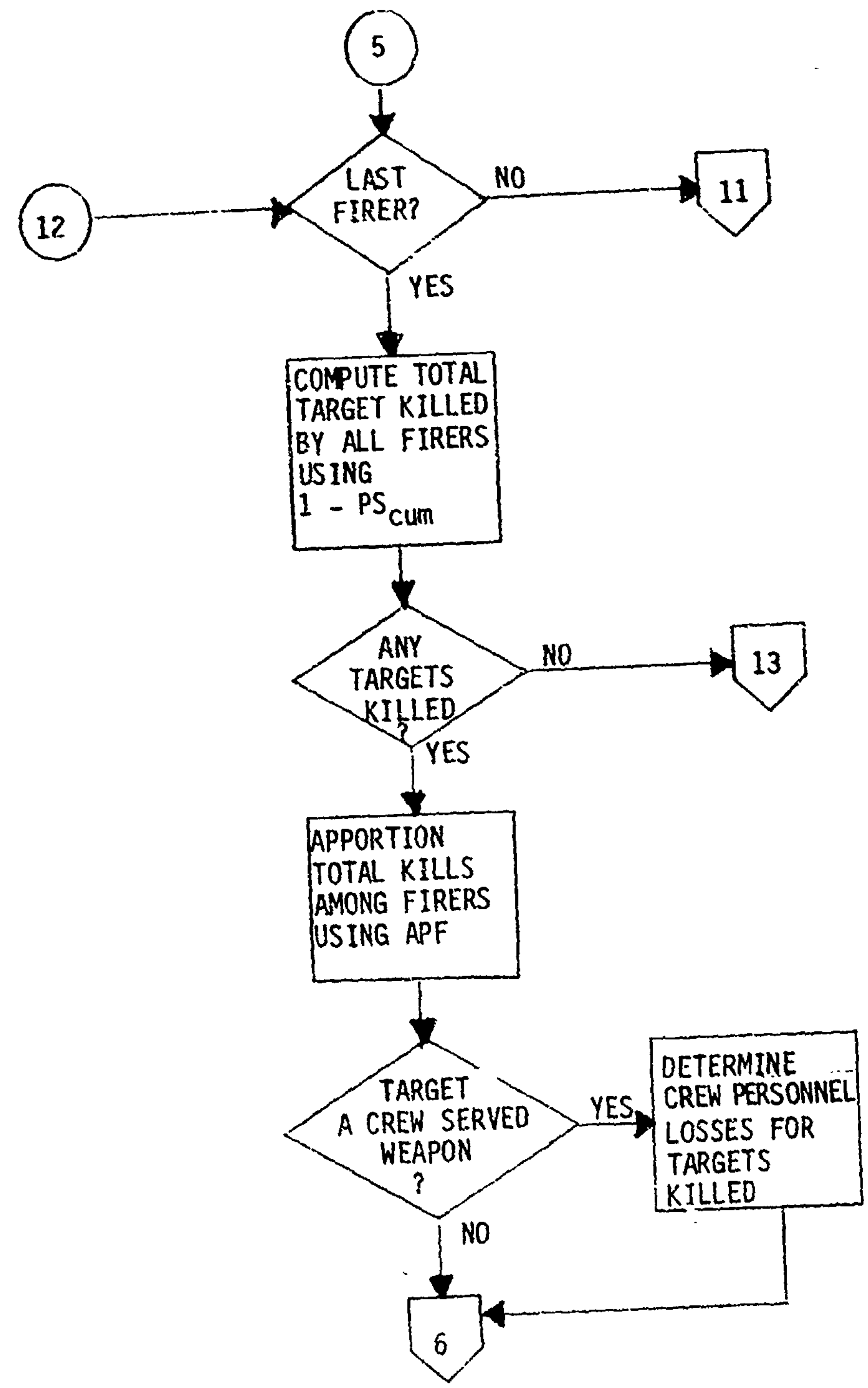

Figure 18. CANNON logic flow diagram (continued).

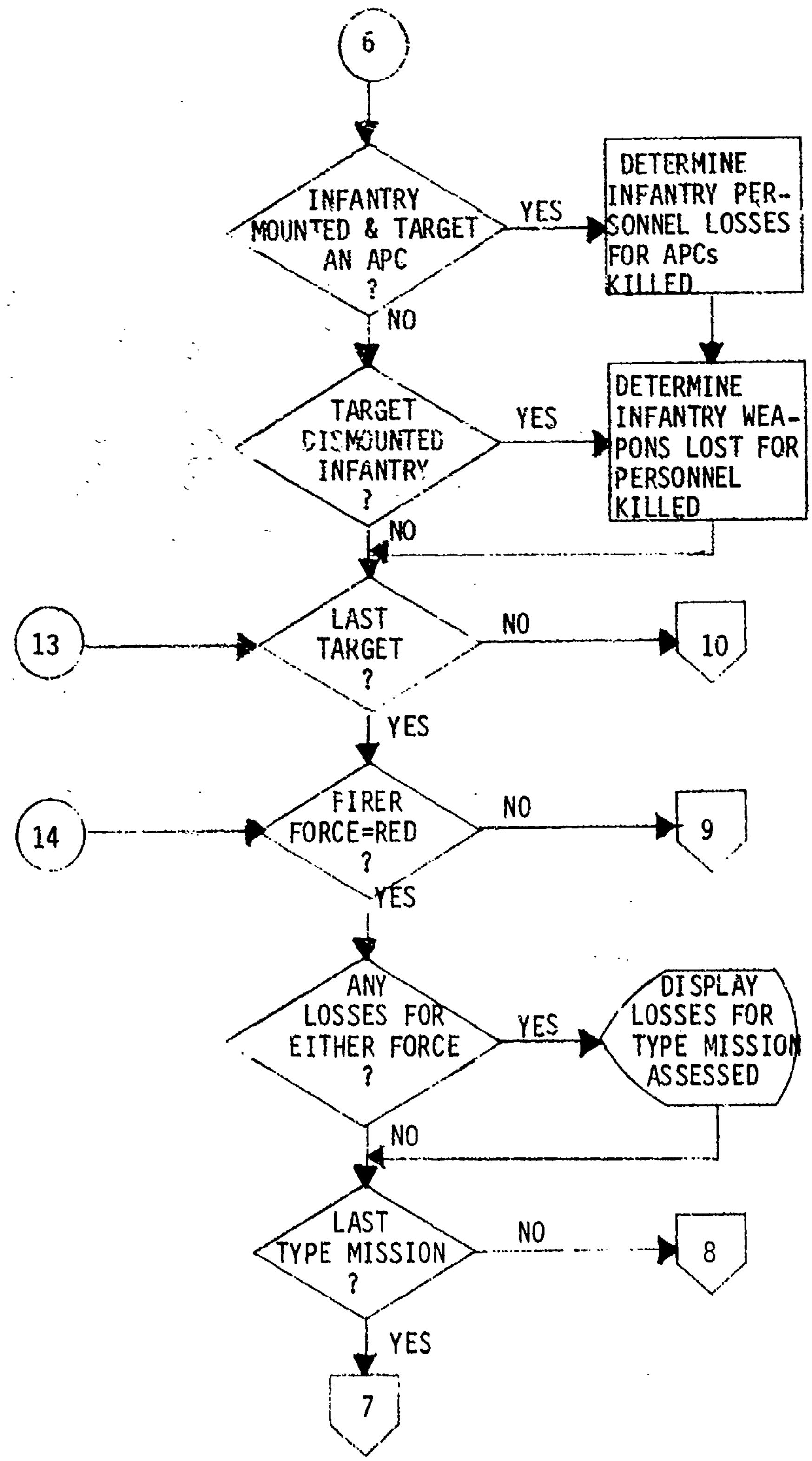

Figure 18. CANNON logic flow diagram (continued).

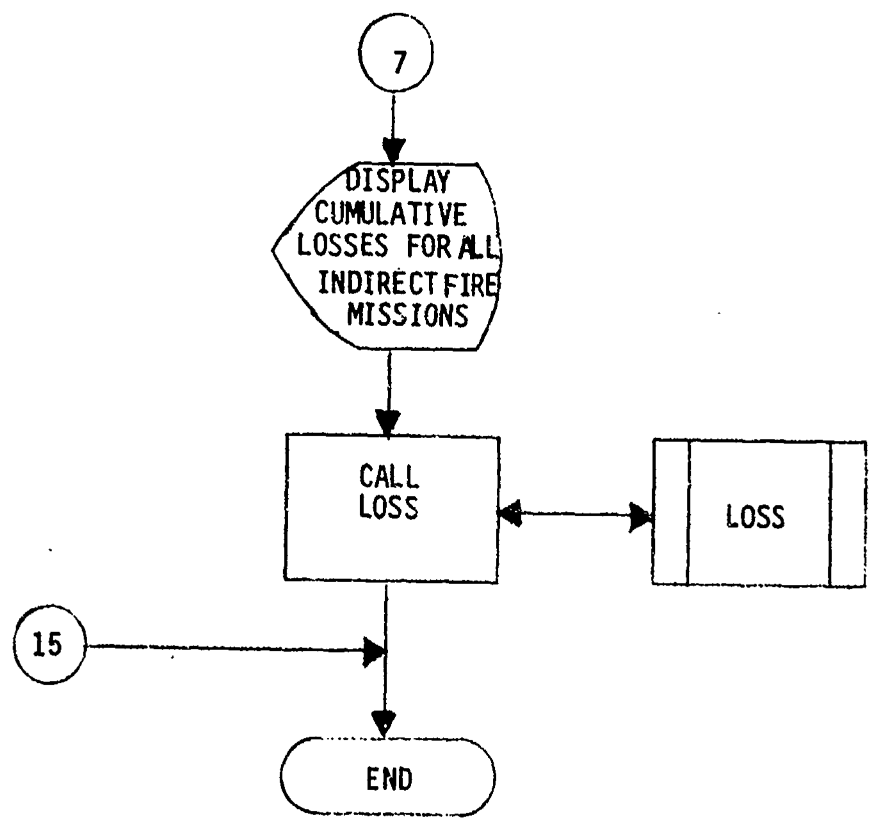

Figure 18. CANNON logic flow diagram (concluded).

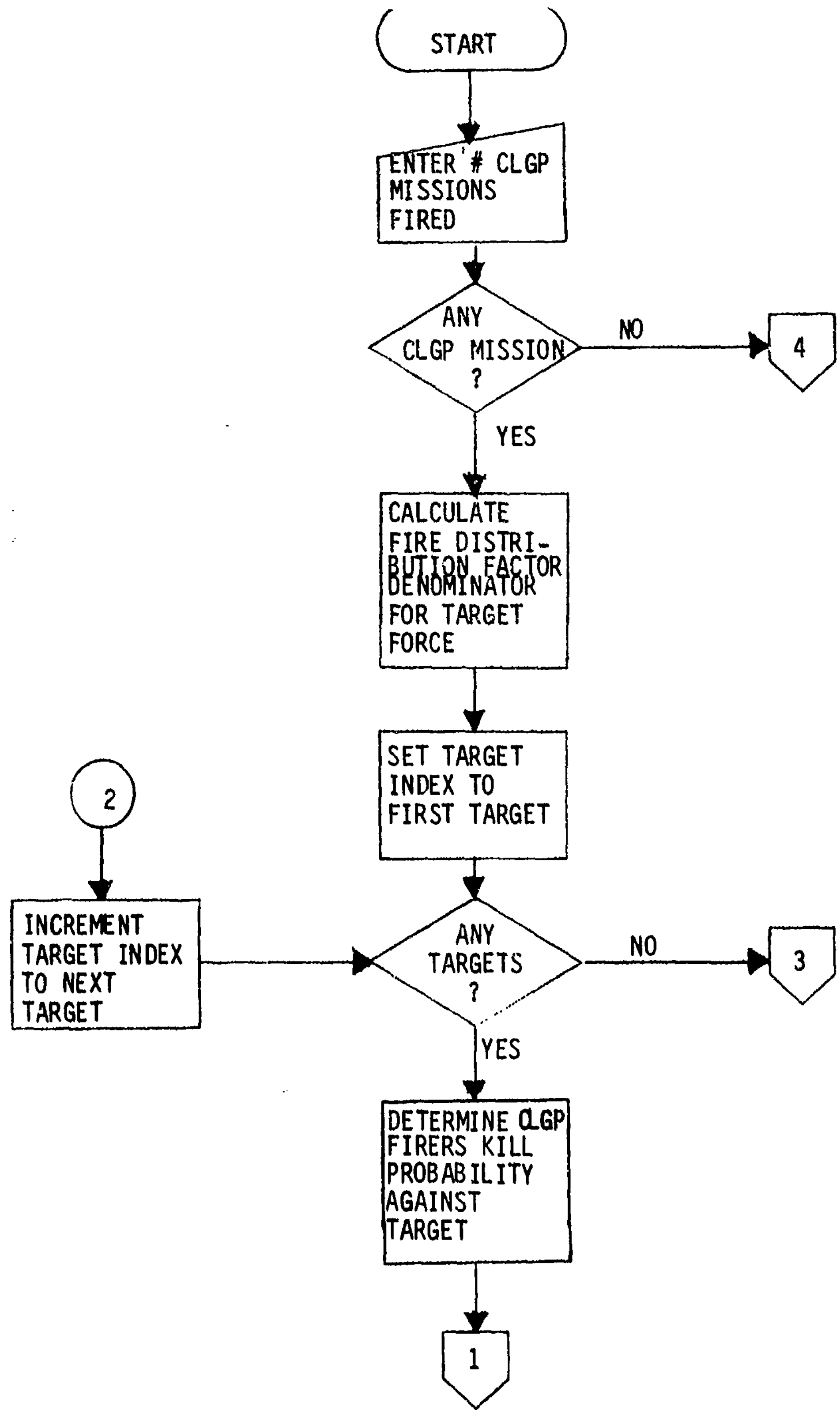

Figure 19. Subroutine CLGP flow diagram.
(Continued next page)

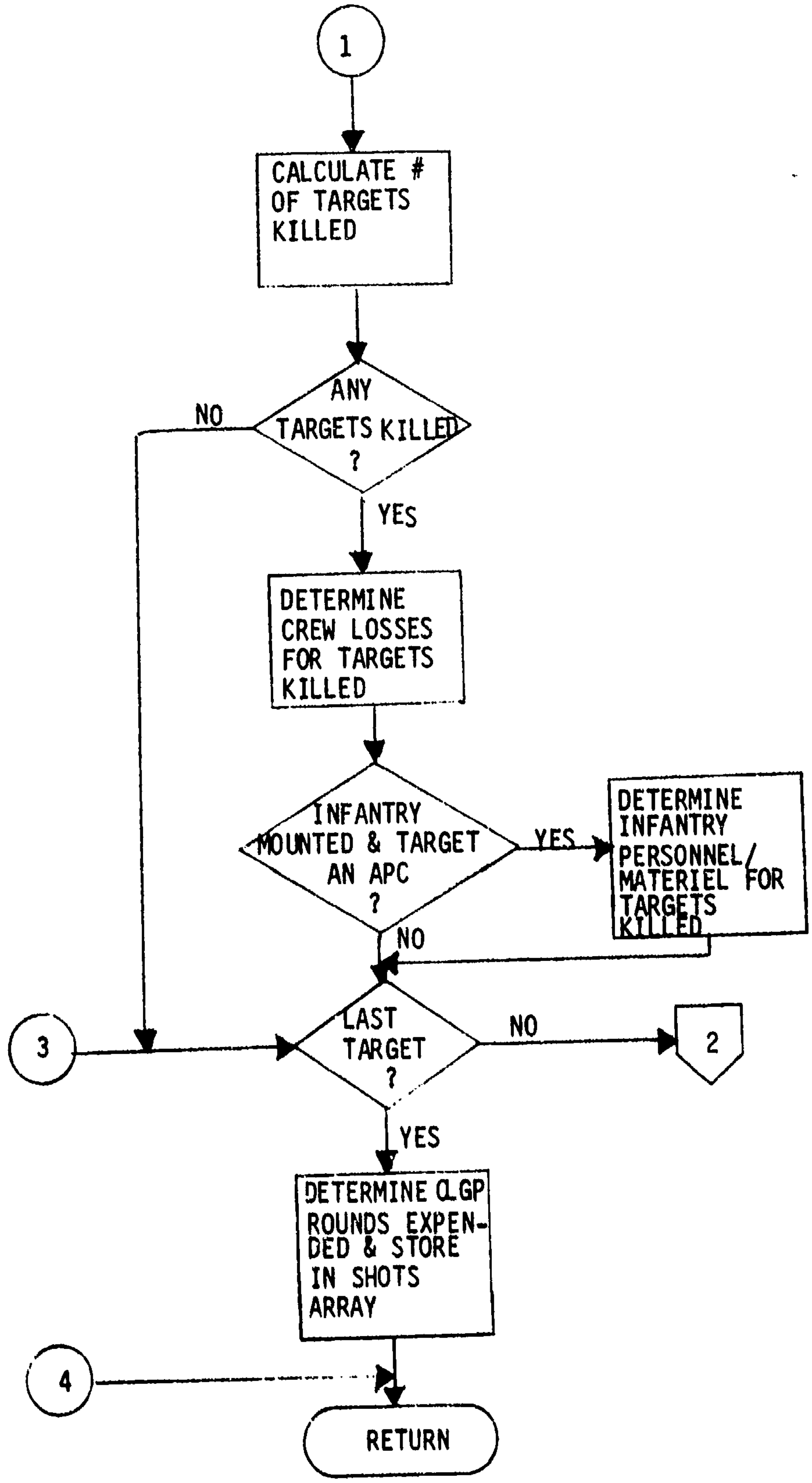

Figure 19. Subroutine CLGP flow diagram (concluded).

are returned to the main program and displayed from there as part of the indirect fire results. Since there is only one type of weapon that fires CLGP, the routine contains only one major DO loop to assess each possible target in the Red force. The CLGP program variables are listed in table L-2, and the FORTRAN source code is given in figure L-2.

(8) OVERLAY 8. SUPRES, the routine that determines the suppression factors for the attacking and defending forces, is contained in OVERLAY 8. SUPRES is composed of an array of the suppression factors used in the Jiffy Game and a few lines of code that access the data and set the suppression factors for both forces. A list of the program variables and a listing of the SUPRES FORTRAN source code are contained in appendix M. The program logic flow diagram is presented in figure 20.

(9) OVERLAY 9. Program OVLY 9 is called from the Jiffy Game supervisory program (SUPER) at DECISION POINT number 5 (see table 1). This overlay contains no subroutine nor does it call any external subroutines from OVLYO. The purpose of the program is to provide hard copy output of the results for a battle gamed with the Jiffy Game assessment routines. Figure 21 contains the logic flow diagram of RESULT. The routine tabulates the killer/victim results from the ALOSS array and the ammunition expenditures from the SHOTS array. Several tables are created to be output from a high speed printer. These tables display the cumulative results in formats determined to be most meaningful for analyzing and summarizing the outcome of the battle. The OVLY 9 FORTRAN source code is given in figure N-1, and the program variables are listed in table N-1.

(10) OVERLAY 10. OVERLAY 10 (FORCE) is the program by which the gamers manipulate their forces in the Jiffy Game. OVLY 10 is reached by a gamer response of "1" at the DECISION POINT in SUPER (see table 1). After the gamer defines the critical incident and sector, he is presented his choice of the eight force manipulation options in table 2. Upon completion of all but OPTION 0, the gamer is returned to the OPTION point. A response of "0" loads the weapon systems of all units loaded into the defined sector and critical incident into the weapon system (ELMT) array for both forces and returns control to SUPER. The display option (6) provides the gamer the capability to examine the FORCE file in four ways. The four types of displays accessible at OPTION 6 are given in table 3. Subroutine DISPLAY is used for display type 4. The program logic flow diagram for OVLY 10 is contained in figure 22. The FORTRAN program source code for OVLY 10 and a list of the program variables used in the overlay are presented in appendix O.

(11) OVERLAY 11. OVERLAY 11 apportions the personnel casualties and weapon system losses determined in the Jiffy Game combat assessment routines to the units on the FORCE file. The program in OVERLAY 11 is named APPORT. The apportionment is based on an algorithm that considers quantity of losses, number of weapon systems in the unit, and the level of combat intensity of the actions in which the unit was involved during the

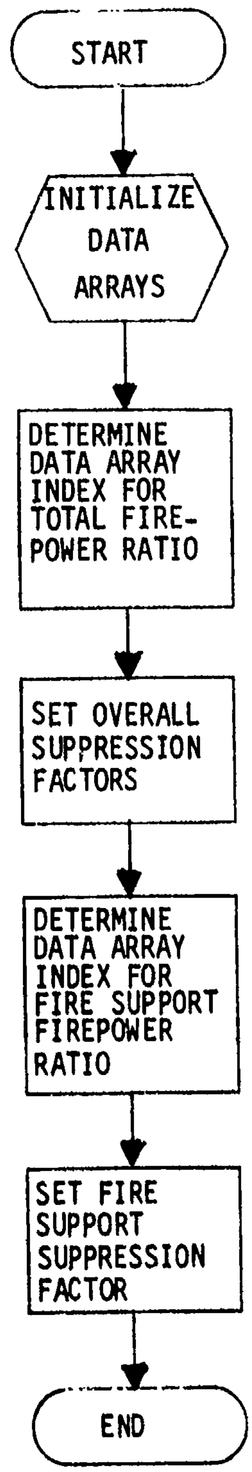

Figure 20. SUPRES flow diagram.

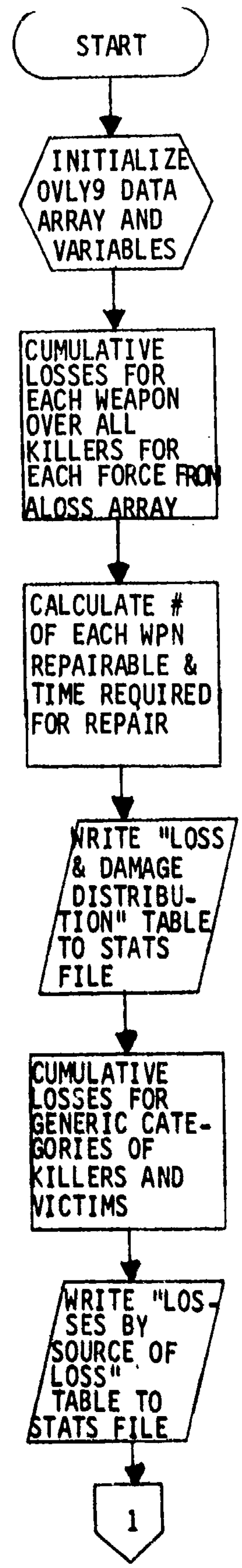

Figure 21. OVLY9 (RESULT) flow diagram.
(Continued next page)

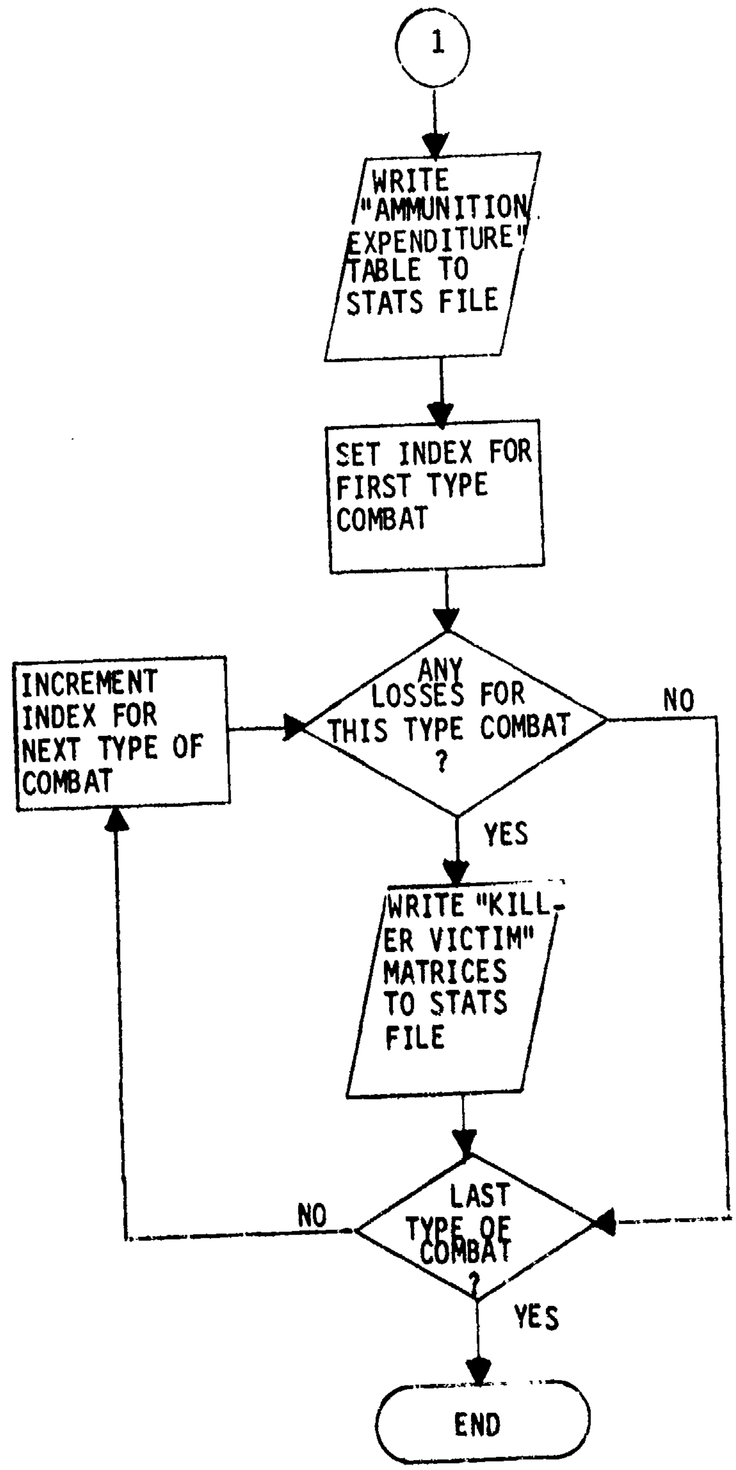

Figure 21. OVLY9 (RESULT) flow diagram (concluded).

Table 2. OVLY10 force manipulation options.

Response Code	Option Description
0	Proceed with assessments
1	Load units into sector
2	Remove units from sector
3	Create a new unit
4	Adjust weapons in a unit
5	Attach a unit to a new parent
6	Display a unit
7	Delete a unit from FORCE file

Table 3. Types of displays.

Display Index	Type Display	Information Displayed
1	Lists all parent units on FORCE file	Parent ID, force designator, Parent unit effectiveness, sector and critical incident
2	Lists all parent units in defined sector and critical incident	Parent ID, force designator and Parent unit effectiveness
3	Lists all units attached to a specific parent unit	Parent ID, Unit ID, force designator, unit effectiveness, sector and critical incident
4	Lists all weapon systems in a specific unit or Parent unit	Parent ID, Parent unit effectiveness, Unit ID, unit effectiveness, quantity and type of weapon systems.

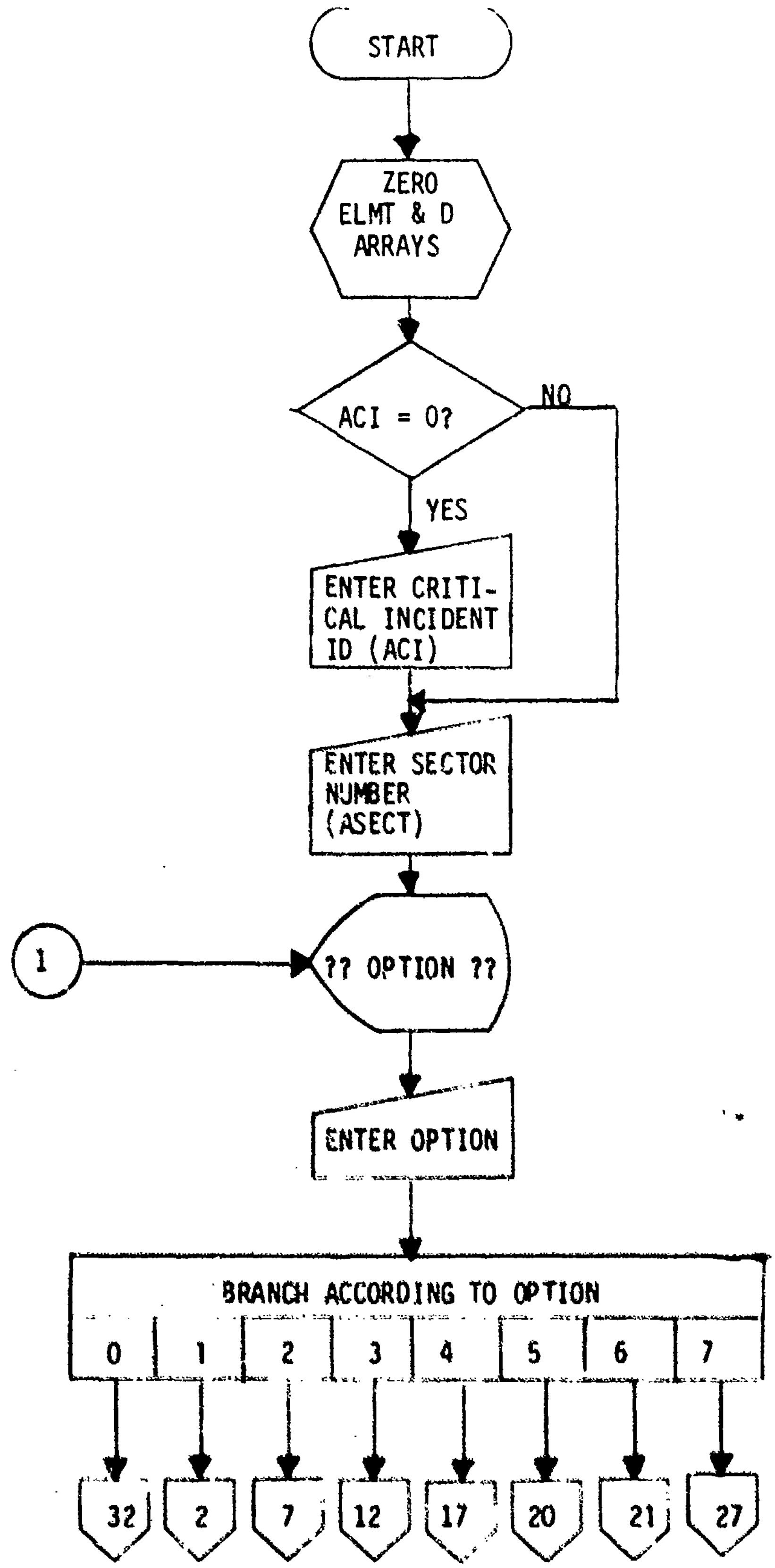

Figure 22. FORCE flow diagram.
(Continued next page)

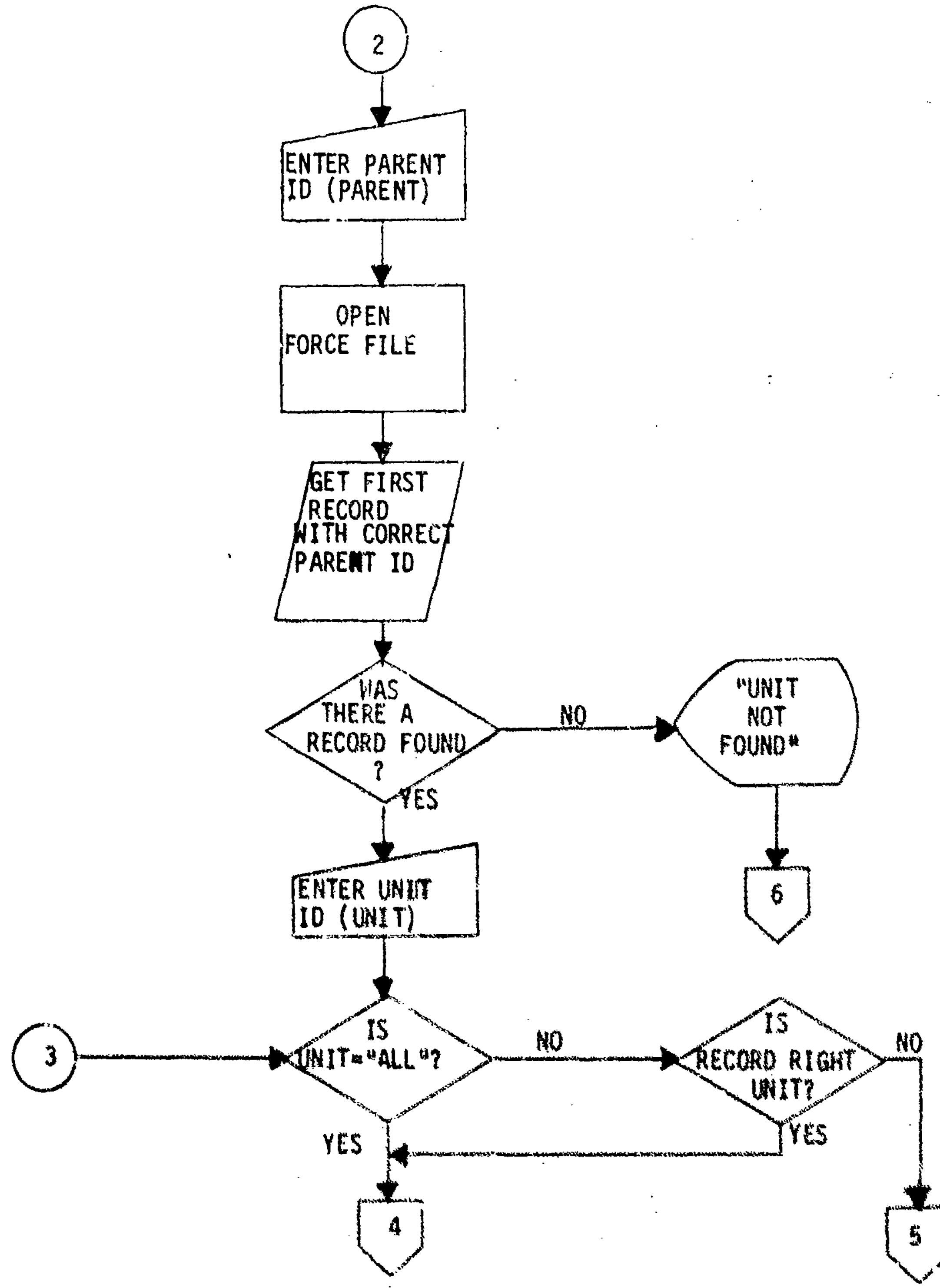

Figure 22. FORCE flow diagram (continued).

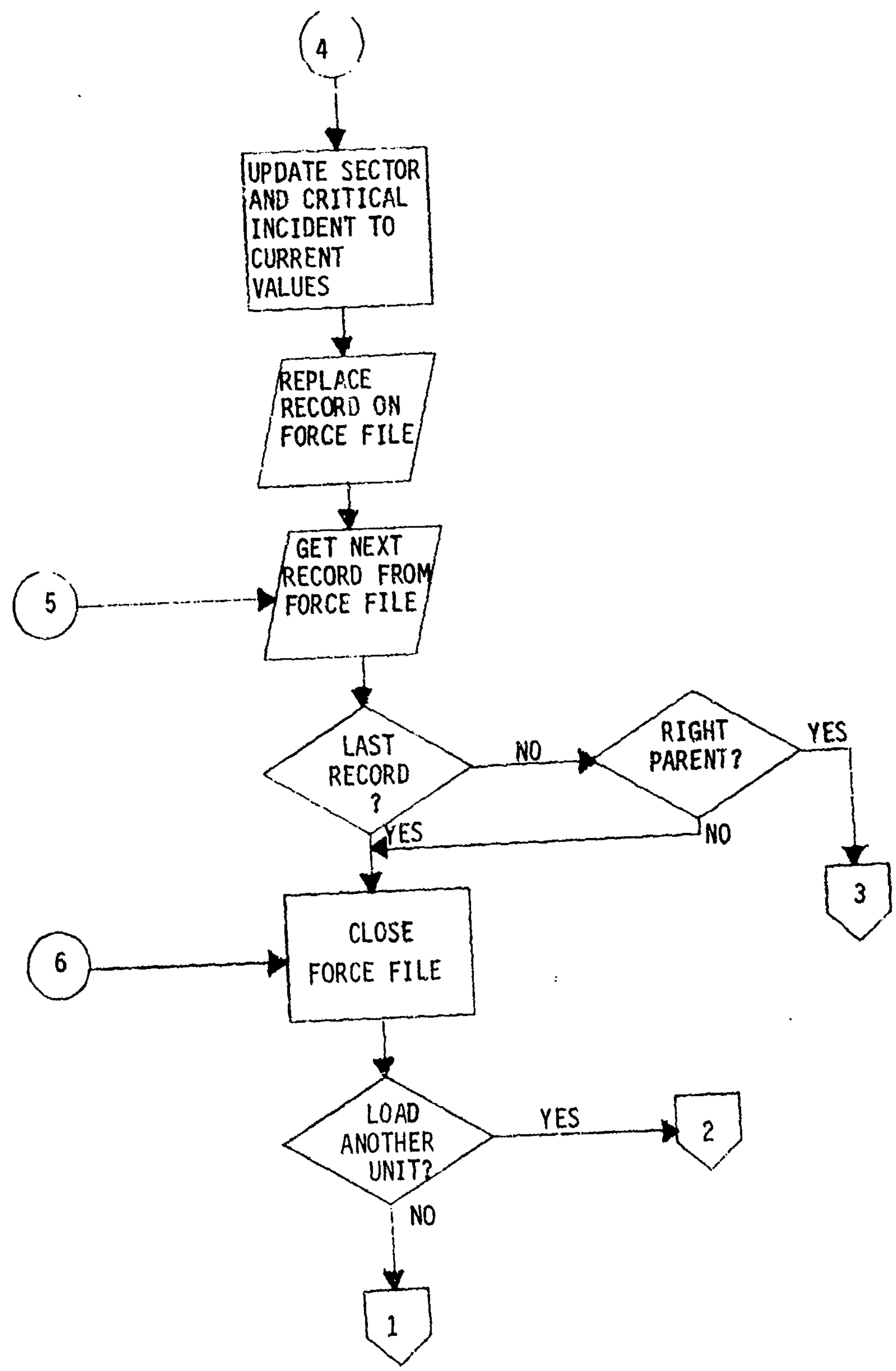

Figure 22. FORCE flow diagram (continued).

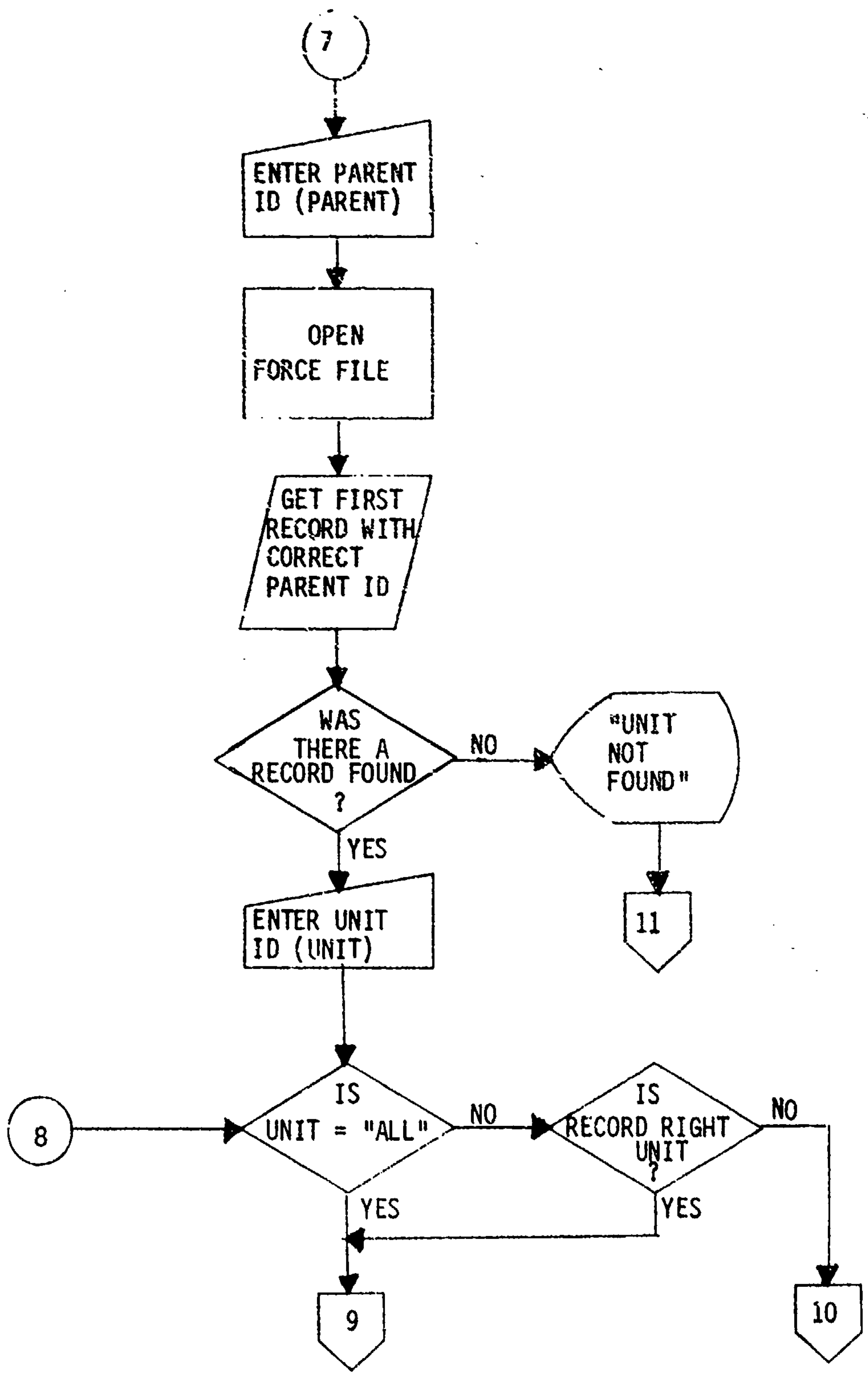

Figure 22. FORCE flow diagram (continued).

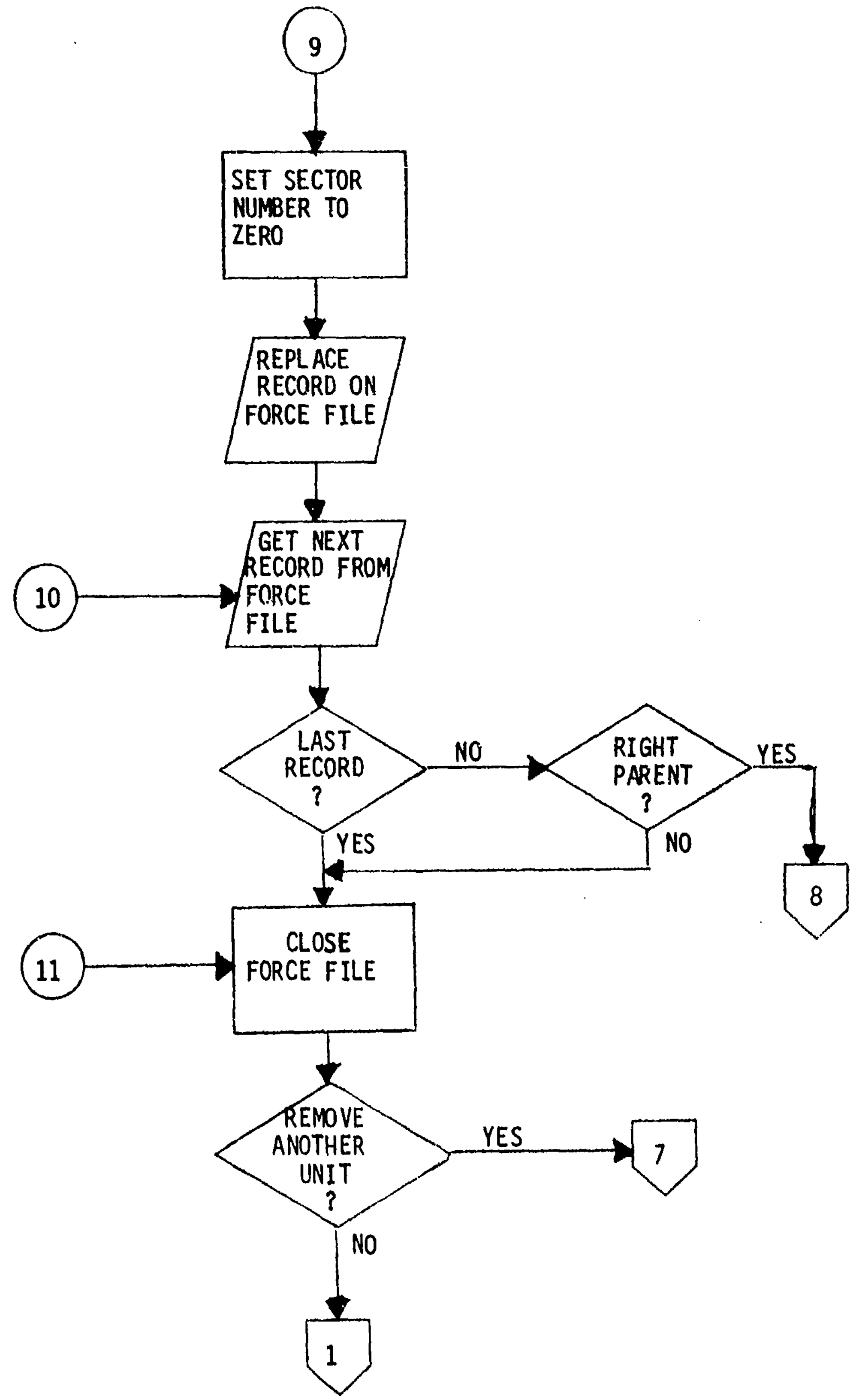

Figure 22. FORCE flow diagram (continued).

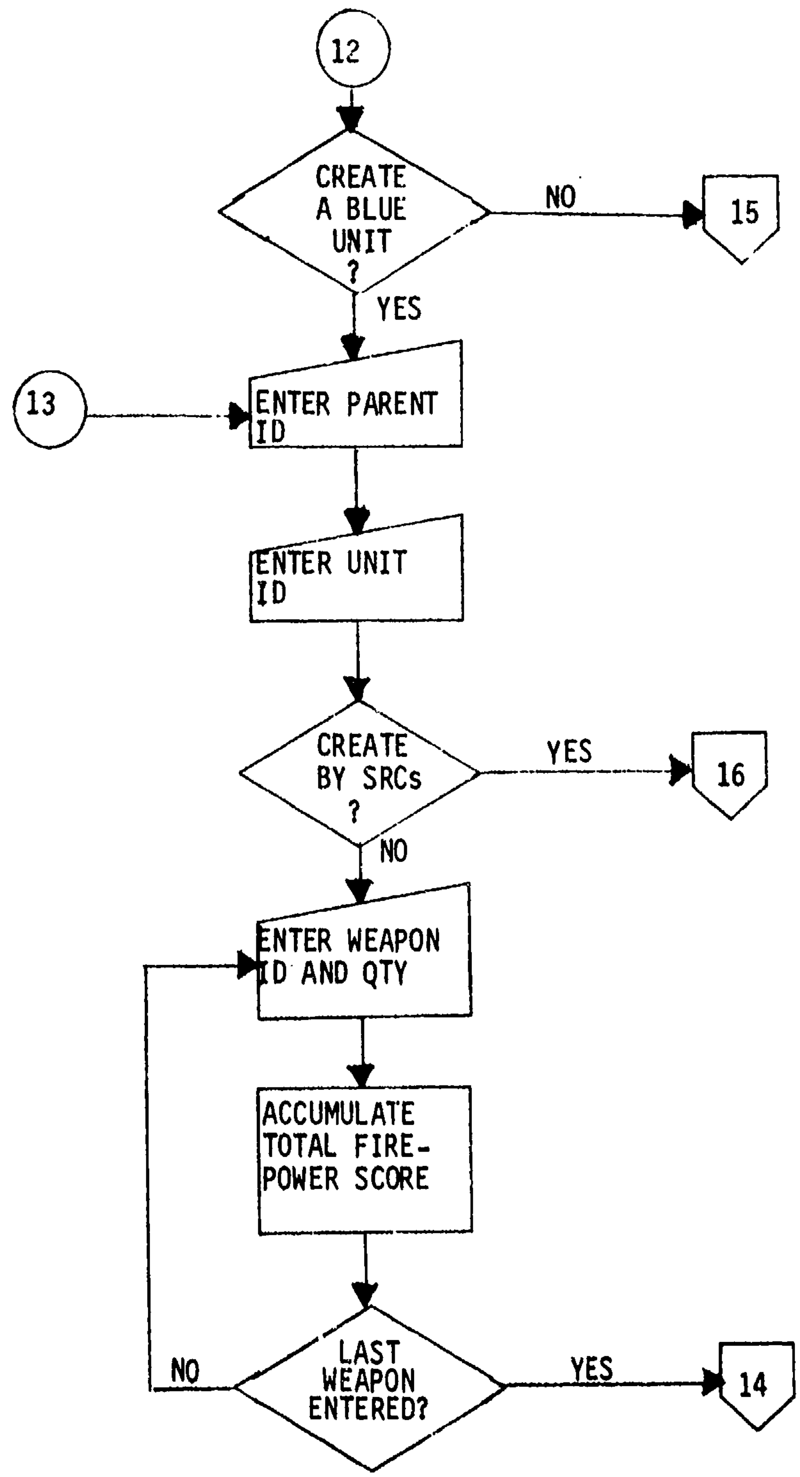

Figure 22. FORCE flow diagram (continued).

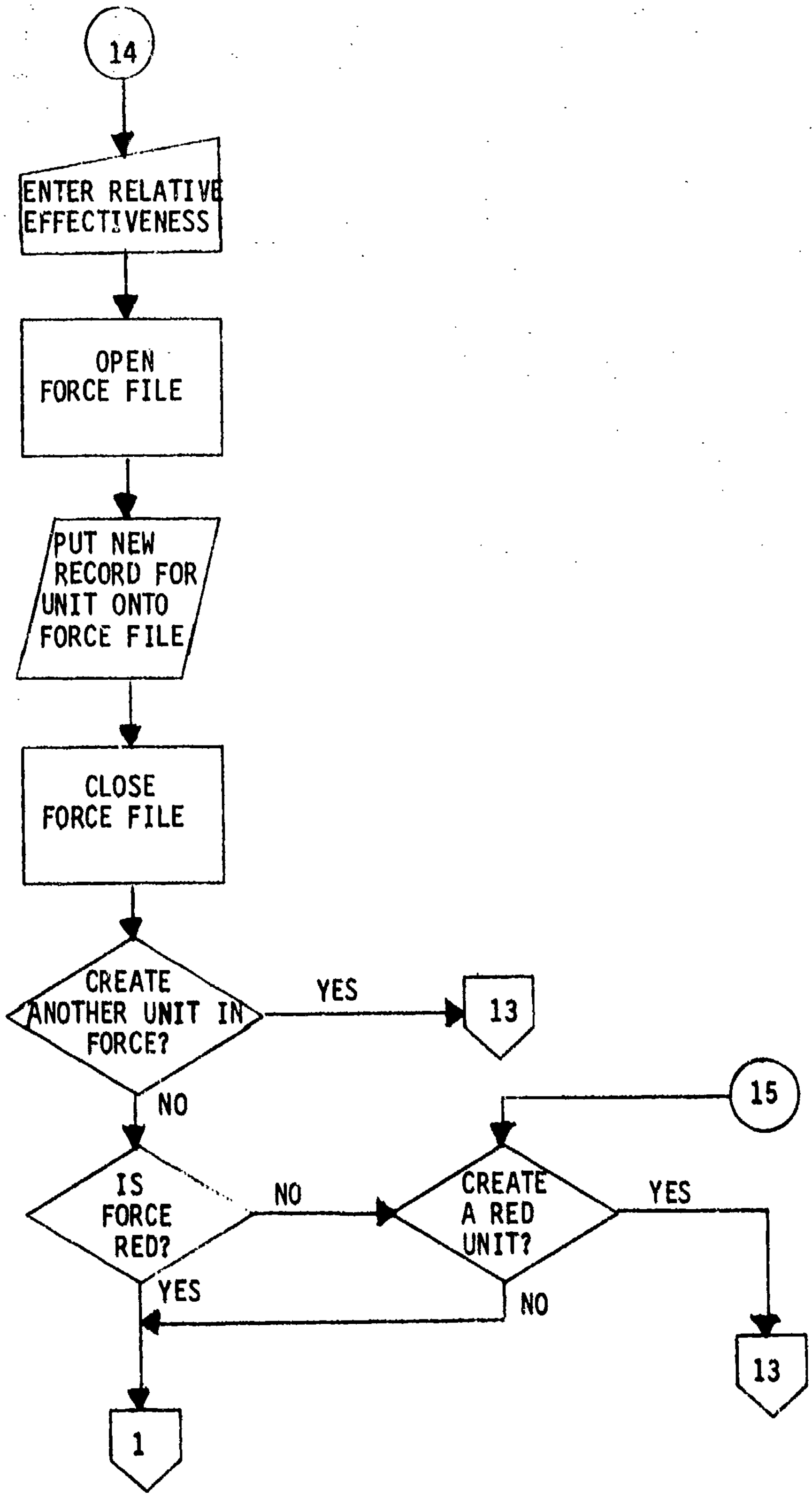

Figure 22. FORCE flow diagram (continued).

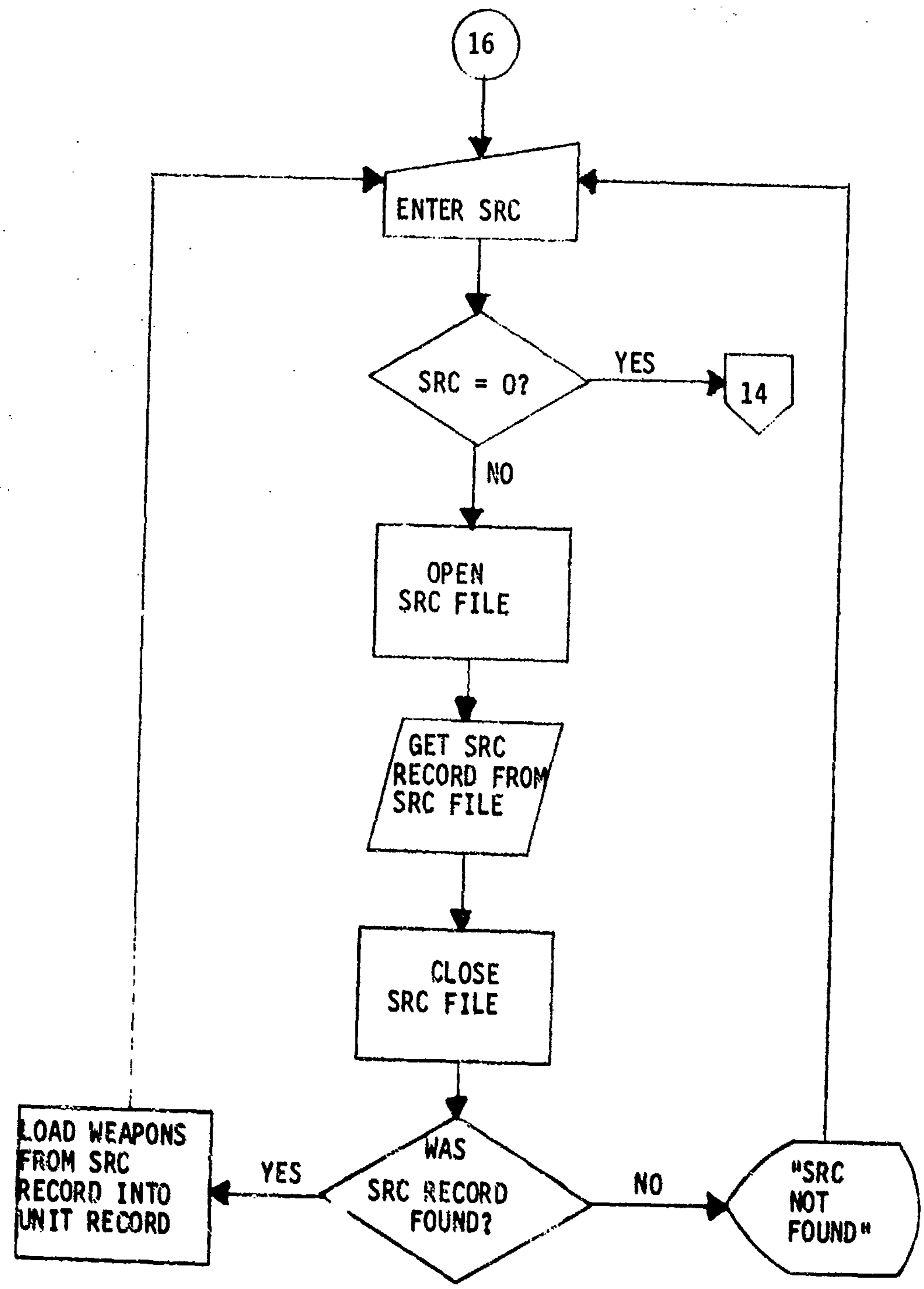

Figure 22. FORCE flow diagram (continued).

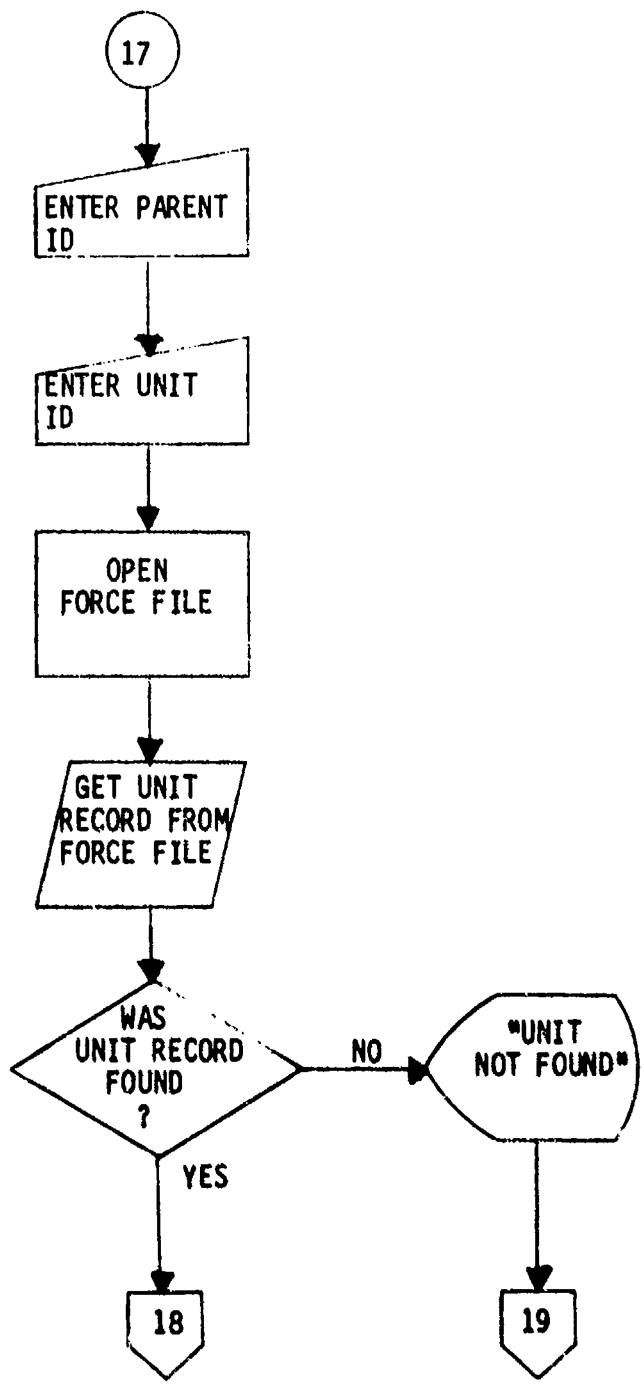

Figure 22. FORCE flow diagram (continued).

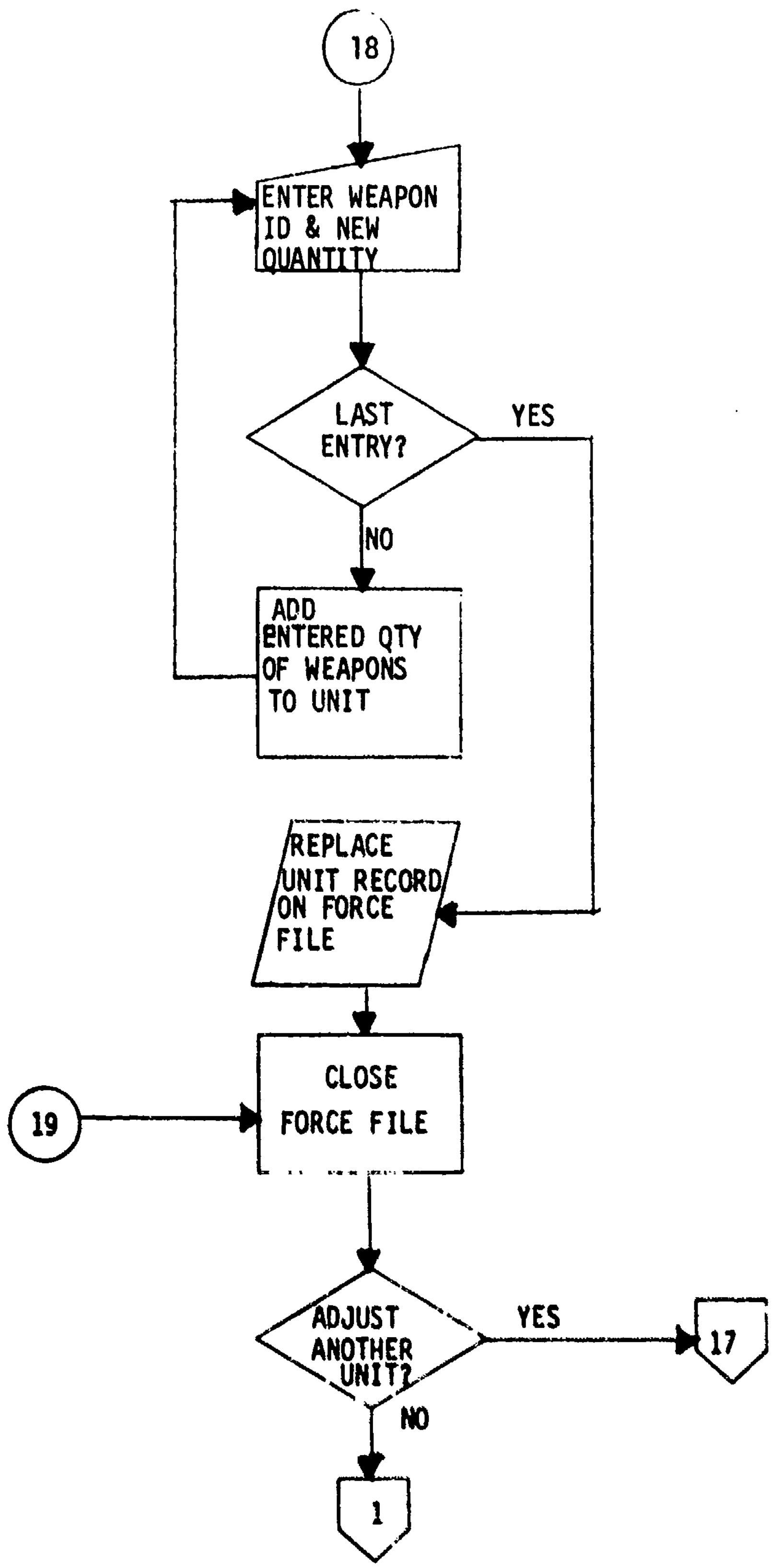

Figure 22. FORCE flow diagram (continued).

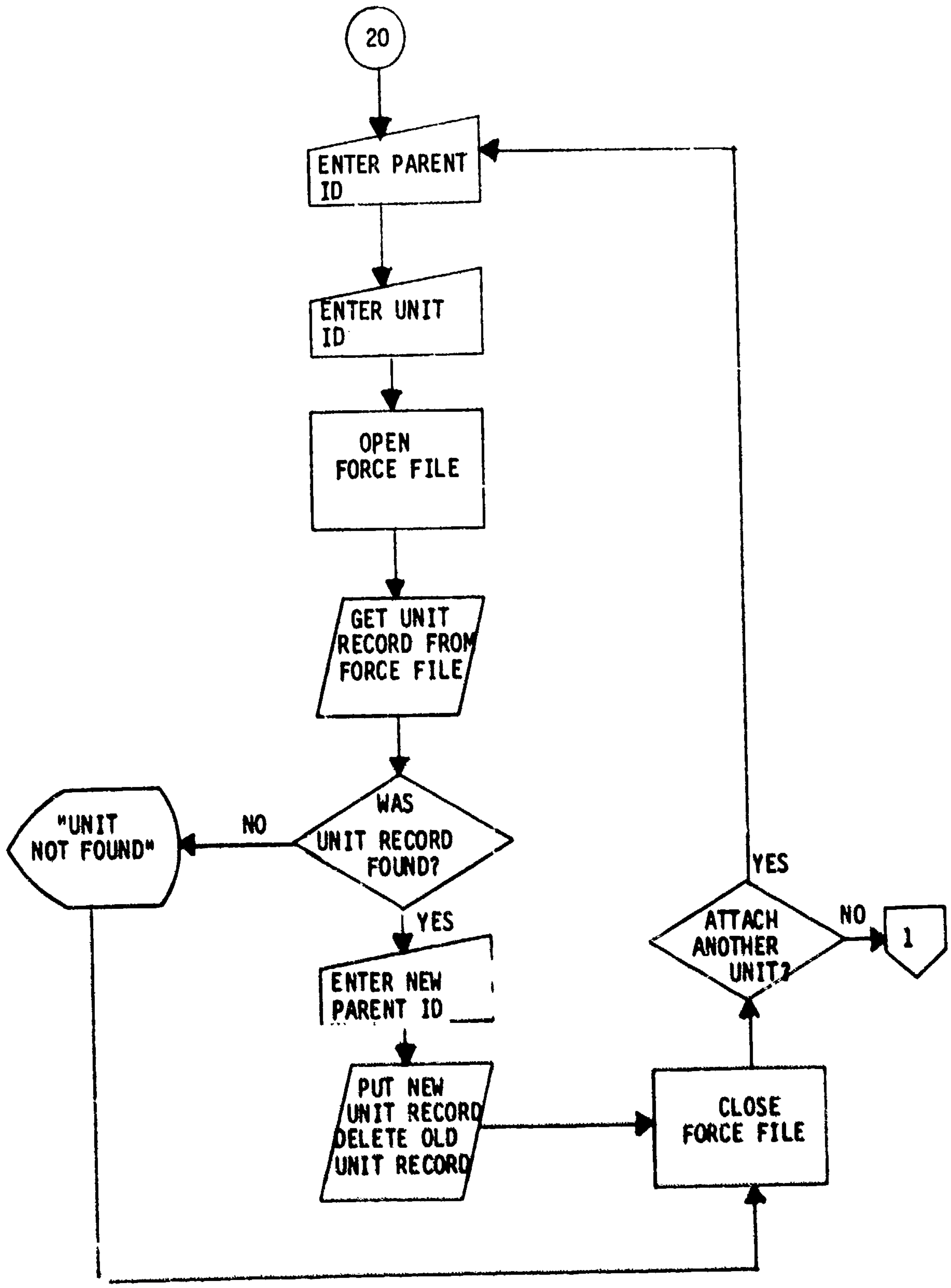

Figure 22. FORCE flow diagram (continued).

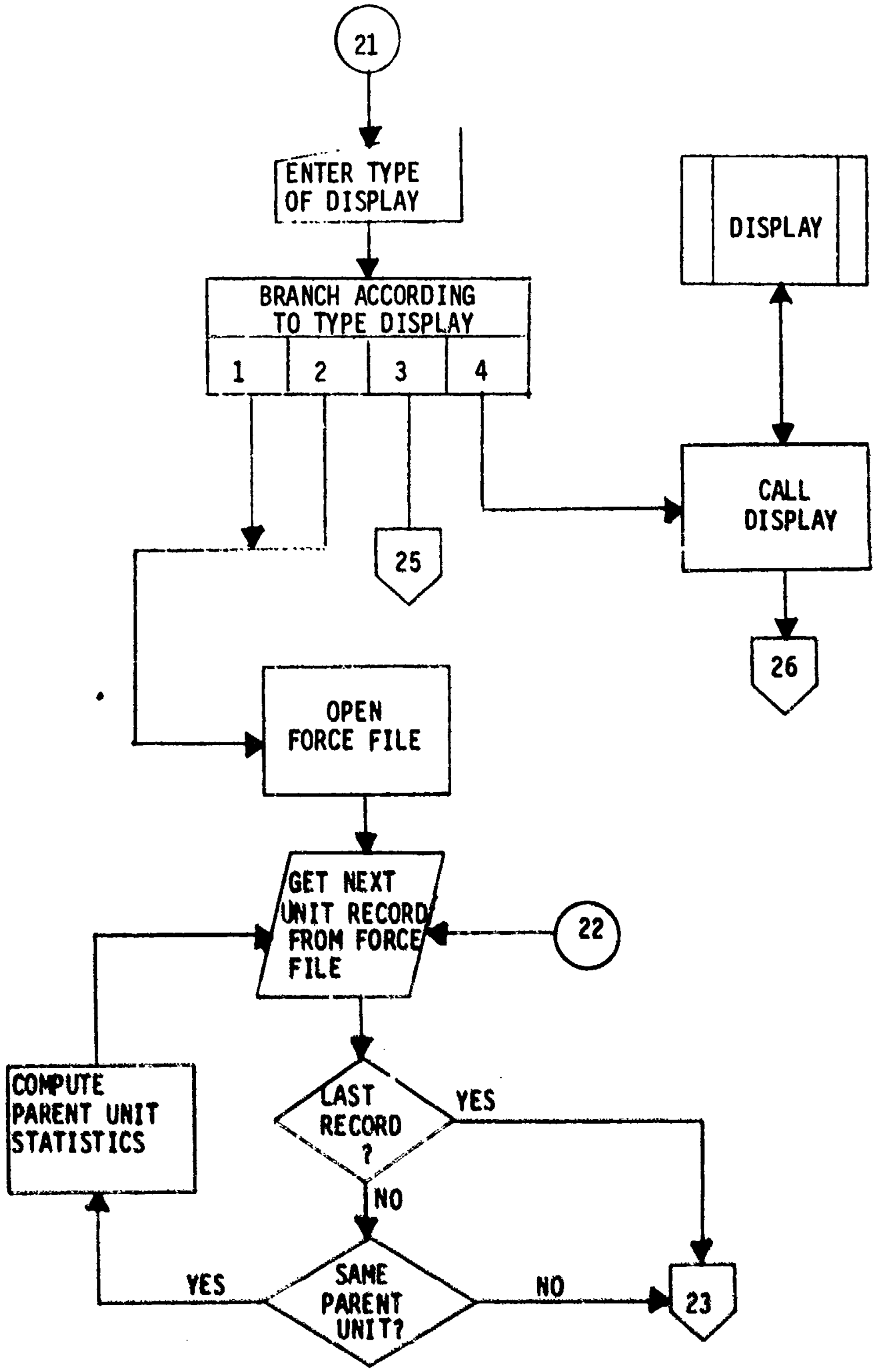

Figure 22. FORCE flow diagram (continued).

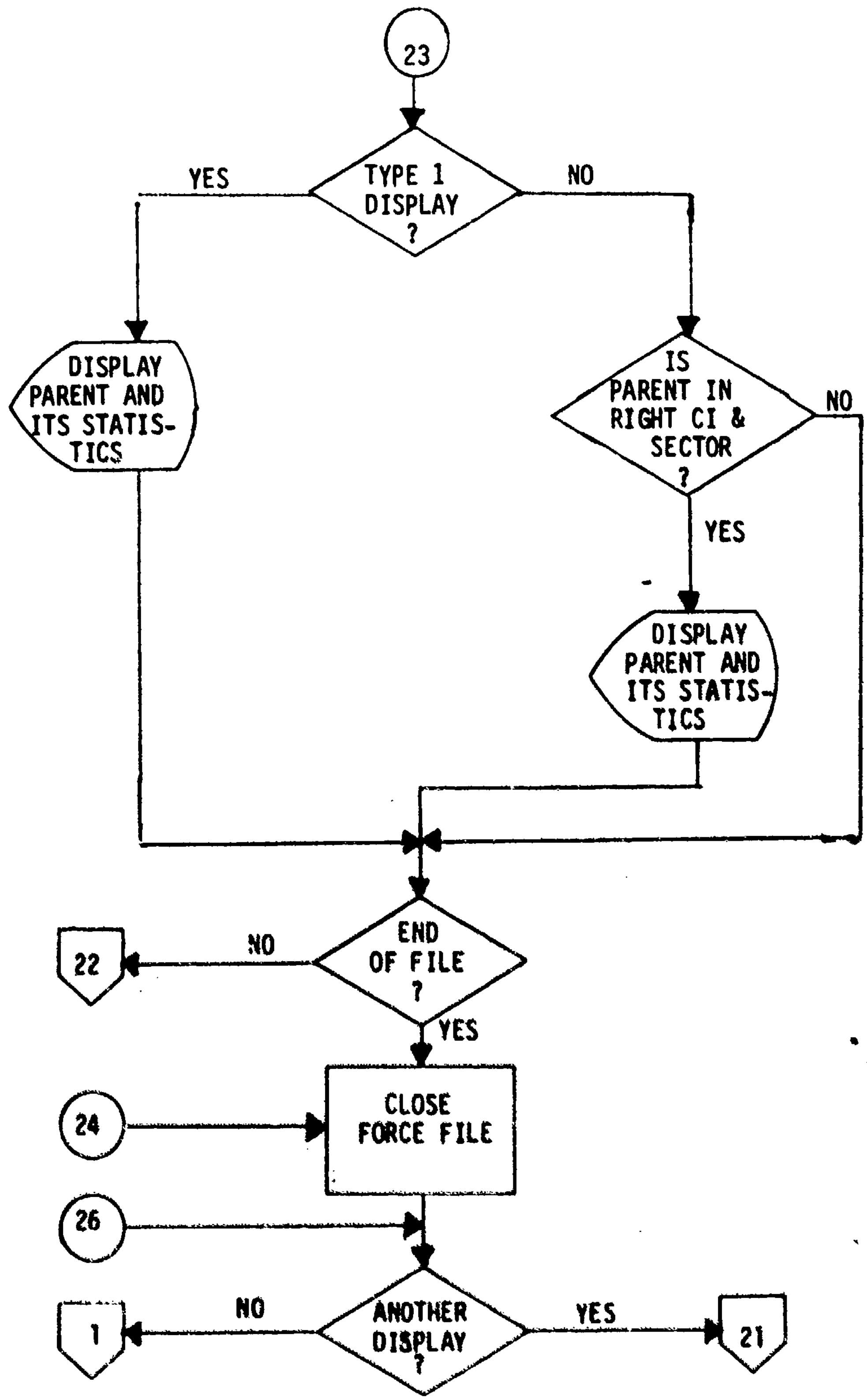

Figure 22. FORCE flow diagram (continued).

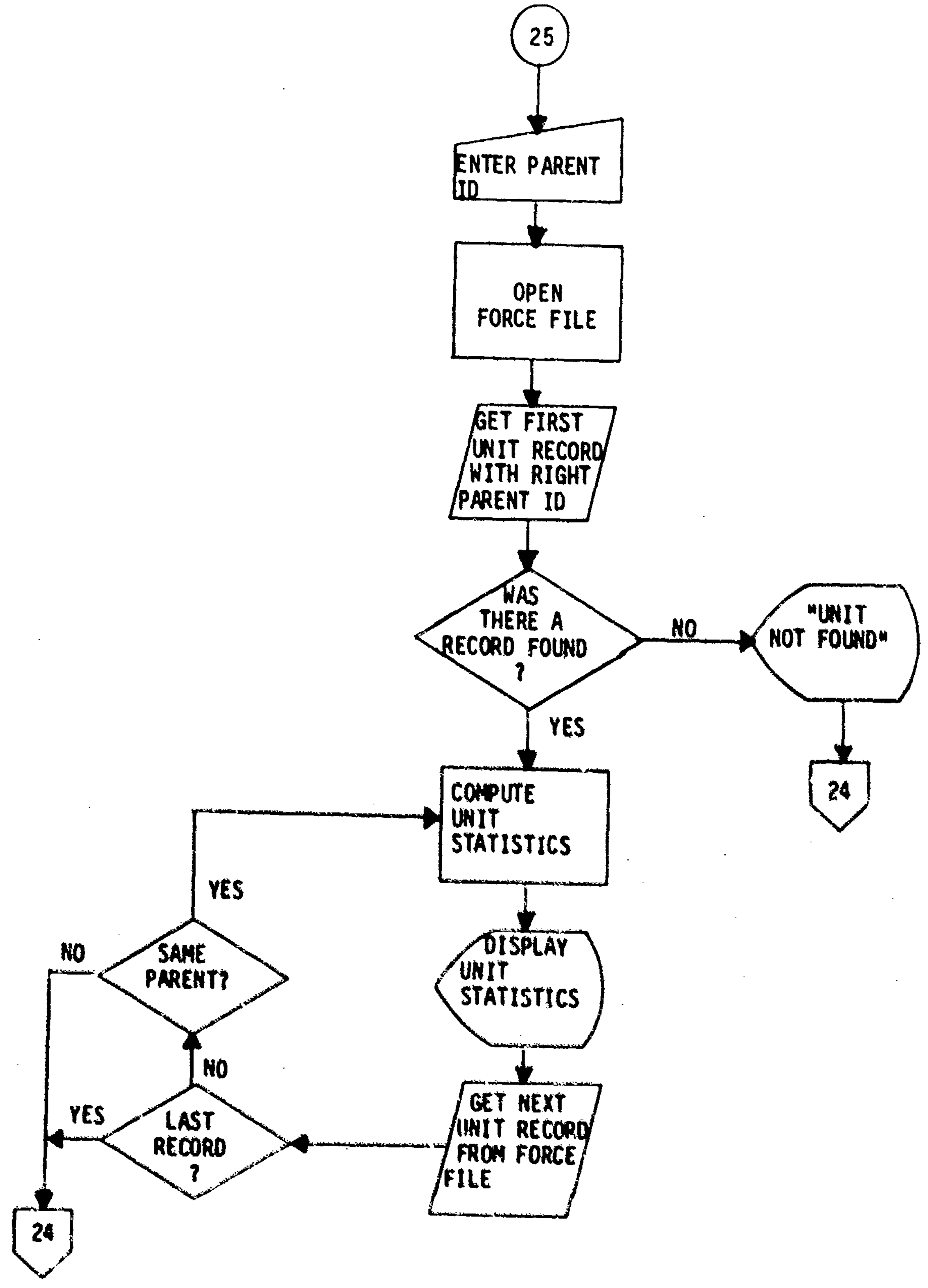

Figure 22. FORCE flow diagram (continued).

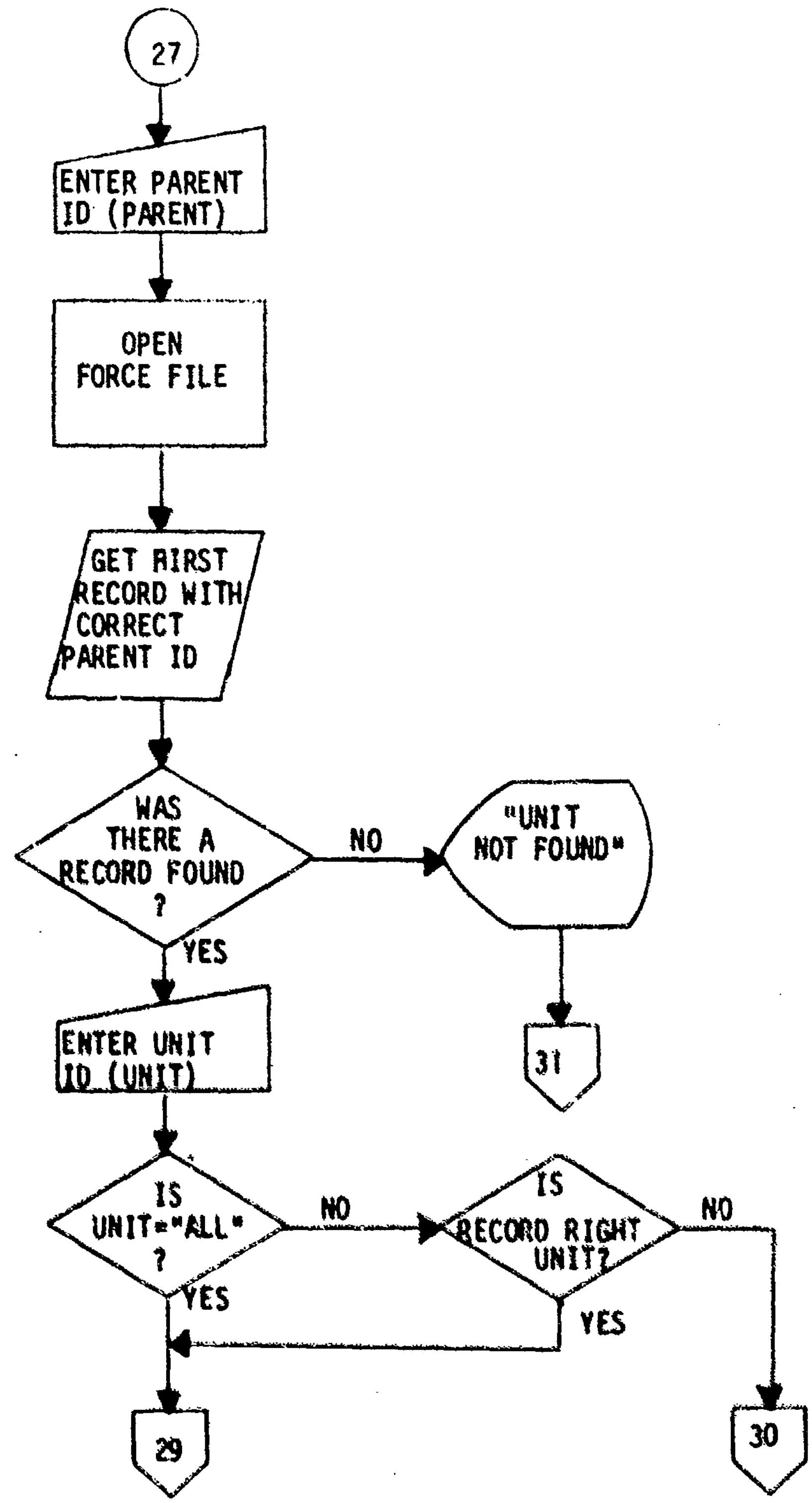

Figure 22. FORCE flow diagram (continued).

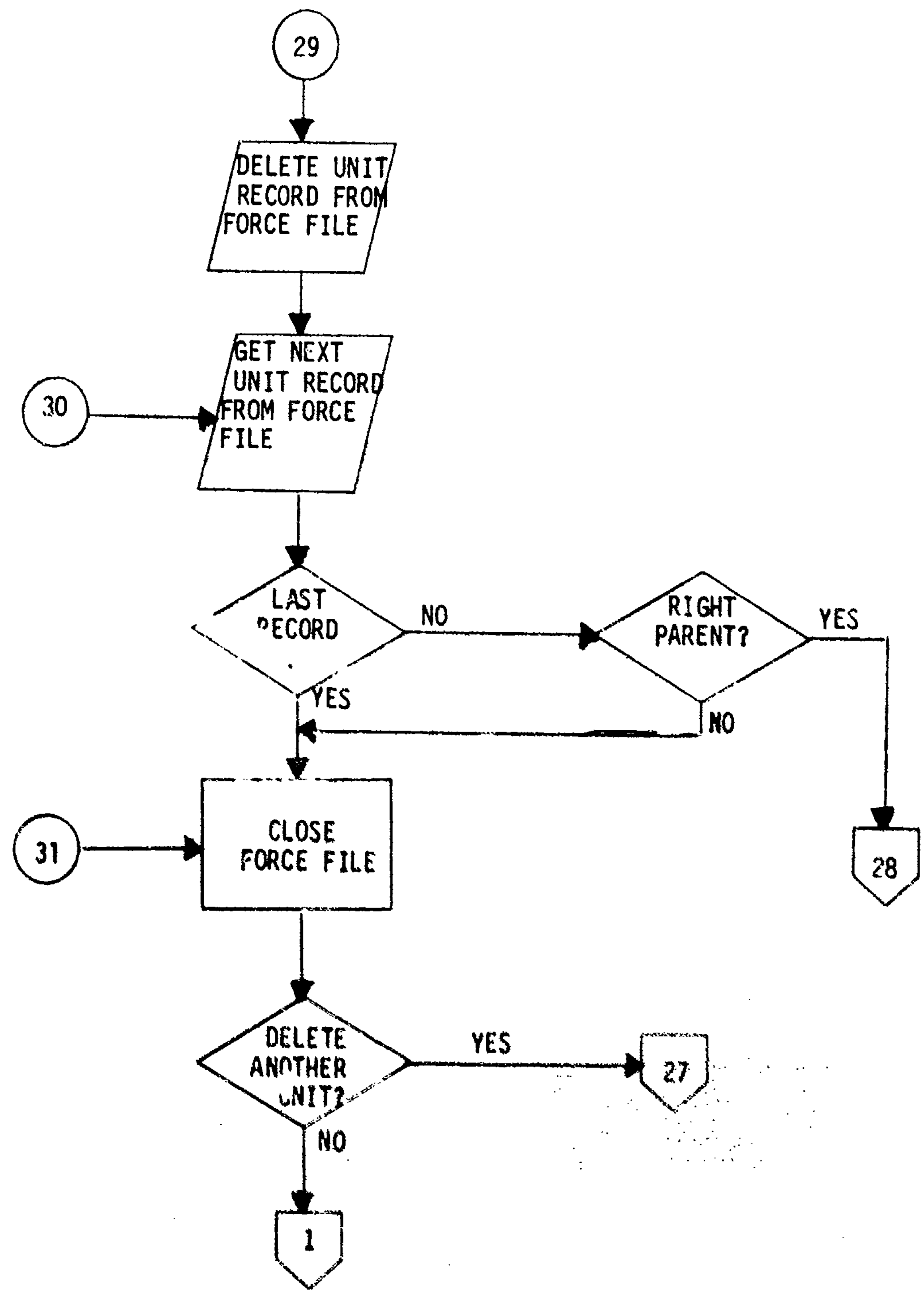

Figure 22. FORCE flow diagram (continued).

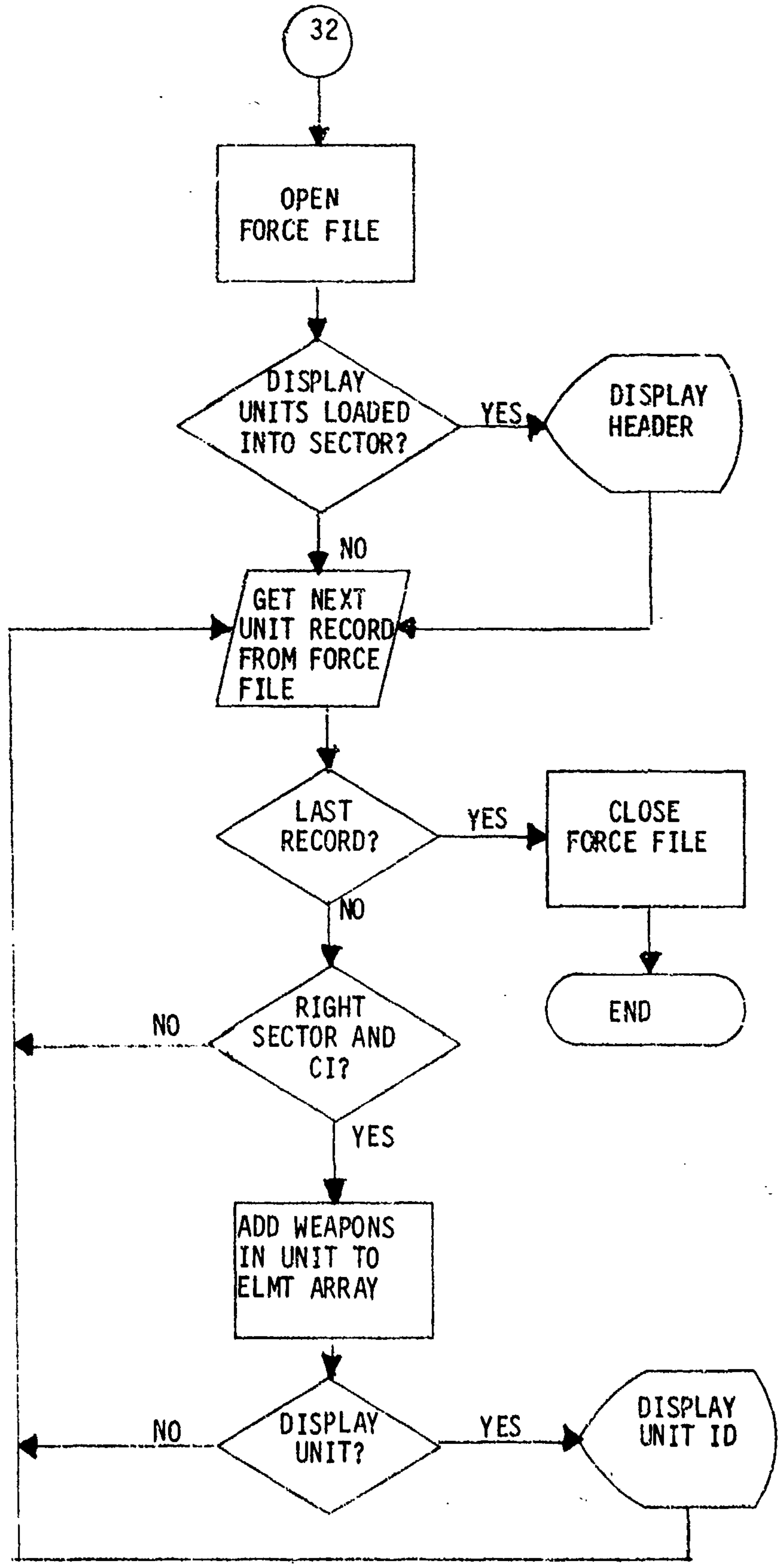

Figure 22. FORCE flow diagram (concluded).

assessment period. The combat intensity level for each unit in a given sector during a critical incident is input interactively by the gamers. The APPORT overlay also compiles the cumulative combat statistics of all sectors for the entire critical incident. The cumulative loss and ammunition expenditures are kept on the HISTORY file. The overlay also provides the gamers with the capability to display any specified parent unit or unit after the apportionment process by calling the DISPLAY subroutine. Figure 23 is the APPORT logic flow diagram. A list of the program variables along with a listing of the FORTRAN source code is contained in appendix P, table P-1 and figure P-1, respectively.

(12) OVLY 12. OVERLAY 12 (BUILD) contains a single program, which is a duplicate of the SRC program (see paragraph 3c(1)). A copy of the SRC program was included in the Jiffy Game overlays to provide the gamers the capability to create interactively new units in a force with existing or new SRCs during actual processing of the Jiffy Game. BUILD allows the gamers to develop new SRCs. The program logic flow diagram for BUILD is identical to the flow diagram presented for the SRC program (figure 2). The FORTRAN source code for BUILD is contained in appendix Q, figure Q-1. The BUILD program variables are presented in table Q-1.

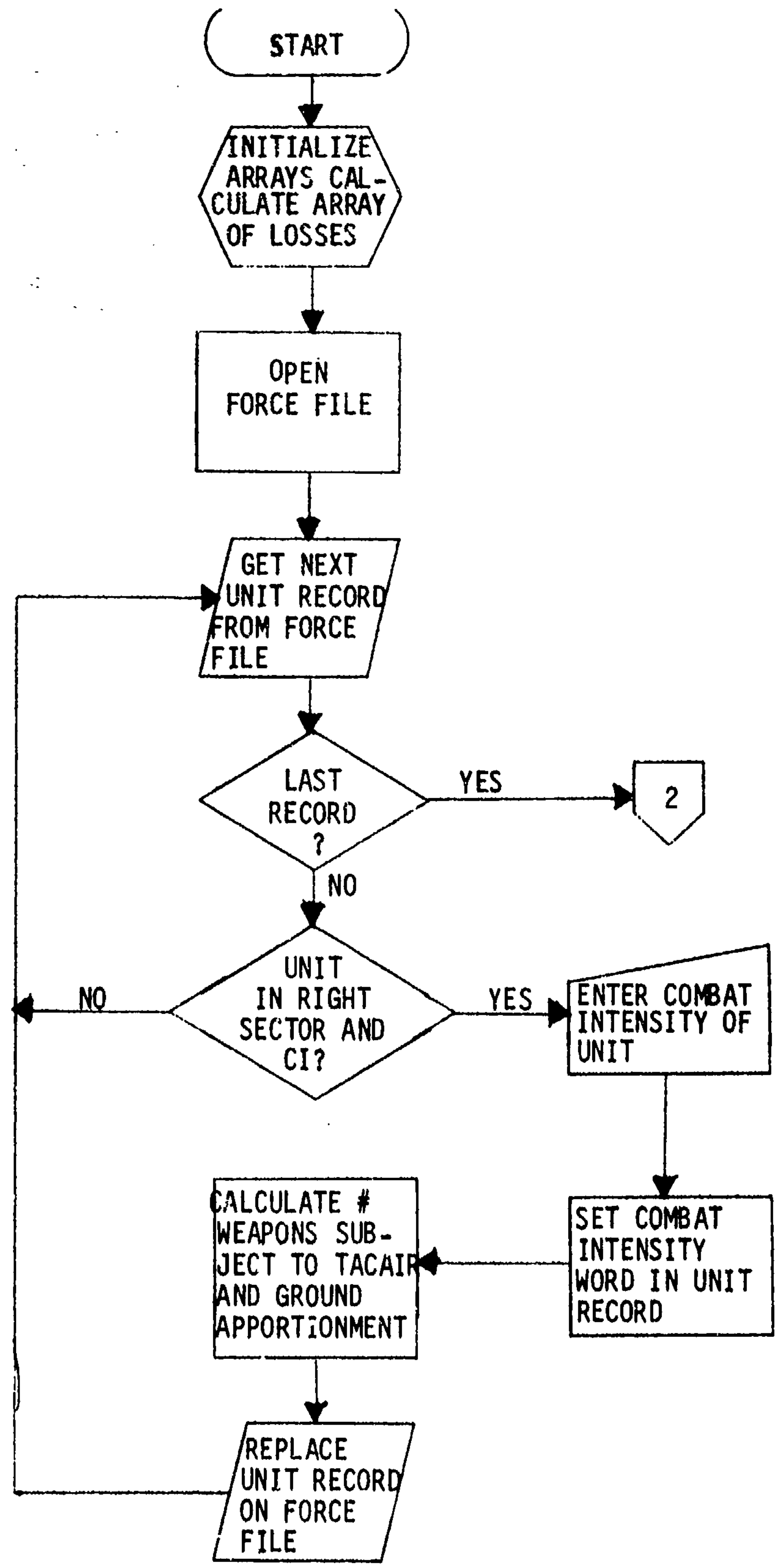

Figure 23. APPORT flow diagram.
(Continued next page)

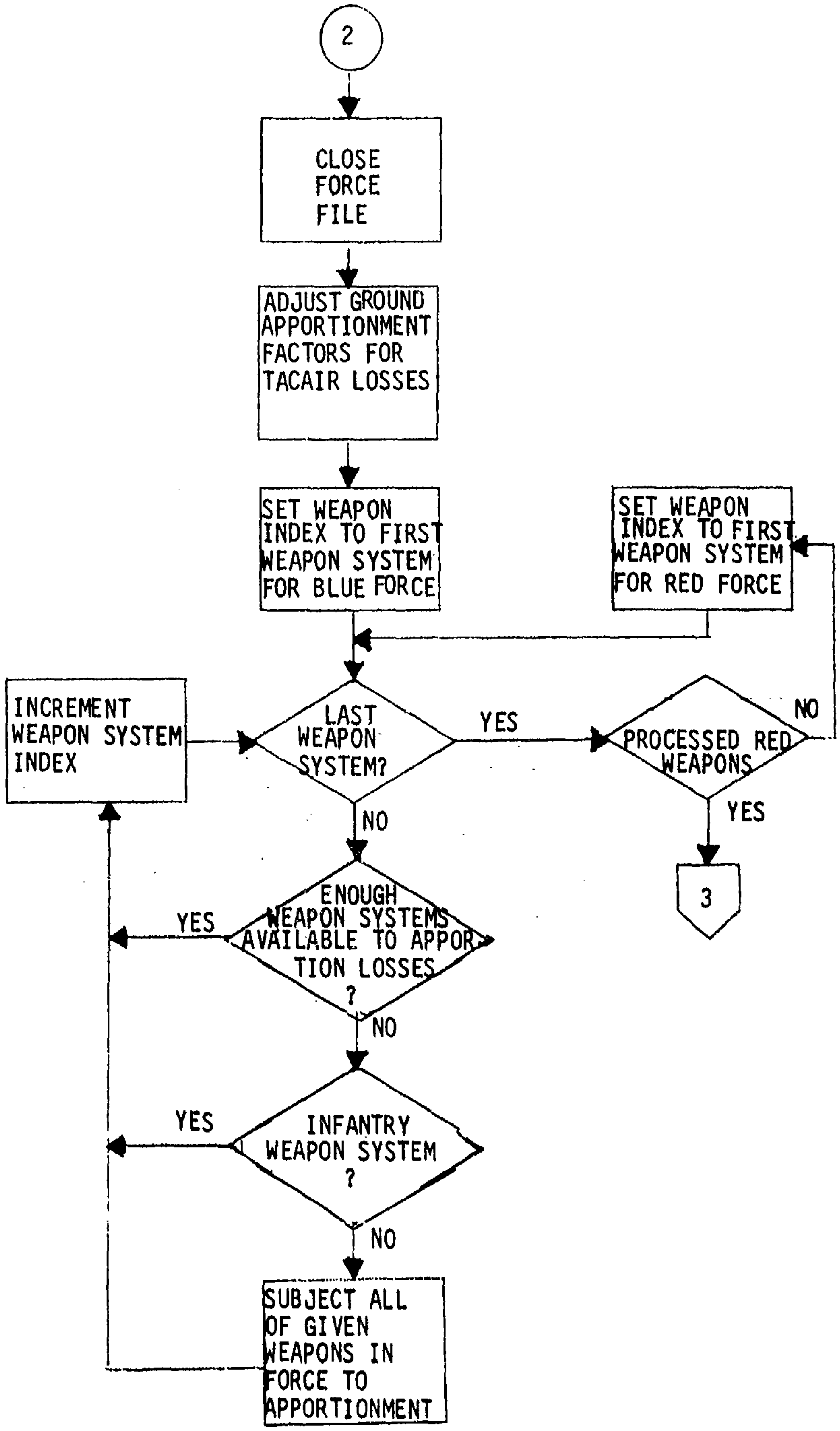

Figure 23. APPORT flow diagram (continued).

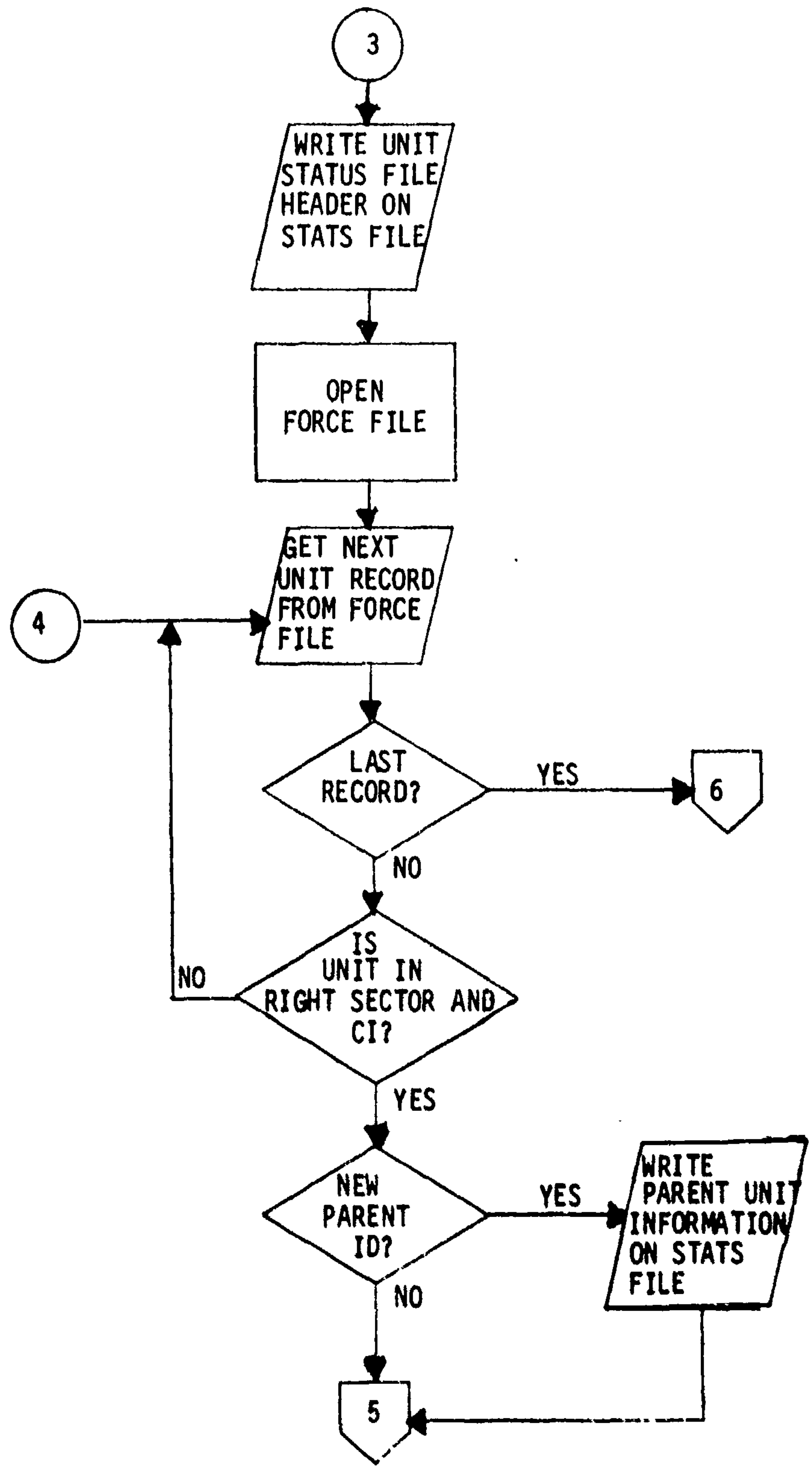

Figure 23. APPORT flow diagram (continued).

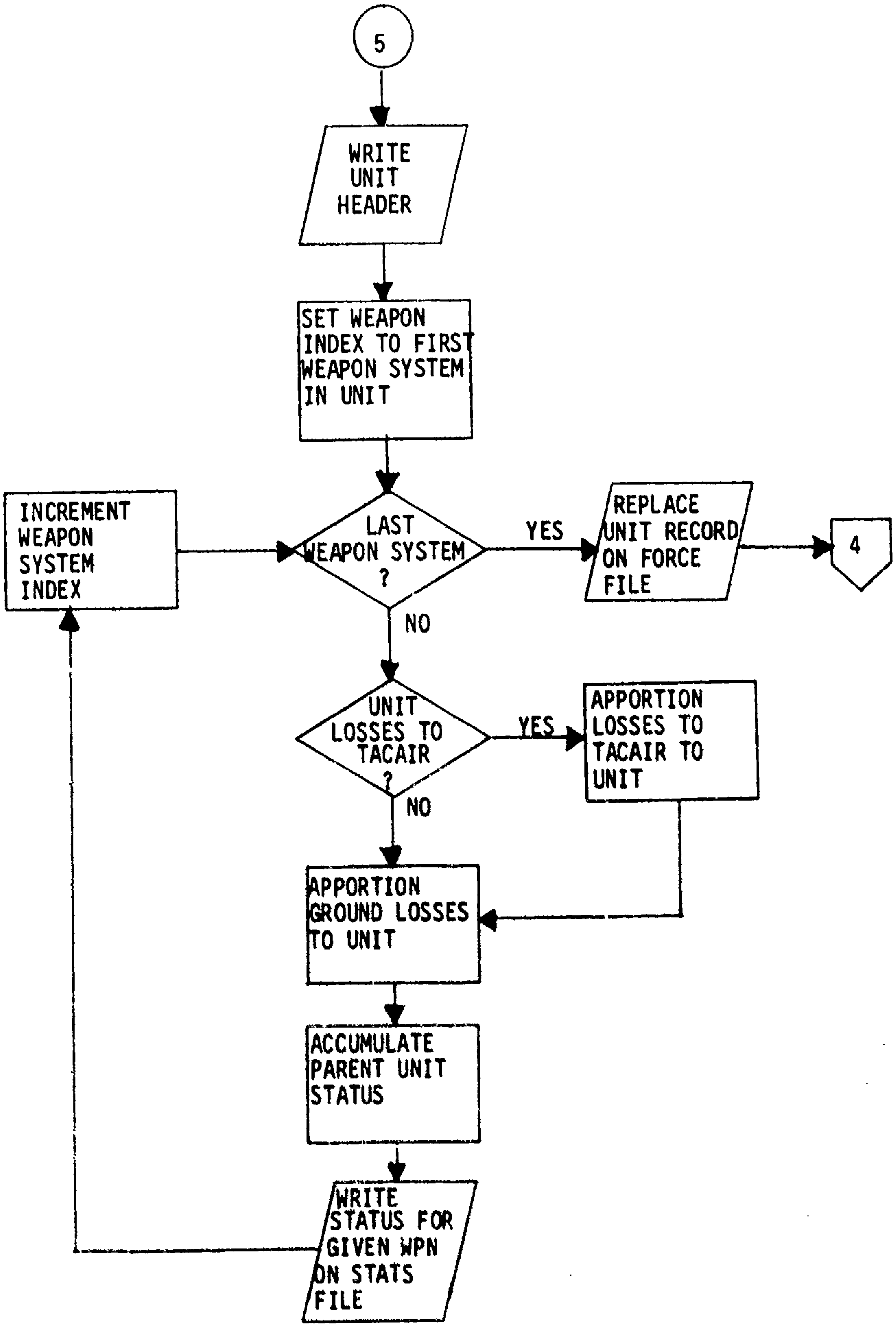

Figure 23. APPORT flow diagram (continued).

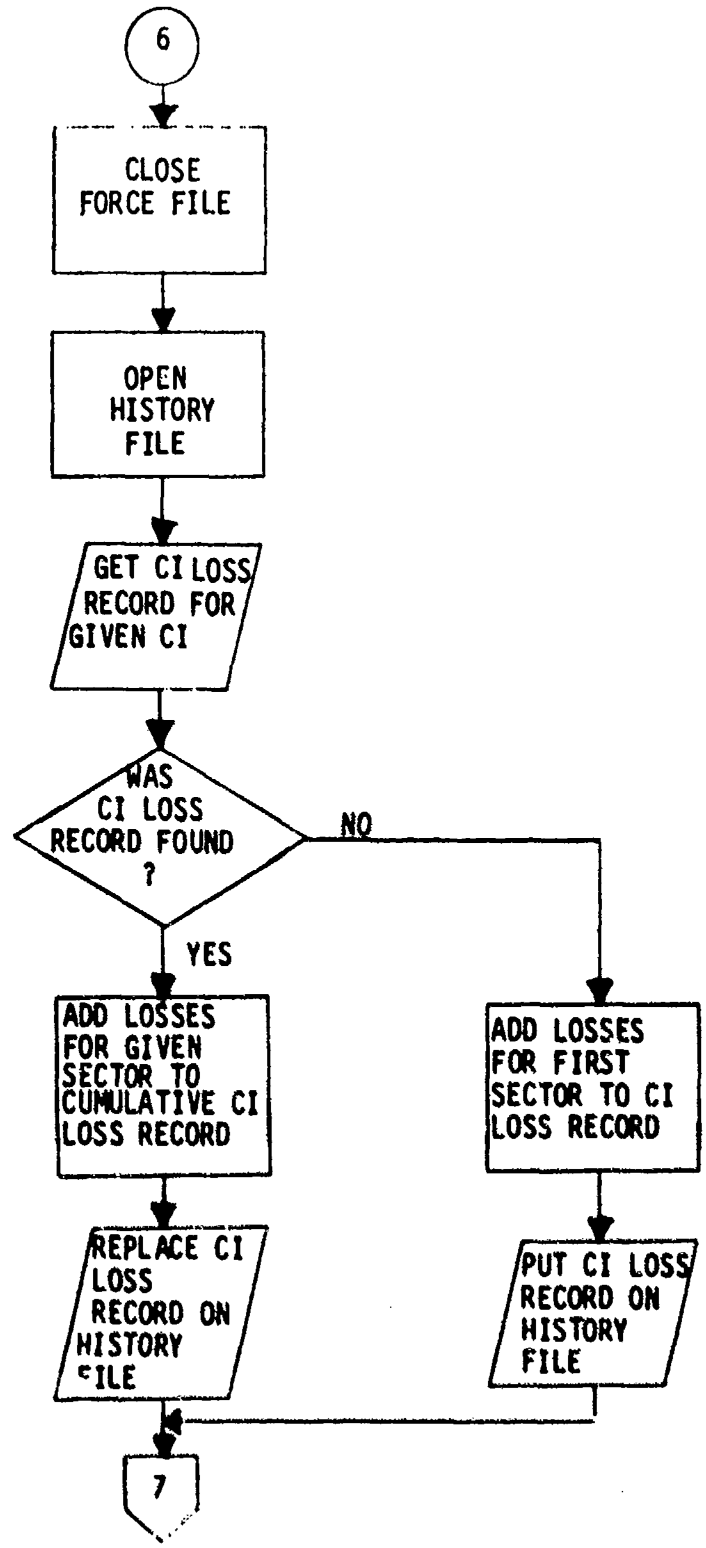

Figure 23. APPORT flow diagram (continued).

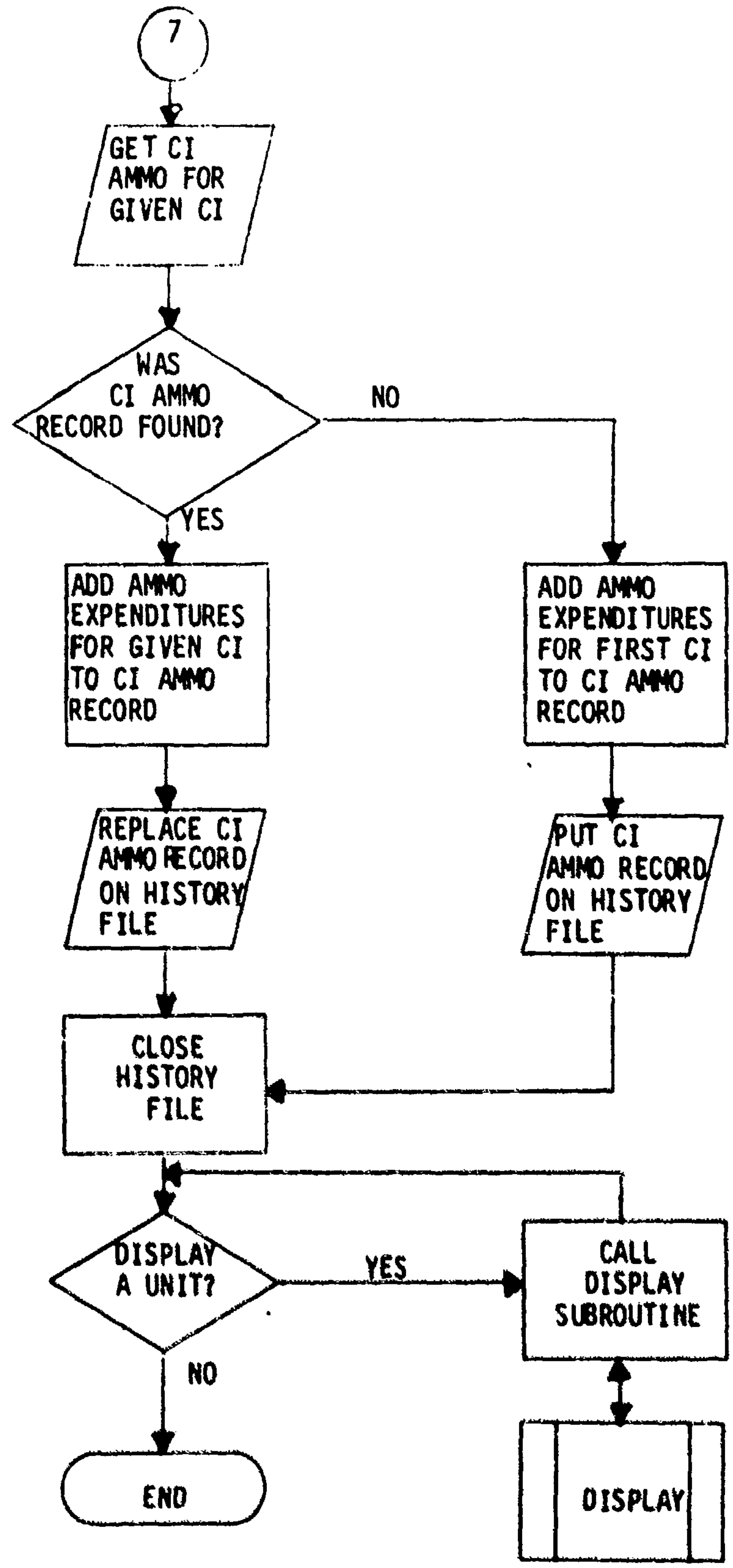

Figure 23. APPORT flow diagram (concluded).

APPENDIX A

INDEXED SEQUENTIAL FILE CREATION PROGRAMS

APPENDIX A

INDEXED SEQUENTIAL FILE CREATION PROGRAMS

This appendix contains the program code listings of the five programs used to create the indexed sequential-random access files used in the CACDA "Jiffy" War Gaming process. The FORTRAN code listings are presented in figures A-1 through A-5 for the SRC, UNIT, PARENT, FORCE, and HISTORY files, respectively.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(46)
CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,460,2LRL,460,
,3LMRL,460,2LKA,IARRAY(1),2LKP,0,2LKL,28,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
CALL PUT(IFIT)
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-1. Create program for SRC file.

```
PROGRAM CREATE(INPUT,OUTPUT)
DIMENSION IFIT(35),IARRAY(24)
CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,240,2LRL,240,
.3LMRL,240,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
CALL STOREF(IFIT,3LERL,100)
CALL OPENM(IFIT,3LNEW)
CALL PUT(IFIT)
CALL CLOSEM(IFIT)
STOP 123
END
```

Figure A-2. Create program for UNIT file.

```
 PROGRAM CREATE(INPUT,OUTPUT)
 DIMENSION IFIT(35),IARRAY(20)
 CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,200,2LRL,200,
.3LMRL,200,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LDKI,2LNO)
 CALL STOREF(IFIT,3LERL,100)
 CALL OPENM(IFIT,3LNEW)
 CALL PUT(IFIT)
 CALL CLOSEM(IFIT)
 STOP 123
 END
```

Figure A-3. Create program for PARENT file.

```
      PROGRAM CREATE(INPUT,OUTPUT)
      DIMENSION IFIT(35),IARRAY(90)
      CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,900,2LRL,900,
     .3LMRL,900,2LKA,IARRAY(1),2LKP,0,2LKL,20,3LOKI,2LNO)
      CALL STOREF(IFIT,3LERL,100)
      CALL OPENM(IFIT,3LNEW)
      IARRAY(1)=IARRAY(2)=IARRAY(5)="INITIAL"
      IARRAY(4)=0
      IARRAY(3)="B"
      DO 10 I=6,90
   10 IARRAY(I)=0
      CALL PUT(IFIT,IARRAY,900,IARRAY(1))
      CALL CLOSEM(IFIT)
      STOP 123
      END
```

Figure A-4. Create program for FORCE file.

```
      PROGRAM CREATE(INPUT,OUTPUT)
      DIMENSION IFIT(35),IARRAY(90)
      CALL FILEIS(IFIT,3LLFN,5LTAPE9,3LWSA,IARRAY,3LMNR,900,2LRL,900,
     .3LMRL,900,2LKA,IARRAY(1),2LKP,0,2LKL,30,3LDKI,2LNO)
      CALL STOREF(IFIT,3LERL,100)
      CALL OPENM(IFIT,3LNEW)
      IARRAY(1)="INITIAL"
      IARRAY(2)="INITIAL"
      IARRAY(3)="INITIAL"
      DO 10 I=4,90
10    IARRAY(I)=0
      CALL PUT(IFIT,IARRAY,900,IARRAY(1))
      CALL CLOSEM(IFIT)
      STOP 123
      END
```

Figure A-5. Create program for HISTORY file.

APPENDIX B

SRC PROGRAM LISTING

APPENDIX B

SRC PROGRAM LISTING

A listing of the FORTRAN code of the SRC program with a list of the program parameters is contained in this appendix. The list of variables is presented in table B-1. The FORTRAN source code of the program is given in figure B-1.

Table B-1. List of variables for SRC program.

Variable	Description
ACHG	Weapons to change (delete)
AERR	Weapons not found to change (delete)
AHOLD	Keeps the force type
AJ	Quantity of weapon to be added
AM	Weapon to be added
ARRAY	Work storage array (SRC File)
ARRAY (1)	Force type (key)
ARRAY (2)	SRC name (key)
ARRAY (3)	First weapon on record
ASRC	SRC name specified
ICK	Action code
IDO	Weapon listed
IEND	Number of weapons to be changed (deleted)
IFIT	FIT array (SRC File)
K	Number of weapons not found
NN	Weapon position on record
NY	Answer to question

```
      PROGRAM BUILD(INPUT,OUTPUT,TAPE9,TAPE6=INPUT)
      COMMON/ONE/IFIT(35),IFLG(1)
      DIMENSION ARRAY(46),ACHG(44),AERR(22),MYBUF(1024)
   10 ICK=0
   20 CALL FILMTS(IFIT,7LLFN,5LTAPE9,2LKA,ARRAY,2LPM,1LF,
     .4LFNT,3LYFC,7LREC,1024,3LFNM,MYBUF)
      CALL CONNEC(FLTABLE)
      IF(ICK.EQ.1HD) GO TO 9
      IF(ICK.EQ.1HL) GO TO 9
      ARRAY(1)="[illegible]"
  101 FORMAT(1A10)
      AHOLD=ARRAY(1)
    9 ARRAY(1)=AHOLD
      IF(ICK.EQ.1HD) GO TO [illegible]
      IF(ICK.EQ.1HL) GO TO 14
   10 DO 11 I=2,46
      ARRAY(I) = 0
   11 CONTINUE
      DO 12 I = 1,44
      ACHG(I) = 0
   12 CONTINUE
      DO 17 I = 1,22
      AERR(I) = 0
   17 CONTINUE
C
C                ABOVE DO LOOPS ZERO OUT WORK ARRAYS
C
   14 PRINT 102
  102   FORMAT(1X,"ENTER ACTION TYPE( X FOR LIST)- ")
  111   FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,
       11X,"R= READ (REVIEW) A RECORD",/,
       21X,"A= ADD A NEW SRC ",/,
       31X,"C= CHANGE/ADD WRN ID'S/QTY'S WITHIN AN EXISTING SRC",/,
       41X,"D= DELETE AN SRC AND/OR WRN SYS ID WITHIN THE SRC ",/,
       51X,"L= LIST ALL SRC'S ON FILE",/,
       61X,"E= END THE PROGRAM")
        READ(6,103) ICK
  103 FORMAT(1A1)
      IF(ICK.EQ.1HX) PRINT 111
      IF(ICK.EQ.1HX) GO TO 14
      IF(ICK.EQ.1HR) GO TO 510
      IF(ICK.EQ.1HA) GO TO 600
      IF(ICK.EQ.1HC) GO TO 710
      IF(ICK.EQ.1HD) GO TO 800
      IF(ICK.EQ.1HL) GO TO 1000
      IF(ICK.EQ.1HE) GO TO 900
      PRINT 104
  104 FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN")
      GO TO 14
  500 PRINT 501
  501 FORMAT(1X,"READ-ENTER SRC(END TO EXIT)-    ")
      READ(6,502)  ASRC
      CALL OPENM(IFIT,3LI-O)
  502 FORMAT(1A10)
      IF(ASRC.EQ.3HEND) GO TO 16
      ARRAY(2) = ASRC
      ARRAY(3) = 99919
```

Figure B-1. SRC program code (continued next page).

```
      CALL GET(IFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).EQ.99999) GO TO 550
      PRINT 503 ,ARRAY(2)
  503 FORMAT(1X,"SRC=",A10,5X," ID  QTY")
      DO 505 I=3,45,2
      IF (ARRAY(I).EQ. 0)  GO TO 505
      IDD=ARRAY(I)
      PRINT 504, ( IDD,ARRAY(I+1)  )
  504 FORMAT(20X,I3,1X,F5.0)
  505 CONTINUE
      CALL CLOSEM(IFIT)
      GO TO 500
  550 PRINT 551 , ASRC
  551 FORMAT(1X,"SRC ",A10," NOT ON FILE")
      CALL CLOSEM(IFIT)
      GO TO 500
 16   CALL CLOSEM(IFIT)
      GO TO 10
  600 PRINT 601
 601  FORMAT(1X,"ADD=ENTER NEW SRC(END TO EXIT)--  ")
      CALL OPENM(IFIT,3LI-O)
      READ(6,502) ASRC
      IF(ASRC.EQ.3HEND) GO TO 16
      ARRAY(2)=ASRC
      ARRAY(3) = 99999
      CALL GET(IFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).NE.99999) GO TO 610
 7000 FORMAT(1X,"ENTER TOTAL NO. OF WPN SYS IDS ",/)
      NN=1
      PRINT 7001
 7001 FORMAT(1X,"ENTER WPN ID,QTY--0,0 IF DONE ")
  887 READ(6,*) AM,AJ
      IF(AM.EQ.0) GO TO 888
      NN=NN+2
      ARRAY(NN)=AM
      ARRAY(NN+1)=AJ
      PRINT*,"NEXT-"
      GO TO 887
  888 CONTINUE
      CALL PUT(IFIT,ARRAY,469,ARRAY(1))
      M=IFETCH(IFIT,5LIFC)
      IF (M.EQ.446B)  GO TO 610
  607 FORMAT(1X,"SRC",A10," ALREADY ON FILE")
      GO TO 612
  610 PRINT 607, ASRC
  612 DO 611 I=2,46
      ARRAY(I)=0
  611 CONTINUE
      CALL CLOSEM(IFIT)
      GO TO 600
  700 PRINT 701
  701 FORMAT(1X,"CHANGE-ENTER SRC(END TO EXIT)-- ")
      READ(6,702) ASRC
      CALL OPENM(IFIT,3LI-O)
  702 FORMAT(A10)
      IF(ASRC.EQ.3HEND) GO TO 16
      ARRAY(3) = 99999
```

Figure B-1. SRC program code (continued).

```
      ARRAY(2) = ASRC
      CALL GET(IFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).NE. 99999)  GO TO 750
  700 PRINT 707 , ARRAY(2)
  707 FORMAT(1X,"SRC ",A10," NOT ON FILE")
      CALL CLOSEM(IFIT)
      GO TO 795
  750 PRINT 7030
      READ(6,*) IEND
      IEND=IEND*2
      PRINT 7002
 7002 FORMAT(1X,"ENTER WRN ID,QTY,ID,QTY,---",/)
      READ(5,*) (ACHG(I),I=1,IEND)
      K=0
      DO 780 J=1,IEND,2
      IF (ACHG(J).EQ.  0)  GO TO 780
      DO 770 I=3,45,2
      IF(ARRAY(I).EQ.ACHG(J))  GO TO 776
  770 CONTINUE
      K=K+1
      ALER(K) = ACHG(J)
      DO 775 L=3,45,2
      IF(ARRAY(L).NE.0)  GO TO 775
      ARRAY(L)=ACHG(J)
      ARRAY(L+1)=ACHG(J+1)
      GO TO 780
  775 CONTINUE
      GO TO 780
  776 ARRAY(I+1) = ACHG(J+1)
  780 CONTINUE
  790 CALL REPLC(IFIT,ARRAY,45,ARRAY(1))
      IF (K.EQ.0)  GO TO 791
      PRINT 796, ASRC, ( ALER(I) , I=1,K  )
  796 FORMAT(1X,"FOLLOWING WRN IDS NOT FOUND FOR SRC",A10,/,
     *1X,"THEY WERE ADDED TO THE SRC RECORD",/,22(F5.0))
  791 CALL CLOSEM(IFIT)
      GO TO 790
  800 PRINT 801
  801 FORMAT(1X,"DELETE-ENTER SRC(END TO EXIT)-- ")
      READ(6,1255) ASRC
      CALL OPENM(IFIT,3LI-O)
 1255 FORMAT(A10)
      IF(ASRC.EQ.3HEND) GO TO 15
      PRINT 7003
 7003 FORMAT(" ENTER TOTAL NO. OF WRN SYSTEMS TO BE DELETED-",
     *"ENTER 0 IF ALL ")
      READ(6,*) IEND
      ARRAY(2)=ASRC
      ARRAY(3)=99999
      CALL GET(IFIT,ARRAY, ARRAY(1))
      IF(ARRAY(3).EQ.99999) GO TO 845
      IF(IEND.EQ.0) GO TO 890
      PRINT*," ENTER WRN SYS IDS TO BE DELETED  "
      READ(6,*) (ACHG(I),I=1,IEND)
      IF(ARRAY(3).NE.99999) GO TO 850
  845 PRINT*," SRC   ",ARRAY(2),"   NOT ON FILE  "
      CALL CLOSEM(IFIT)
```

Figure B-1. SRC program code (continued).

Best Available Copy

```
      GO TO 900
  980 IF(ACHG(1).EQ. 0 )  GO TO 990
      K=0
      DO 880 J=1,IEND
      IF(ACHG(J).EQ.0)   GO TO 880
      DO 860  I=3,45,2
      IF (ARRAY(I).EQ.0))  GO TO 860
      IF(ARRAY(I).EQ.ACHG(J) )    GO TO 866
  860 CONTINUE
      K=K+1
      ASTE(K) = ACHG(J)
      GO TO 880
  866 ARRAY(I) = 0
      ARRAY(I+1) = 0
  880 CONTINUE
      CALL REPLC(IFIT,ARRAY,460,ARRAY(1))
      IF(K.EQ.0)   GO TO 991
      PRINT 8000,ASTC,(ASTE(I),I=1,K)
 8000 FORMAT(1X,"THE FOLLOWING WEN I.D.'S WERE NOT FOUND FOR SRC"
     F,1X,A10,/,1X,22(F5.0))
  991    CALL CLOSEM(IFIT)
      GO TO 900
  990 CALL DLTE(IFIT,ARRAY(1))
      CALL CLOSEM(IFIT)
      GO TO 900
 1000    CALL OPENM(IFIT,"LI-O")
 1100    CONTINUE
      CALL GETN(IFIT,ARRAY,ARRAY(1))
      M=IFETCH(IFIT,"FP")
      IF(M.EQ.100) GO TO 15
      IF(AHOLD.EQ.ARRAY(1)) GO TO 1200
      GO TO 1100
   15    CALL CLOSEM(IFIT)
      GO TO 10
 1200    PRINT 601,ARRAY(2)
      DO 1205 I=3,45,2
      IF(ARRAY(I).EQ.0) GO TO 1205
      IDD=ARRAY(I)
      PRINT 604,IDD,ARRAY(I+1)
 1205    CONTINUE
      DO 1210 I=2,45
      ARRAY(I)=0
 1210    CONTINUE
      GO TO 1150
  900    PRINT 9000
 9000 FORMAT(1X,"ANY MORE    STRUCTURES TO BE UPDATED? ",/,)
      READ(5,9001) MY
 9001    FORMAT(A1)
      IF(MY.EQ.1HY) GO TO 10
      PRINT 901
  901    FORMAT(1X,"ALL DONE  JOB HAS ENDED")
      STOP
      END
```

Figure B-1. SRC program code (concluded).

APPENDIX C

UNIT PROGRAM LISTING

APPENDIX C

UNIT PROGRAM LISTING

This appendix contains the FORTRAN code and a list of the variables of the UNIT program. The list of variables is contained in table C-1. The FORTRAN program is presented in figure C-1.

Table C-1. List of variables for UNIT program.

Variable	Description
AERR	SRC's already existing in unit
ARRAY	Work storage area (SRC File)
ARRAY (1)	Force type (key)
ARRAY (2)	SRC name (key)
ARRAY (3)	First weapon on record
BBERR	SRC's which do not exist
BCHG	SRC's to be added (deleted)
BERR	SRC's to be added
BFRC	SRC not on file
BHOLD	Keeps the force type
BRRAY	Work storage area (Unit File)
BRRAY (1)	Force type (key)
BRRAY (2)	Unit Name (key)
BRRAY (3)	First SRC on record
BUNIT	Unit name specified
I	SRC position on record
ICK	Action code
IFIT	FIT array (SRC File)
K	Number of SRC's not found
M	Number of SRC's already existing
N	Number of SRC's to be added
NEND	Number of SRC's to be changed (deleted)
NFIT	FIT array (Unit File)
NN	Number of SRC's which do not exist
NY	Answer to question

```
      PROGRAM UNIT(INPUT,OUTPUT,TAPE10,TAPE9,TAPE6=INPUT)
      COMMON/ONE/IFIT(35),UFIT(35)
      DIMENSION ARRAY(24),ARRAY(46),BCHG(22),
     1ARRE(22),RUN(22),ADD(22),MYBUF(1024),NYBUF(1024)
      ICK=0
      CALL FILETT(UFIT,3LLFN,6LTAPE10,2LKA,ARRAY(1),2LPH,1LR,
     ,3LFWI,3LYES,3LDFS,1024,3LFWB,MYBUF)
      CALL FILEIS(IFIT,3LLFN,6LTAPE9,2LKA,ARRAY(1),2LPH,1LR,
     ,3LFWI,3LYES,3LDFS,1024,3LFWB,MYBUF)
      IF(ICK.EQ.1HE) GO TO 9
      IF(ICK.EQ.1HL) GO TO 9
      ARRAY(1)="ONE"
  101 FORMAT(A10)
      ARRAY(1)=ARRAY(1)
      BHOLD=ARRAY(1)
    9 ARRAY(1)=ARRAY(1)=BHOLD
      IF(ICK.EQ.1HE) GO TO 500
      IF(ICK.EQ.1HL) GO TO 14
  111 FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,
     11X,"R=READ(REVIEW) A RECORD",/,
     21X,"A=ADD A NEW UNIT",/,
     31X,"C=ADD SEC,S WITHIN AN EXISTING UNIT",/,
     41X,"D=DELETE A UNIT AND/OR SEC,S WITHIN THE UNIT",/,
     51X,"L=LIST ALL UNITS ON FILE",/,
     61X,"E=END THE PROGRAM")
   10 DO 11 I=2,24
      ARRAY(I)=0
      IF(I.GT.22) GO TO 11
      BCHG(I)=0
      ADD(I)=0
   11 CONTINUE
C
C     ABOVE DO LOOP ZERO OUT WORK ARRAYS
C
   14 PRINT 102
  102 FORMAT(1X,"ENTER ACTION TYPE( X FOR LIST)--")
      READ(6,103) ICK
  103 FORMAT(A1)
      IF(ICK.EQ.1HX) PRINT 111
      IF(ICK.EQ.1HX) GO TO 14
      IF(ICK.EQ.1HR) GO TO 500
      IF(ICK.EQ.1HA) GO TO 600
      IF(ICK.EQ.1HC) GO TO 700
      IF(ICK.EQ.1HD) GO TO 800
      IF(ICK.EQ.1HL) GO TO 1000
      IF(ICK.EQ.1HE) GO TO 900
      PRINT 104
  104 FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN ")
      GO TO 14
C
C     **THIS PORTION OF PROGRAM IS TO READ(REVIEW) UNITS**
C
  500 PRINT 501
  501 FORMAT(1X,"READ-ENTER UNIT ID(END TO EXIT)--    ")
      READ(6,502) BUNIT
      CALL OPENM(UFIT,3LI-O)
  502 FORMAT(A10)
```

Figure C-1. UNIT program code (continued next page).

```
      IF(RUNIT.EQ.THEND) GO TO 11
      ARRAY(2)=RUNIT
      ARRAY(3)=99999
      CALL GET(NFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).EQ.99999) GO TO 550
      PRINT 503,ARRAY(2)
  503 FORMAT(1X,"UNIT=",A10,5X,"REC")
      DO 505 I=3,24
      IF(ARRAY(I).EQ.0.0) GO TO 505
      PRINT 504,(ARRAY(I))
  504 FORMAT(17X,A10)
  505 CONTINUE
      CALL CLOSEM(NFIT)
      GO TO 500
  550 PRINT 551,RUNIT
  551 FORMAT(1X,"UNIT",A10,"NOT ON FILE")
      CALL CLOSEM(LFIT)
      PRINT 1111,ARRAY(2)
C
C     **THIS PORTION OF PROGRAM IS TO ADD A NEW UNIT**
C
  600 PRINT 601
  601 FORMAT(1X,"ADD--ENTER NEW UNIT ID(END TO EXIT)-- ")
      READ(5,502)RUNIT
      CALL OPENM(NFIT,FLI=0)
      IF(RUNIT.EQ.THEND) GO TO 16
      ARRAY(2)=RUNIT
      ARRAY(3)=99999
      CALL GET(NFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).NE.99999) GO TO 610
 6000 FORMAT(1X,"ENTER TOTAL NO. OF REC.",/)
      I=0
      CALL OPENM(IFIT,FLI=0)
      PRINT 606
  606 FORMAT(1X,"ENTER REC(0 IF DONE)--")
 6061 I=I+1
      READ(5,502) ARRAY(I+2)
      IF(ARRAY(I+2).NE."0") GO TO 6062
      ARRAY(I+2)=0
      CALL CLOSEM(IFIT)
      GO TO 616
 6062 CONTINUE
      ARRAY(2)=ARRAY(I+2)
      ARRAY(3)=99999
      CALL GET(IFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).NE.99999) GO TO 607
      ARRAY=ARRAY(2)
      GO TO 606
  607 CONTINUE
      PRINT*,"REENTER"
      GO TO 6061
  610 CALL PUT(NFIT,ARRAY,240,ARRAY(1))
      NO=IFETCH(NFIT,3LIES)
      IF(NO.EQ.0,LE0) GO TO 611
  613 FORMAT(1X,"UNIT ",A10," ALREADY ON FILE")
      GO TO 612
  611 PRINT 613,RUNIT
```

Figure C-1. UNIT program code (continued).

```
      GO TO 612
  606 PRINT 6002, BEFC
 6002 FORMAT(1X,"SPEC ",A10," NOT ON FILE AND NOT ADDED ")
      I=I-1
      GO TO 602
  612 DO 611 I=3,24
      BRRAY(I)=0
  611 CONTINUE
      CALL CLOSEM(NFIT)
      GO TO 600
 16   CALL CLOSEM(NFIT)
      GO TO 10
C
C   **THIS PORTION OF PROGRAM ADDS SPECS TO AN EXISTING UNIT**
C
  700 DO 702 I=1,22
      NEW(I)=0
      ARRAY(I)=0
      AREF(I)=0
      BCHG(I)=0
  702 CONTINUE
      PRINT 701
  701 FORMAT(1X,"CHANGE-ENTER UNIT ID(END TO EXIT)-- ")
      READ(6,502) BUNIT
      CALL OPENM(NFIT,3LI-O)
      IF(BUNIT.EQ.THEND) GO TO 16
      BRRAY(1)=99999
      BRRAY(2)=BUNIT
      CALL GET(NFIT,BRRAY,BRRAY(1))
      IF(BRRAY(3).NE.99999) GO TO 750
  700 PRINT 703,BRRAY(2)
  703 FORMAT(1X,"UNIT ",A10," NOT ON FILE")
      CALL CLOSEM(NFIT)
      GO TO 700
  750 PRINT 6000
      READ(6,*) NEND
      DO 751 I=1,NEND
      PRINT*,"ENTER SPEC NO. ",I," --"
      READ(6,502) BCHG(I)
  751 CONTINUE
      N=0
      M=0
C
C   **ARRAY NEW KEEPS THOSE SPECS TO BE ADDED**
C   **ARRAY AREF KEEPS THOSE SPECS ALREADY EXISTING IN THIS UNIT**
C   **ARRAY BREF KEEPS THE SPECS IN AREF WHICH DO NOT EXIST**
C
      DO 780 J=1,NEND
      IF(BCHG(J).EQ.0) GO TO 780
      DO 770 I=3,24
      IF(BRRAY(I).EQ.BCHG(J)) GO TO 776
  770 CONTINUE
      N=N+1
      NEW(N)=BCHG(J)
      GO TO 780
  776 M=M+1
      AREF(M)=BCHG(J)
```

Figure C-1. UNIT program code (continued).

```
 7?? CONTINUE
     NN=0
     CALL OPENM(IFIT,3LI-O)
     DO 705 I=1,N
     ARRAY(2)=R [illegible](I)
     ARRAY(3)=99999
     CALL GET(IFIT,ARRAY,ARRAY(1))
     IF(ARRAY(3).NE.99999) GO TO 715
     NN=NN+1
     [illegible](NN)=[illegible](I)
     GO TO 705
 715 DO 7?5 JK=3,26
     IF(ARRAY(JK).NE.0) GO TO 7?5
     ARRAY(JK)=[illegible](I)
     GO TO 7?5
 7?5 CONTINUE
 7?5 CONTINUE
     CALL CLOSEM(IFIT)
     CALL REPLC(IFIT,ARRAY,26,ARRAY(1))
 7?? IF(N.EQ.0) GO TO 720
     PRINT 7??,UNIT
 7?? FORMAT(1X,"FOLLOWING REC,S WERE ALREADY PRESENT IN",
    1" UNIT ",A10)
     DO 7100 I=1,N
     PRINT 6?5,[illegible](I)
 6??  FORMAT(12X,A10)
 7100 CONTINUE
 720 IF(NN.EQ.0) GO TO 7201
 7?5 PRINT 7???
 7??? FORMAT(1X,"FOLLOWING REC,S NOT FOUND ON TOE-FILE",/,
    21X,"THE RECORD WERE NOT ADDED ")
     DO 7200 I=1,NN
     PRINT 6?5,[illegible](I)
 7200 CONTINUE
 7201  CALL CLOSEM(IFIT)
     GO TO [illegible]
C
C    **THIS PORTION OF THE PROGRAM DELETES AN ENTIRE UNIT OR**
C    **DELETES REC'S WITHIN A UNIT**
C
 ?00 PRINT ?01
 ?01 FORMAT(1X,"DELETE-ENTER UNIT ID(END TO EXIT)-- ")
     READ(5,8000) RUNIT
     CALL OPENM(IFIT,3LI-O)
 8000 FORMAT(A10)
     IF(RUNIT.EQ.4HEND) GO TO 1?
     PRINT 8001
 8001 FORMAT(1X,"[illegible] TOTAL NO. OF REC,S TO BE DELETED-",
    3[illegible] TO DELETE THE ENTIRE UNIT ")
     READ(5,*) [illegible]
     ARRAY(2)=RUNIT
     ARRAY(3)=99999
     CALL GET(IFIT,ARRAY,ARRAY(1))
     IF(ARRAY(3).EQ.99999) GO TO ?40
     IF([illegible].[illegible].0) GO TO ?9?
     DO 8??? I=1,[illegible]
     PRINT*,"[illegible] REC NO. ",I," --"
```

Figure C-1. UNIT program code (continued).

```
      READ(6,502) SCHG(I)
 8500 CONTINUE
      GO TO 850
 845  PRINT1111,RUNIT
 1111 FORMAT(1X,A10," NOT ON FILE")
      CALL CLOSEM(NEFIT)
      GO TO 850
  855 IF(SCHG(1).EQ.0) GO TO 890
      K=0
      DO 880 J=1,NEND
      IF(SCHG(J).EQ.0) GO TO 880
      DO 860 I=3,24
      IF(ARRAY(I).EQ.1) GO TO 860
      IF(ARRAY(I).EQ.SCHG(J)) GO TO 866
  860 CONTINUE
      K=K+1
C
C    **ASTC HERE KEEPS TRACK OF THE SECS NOT FOUND FOR THE UNIT**
C
      ASTC(K)=SCHG(J)
      GO TO 880
  866 ARRAY(I)=0
  880 CONTINUE
      CALL REPLC(NEFIT,ARRAY,240,ARRAY(1))
      IF(K.EQ.0) GO TO 9101
      PRINT 8002,RUNIT
 8002 FORMAT(1X,"THE FOLLOWING SECS WERE NOT FOUND FOR UNIT ",
     *A10,1X)
      DO 9100 I=1,K
      PRINT 505,ASTC(I)
 9100 CONTINUE
 9101 CALL CLOSEM(NEFIT)
      GO TO 200
  890 CALL DLTE(NEFIT,ARRAY(1))
      CALL CLOSEM(NEFIT)
      GO TO 850
C
C    **THIS PORTION OF TH  PROGRAM LISTS ALL UNITS PRESENTLY ON FILE**
C
 1000 CALL OPENM(NEFIT,3LI-O)
 1100 CONTINUE
      CALL GETN(NEFIT,ARRAY,ARRAY(1))
      M=IFETCH(NEFIT,3LFP)
      IF(M.EQ.100B)GO TO 15
      IF(3HOLD.EQ.ARRAY(1)) GO TO 1200
      GO TO 1100
   15 CALL CLOSEM(NEFIT)
      GO TO 20
 1200 PRINT 513,ARRAY(2)
      DO 1205 I=3,24
      IF(ARRAY(I).EQ.0) GO TO 1205
      PRINT 514, ARRAY(I)
 1205 CONTINUE
      DO 1206 I=3,24
      ARRAY(I)=0
 1206 CONTINUE
      GO TO 1100
```

Figure C-1. UNIT program code (continued).

```
[illegible] PRINT 9000
9000 FORMAT(1X,"ANY MORE FORCE STRUCTURES TO BE UPDATED? ",/)
     READ(5,9001) NY
9001 FORMAT(A1)
     IF(NY.EQ.1HY) GO TO 1[illegible]
     PRINT 901
 901 FORMAT(1X," ALL FORCE JOB HAS ENDED ")
     STOP
     END
```

Best Available Copy

Figure C-1. UNIT program code (concluded).

APPENDIX D

PARENT PROGRAM LISTING

APPENDIX D

PARENT PROGRAM LISTING

Appendix D contains a list of the variables used in and a listing of the FORTRAN source code for the PARENT program. A list of the program variables is given in table D-1, and the program code is presented in figure D-1.

Table D-1. List of variables for PARENT program.

Variable	Description
BBERR	Units which do not exist
BERR	Units which are to be added
BRRAY	Work storage area (Unit File)
BRRAY (1)	Force type (Key)
BRRAY (2)	Unit Name (Key)
BRRAY (3)	First SRC on record
CCHG	Units to be added (deleted)
CERR	Units already existing
CFRC	Parent specified
CHOLD	Keeps the force type
CRRAY	Work storage area (PARENT file)
CRRAY (1)	Force Type (Key)
CRRAY (2)	Parent name (Key)
CRRAY (3)	First unit on record
I	Unit position on record
ICK	Action code
K	Number of units not found
LEND	Number of units to be changed (deleted)
LFIT	FIT Array (Parent File)
M	Number of units already existing
N	Number of units to be added
NFIT	FIT array (Unit File)
NN	Number of units which do not exist
NY	Answer to question

```
      PROGRAM PARENT(INPUT,OUTPUT,TAPE11,TAPE10,TAPE5=INPUT)
      COMMON/UNIT/UNIT(35),LEFT(35)
      DIMENSION ARRAY(20),BARRAY(24),ICHG(15),
     1CR(18),PER(18),B3LR(18),NYBUF(1024),MYBUF(1024)
      ICK=0
      CALL FILEIS(LEFT,3LLFN,6LTAPE11,2LKA,ARRAY(1),2LPM,1LR,
     .3LFWI,3LYES,3LBFS,1024,3LFWB,MYBUF)
      CALL FILEIS(LEFT,3LLFN,6LTAPE10,2LKA,BARRAY(1),2LPM,1LR,
     .3LFWI,3LYES,3LBFS,1024,3LFWB,NYBUF)
      IF(ICK.EQ.1HC) GO TO 9
      IF(ICK.EQ.1HL) GO TO 9
      ARRAY(1)="  "
  101 FORMAT(A10)
    9 BARRAY(1)=ARRAY(1)
      CHOLD=ARRAY(1)
      BARRAY(1)=ARRAY(1)=CHOLD
      IF(ICK.EQ.1HC) GO TO 830
      IF(ICK.EQ.1HL) GO TO 14
  111 FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,
     11X,"R=READ(REVIEW) A RECORD",/,
     21X,"A=ADD A NEW PARENT",/,
     31X,"C=ADD UNIT,S WITHIN AN EXISTING PARENT",/,
     41X,"D=DELETE A PARENT AND/OR UNIT,S WITHIN THE PARENT",/,
     51X,"L=LIST ALL PARENTS ON FILE",/,
     61X,"E=END THE PROGRAM")
   10 DO 11 I=2,20
      ARRAY(I)=0
      IF(I.GT.15) GO TO 11
      ICHG(I)=0
      CR(I)=0
   11 CONTINUE
C
C     ABOVE DO LOOP ZERO OUT WORK ARRAYS
C
   14 PRINT 102
  102 FORMAT(1X,"ENTER ACTION TYPE( X FOR LIST)-- ")
      READ(5,103) ICK
  103 FORMAT(A1)
      IF(ICK.EQ.1HX) PRINT 111
      IF(ICK.EQ.1HX) GO TO 14
      IF(ICK.EQ.1HR) GO TO 500
      IF(ICK.EQ.1HA) GO TO 600
      IF(ICK.EQ.1HC) GO TO 700
      IF(ICK.EQ.1HD) GO TO 800
      IF(ICK.EQ.1HL) GO TO 1500
      IF(ICK.EQ.1HE) GO TO 900
      PRINT 104
  104 FORMAT(1X,"ACTION CODE ERROR-TRY AGAIN ")
      GO TO 14
C
C     **THIS SECTION OF PROGRAM IS TO READ(REVIEW) PARENTS**
C
  500 PRINT 501
  501 FORMAT(1X,"READ-ENTER PARENT ID(END TO EXIT)-- ")
      READ(5,502) PERC
  502 FORMAT(A10)
      CALL OPENM(LEFT,3LI-O)
```

Figure D-1. PARENT program code (continued next page).

```
      IF(CFRC.EQ.3HEND) GO TO 16
      CFRAY(2)=CFRC
      CFRAY(3)=99999
      CALL GET(LFIT,CFRAY,CFRAY(1))
      IF(CFRAY(3).EQ.99999) GO TO 550
      PRINT 503,CFRAY(2)
  503 FORMAT(1X,"PARENT=",A10,5X,"UNIT")
      DO 505 I=3,20
      IF(CFRAY(I).EQ.0) GO TO 505
      PRINT 504,(CFRAY(I))
  504 FORMAT(17X,A10)
  505 CONTINUE
      CALL CLOSEM(LFIT)
      GO TO 500
  550 PRINT 551,CFRC
  551 FORMAT(1X,"PARENT",A10,"NOT ON FILE")
      CALL CLOSEM(LFIT)
      GO TO 500
16    CALL CLOSEM(LFIT)
      GO TO 10
C
C     **THIS PORTION OF PROGRAM IS TO ADD A NEW PARENT**
C
  600 PRINT 601
  601 FORMAT(1X,"ADD-ENTER NEW PARENT ID(END TO EXIT)-- ")
      READ(6,602)CFRC
      CALL OPENM(LFIT,3LI-O)
      IF(CFRC.EQ.3HEND) GO TO 16
      CFRAY(2)=CFRC
      CFRAY(3)=99999
      CALL GET(LFIT,CFRAY,CFRAY(1))
      IF(CFRAY(3).NE.99999) GO TO 610
 6000 FORMAT(1X,"ENTER TOTAL NO. OF UNIT,S",/)
      I=0
      CALL OPENM(NFIT,3LI-O)
      PRINT 606
606   FORMAT(1X,"ENTER UNIT(0 IF DONE)-")
6061  I=I+1
      READ(6,602) CFRAY(I+2)
      IF(CFRAY(I+2).NE."0") GO TO 6600
      CFRAY(I+2)=0
      CALL CLOSEM(NFIT)
      GO TO 615
C
C
6600  NE=0
      CFRAY(2)=CFRAY(I+2)
      CFRAY(3)=99999
      CALL GET(NFIT,CFRAY,CFRAY(1))
      IF(CFRAY(3).NE.99999) GO TO 602
      CF C=CFRAY(2)
      GO TO 616
  602 CONTINUE
      PRINT*,"NEXT="
      GO TO 6061
  615 CALL PUT(LFIT,CFRAY,200,CFRAY(1))
      NO=IFETCH(LFIT,7LIT C)
```

Figure D-1. PARENT program code (continued).

```
      IF(MO.EQ.4.F.) GO TO 610
607   FORMAT(1X,"PARENT ",A10," ALREADY ON FILE")
      GO TO 612
610   PRINT 607,CERAY(2)
      GO TO 612
606   PRINT 6002, CERC
6002  FORMAT(1X,"UNIT ",A10,"NOT ON FILE AND NOT ADDED ")
      I=I-1
      GO TO 602
612   DO 611 ILK=3,20
      CERAY(ILK)=0
611   CONTINUE
      CALL CLOSE(LFIT)
      GO TO 600
C
C     *THIS PORTION OF PROGRAM ADDS UNITS TO AN EXISTING PARENT**
C
700   DO 702 I=1,10
      REST(I)=0
      CCHK(I)=0
      CERR(I)=0
      CCHG(I)=0
702   CONTINUE
      PRINT 701
701   FORMAT(1X,"CHANGE-ENTER PARENT ID(END TO EXIT)-- ")
      READ(6,502) CERC
      CALL OPENM(LFIT,ZLI=0)
      IF(CERC.EQ.THEND) GO TO 15
      CERAY(3)=99999
      CERAY(2)=CERC
      CALL GET(LFIT,CERAY,CERAY(1))
      IF(CERAY(3).NE.99999) GO TO 750
790   PRINT 707,CERAY(2)
707   FORMAT(1X,"PARENT ",A10," NOT ON FILE")
      CALL CLOSEM(LFIT)
      GO TO 700
750   PRINT 6000
      READ(6,*) LEND
      DO 760 I=1,LEND
      PRINT*,"ENTER UNIT NO. ",I," --"
      READ(6,502) CCHG(I)
760   CONTINUE
      M=0
      N=0
C
C     *REST ARRAY K.... THOSE UNITS TO BE ADDED**
C     *CCHK ARRAY K.... THOSE UNITS ALREADY EXISTING IN THIS PARENT**
C     *CERR ARRAY .... THE UNITS IN RERR WHICH DO NOT EXIST**
C
      DO 780 J=1,LEND
      IF(CCHG(J).EQ.0.) GO TO 780
      DO 770 I=3,20
      IF(CERAY(I).EQ.CCHG(J)) GO TO 775
770   CONTINUE
      N=N+1
      REST(N)=CCHG(J)
      GO TO 780
```

Figure D-1. PARENT program code (continued).

```
[illegible] M=M+1
      [illegible](M)=[illegible](I)
 [illegible] CONTINUE
      NN=0
      CALL [illegible](NFIT,[illegible]LI-C)
      DO [illegible]05 I=1,N
      ARRAY(2)=[illegible](I)
      ARRAY([illegible])=99999
      CALL [illegible](NFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).[illegible].99999) GO TO [illegible]
      NN=NN+1
      [illegible](NN)=[illegible](I)
      GO TO [illegible]
 [illegible] DO [illegible] JK=3,[illegible]
      IF(ARRAY(JK).NE.0) GO TO [illegible]5
      ARRAY(JK)=[illegible](I)
      GO TO [illegible]
 [illegible] CONTINUE
 [illegible] CONTINUE
      CALL CLOSE[illegible](NFIT)
      CALL [illegible]LS(LFIT,ARRAY,[illegible],ARRAY(1))
 [illegible] IF([illegible].EQ.0) GO TO 719
      PRINT 708,[illegible]
 708 FORMAT(1X,"FOLLOWING UNIT,S WERE ALREADY PRESENT IN",
     1" PARENT ",A10)
      DO [illegible] I=1,M
      PRINT 605,[illegible](I)
 605  FORMAT(10X,A10)
 [illegible] CONTINUE
 [illegible] IF(NN.EQ.0) GO TO [illegible]
 [illegible] PRINT 7[illegible]
 [illegible] FORMAT(1X,"FOLLOWING UNIT,S NOT FOUND ON UNIT-FILE",/,
     21X,"[illegible] NOT ADDED ")
      DO 7[illegible] I=1,NN
      PRINT 605,[illegible](I)
 [illegible] CONTINUE
 7201  CALL CLOSE[illegible](LFIT)
      GO TO [illegible]
C
C    **THIS PORTION OF THE PROGRAM DELETES AN ENTIRE PARENT OR**
C    **DELETE [illegible] WITHIN A PARENT**
C
 [illegible] PRINT [illegible]
 [illegible] FORMAT(1X,"[illegible] PARENT ID(END TO EXIT)-- ")
      READ(5,[illegible]) [illegible]
      CALL OPEN[illegible](LFIT,[illegible])
 [illegible] FORMAT(A10)
      IF([illegible].EQ.[illegible]) GO TO [illegible]
      PRINT [illegible]
 [illegible] FORMAT(1X,"[illegible] TOTAL NO. OF UNIT,S TO BE DELETED-",
     [illegible] PARENT ")
      READ(5,*) [illegible]
      ARRAY(2)=[illegible]
      ARRAY(3)=[illegible]
      CALL [illegible](LFIT,ARRAY,ARRAY(1))
      IF(ARRAY(3).EQ.99993) GO TO 840
      IF([illegible].EQ.[illegible]) GO TO [illegible]
```

Figure D-1. PARENT program code (continued).

```
      [illegible] I=1,L[illegible]
      [illegible] UNIT [illegible] ",I," --"
      [illegible] (6,[illegible]) [illegible]
 [illegible] CONTINUE
      GO TO [illegible]
 [illegible] [illegible]ARRAY(2)
 1111 FORMAT(1X,A1[illegible]," [illegible] FIL [illegible]")
      CALL [illegible]
      [illegible]
 [illegible] IF([illegible](1).[illegible]) [illegible]
      K=0
      [illegible] J=1,L[illegible]
      IF([illegible](J).EQ.[illegible]) [illegible]
      [illegible] I=[illegible]
      IF([illegible]Y(I).EQ.[illegible]) GO TO [illegible]
      IF([illegible]Y(I).[illegible](J)) GO TO [illegible]
 [illegible] CONTINUE
      K=K+1
C
C     **[illegible] KEEP TRACK OF THE UNITS NOT FOUND FOR THE PARENT**
C
      [illegible](K)=[illegible]
      GO TO 880
 [illegible] [illegible]Y(I)=[illegible]
 [illegible] CONTINUE
      CALL [illegible]ARRAY(1))
      IF(K.[illegible]) GO TO [illegible]
      [illegible]
 [illegible] FORMAT(1X,"THE FOLLOWING UNITS WERE NOT FOUND FOR PARENT ",
     [illegible],1X)
      [illegible] I=1,K
      [illegible]
 [illegible] CONTINUE
      CALL [illegible]
      GO TO [illegible]
 [illegible] CALL [illegible]Y(1))
      CALL CLOSE[illegible]
      GO TO [illegible]
C
C     **THIS PORTION OF THE PROGRAM LISTS ALL PARENTS PRESENTLY ON FILE**
C
 1000 CALL [illegible]
 1100 CONTINUE
      CALL [illegible]ARRAY(1))
      [illegible]
      IF([illegible].1[illegible])GO TO 1[illegible]
      IF([illegible].ARRAY(1)) [illegible]
      GO TO 1100
 1[illegible] CALL [illegible]
      [illegible]
 1[illegible] [illegible]ARRAY(2)
      [illegible] 12[illegible] I=[illegible]
      IF([illegible]Y(I).[illegible]) [illegible]
      [illegible]Y(I)
 1[illegible] CONTINUE
      [illegible] 12[illegible] I=[illegible]
      [illegible]Y(I)=[illegible]
```

Figure D-1. PARENT program code (continued).

```
1200 CONTINUE
     GO TO 1100
 900 PRINT 9000
9000 FORMAT(1X,"ANY MORE PARENT STRUCTURES TO BE UPDATED? ",/)
     READ(6,9001) NY
9001 FORMAT(A1)
     IF(NY.EQ.1HY) GO TO 19
     PRINT 901
 901 FORMAT(1X," ALL DONE JOB HAS ENDED ")
     STOP
     END
```

Figure D-1. PARENT program code (concluded).

APPENDIX E

FORCE PROGRAM LISTING

APPENDIX E

FORCE PROGRAM LISTING

A list of the variables used in the FORCE program is given in table E-1. A listing of the FORTRAN source code of the FORCE program is contained in figure E-1.

Table E-1. List of variables for FORCE program.

Variable	Description
AA	Keeps force type
AFOR	Work storage area (Parent File)
AH	Used to check for correct force
AHOLD	Keeps force type
ARRAY	Work storage area (Force File)
ARRAY (1)	Parent Unit (key)
ARRAY (2)	Unit (key)
ARRAY (3)	Force type (number)
ARRAY (4)	Sector
ARRAY (5)	Critical Incident
ARRAY (6)	FPS @ 100%
ARRAY (7)	Combat value
ASCENE	Force to be deleted
ASRC	Work storage area (SRC File)
ATOT	Force specified
AUID	Work storage area (Unit File)
CV	Combat value specified
FPS	Firepower score
ID	Weapon number (1-80)
IDO	Weapon listed
IFIT	FIT Array (Parent File)
JFIT	FIT Array (Unit File)
KFIT	FIT Array (SRC File)
LFIT	FIT Array (Force File)
NUMFOR	Number of forces added
TYPE	Force type specified

```
      PROGRAM MTTLTY(INPUT,OUTPUT,CLDATA,TAPE10=INPUT,TAPE3=CLDATA)
      COMMON/ONE/IFIT(35),JFIT(35),KFIT(35),LFIT(35)
      DIMENSION AFOR(20),AUIR(24),ASRC(46),ARRAY(90),ATOT(25)
      DIMENSION ANYTHA(10),KEY(41),FFS(80,2)
      CALL FILEIS(LFIT,3LLFM,6LTAPE9,2LKA,ARRAY,2LPM,1LR,
     .3LFWT,2LYES)
      CALL FILEIS(IFIT,3LLFM,6LTAPE6,2LKA,AFOR,2LPM,1LR,
     .3LFWT,2LYES)
      CALL FILEIS(JFIT,3LLFM,6LTAPE7,2LKA,AUIR,2LPM,1LR,
     .3LFWT,2LYES)
      CALL FILEIS(KFIT,3LLFM,6LTAPE8,2LKA,ASRC,2LPM,1LR,
     .3LFWT,2LYES)
      CALL OPENMS(3,KEY,41,0)
      CALL READMS(3,FFS,160,34)
      CALL CLOSMS(3)
    1 PRINT 100
  100 FORMAT(1X,"IDENTIFY TYPE FORCE--")
      READ(10,9002) TYPE
      ARRAY(3)=AA=JJ=1
      IF(TYPE.EQ.1HR) ARRAY(3)=AA=JJ=2
      IF(TYPE.EQ.1HB.OR.TYPE.EQ.1HR) GO TO 3
      PRINT*,"INVALID--TRY AGAIN "
      GO TO 1
    3 AHOLD=ARRAY(3)
    5 ARRAY(3)=AHOLD
      DO 10 I=2,20
   10 AFOR(I)= 0
      DO 11 I=2,24
   11 AUIR(I) = 0
      DO 12 I=3,46
   12 ASRC(I) = 0
      DO 13 I=4,90
   13 ARRAY(I)=0
      DO 14 I=1,25
   14 ATOT(I) = 0
      GO TO 25
  111 FORMAT(1X,"FOLLOWING ACTIONS CAN BE EXECUTED",/,
     .1X,"A=ADD NEW FORCES",/,
     .1X,"C=CHANGE A UNIT'S EFFECTIVENESS",/,
     .1X,"R=ROLL TO A FORCE(PARENT UNIT) ",/,
     .1X,"S=START(REVIEW) PARENTS",/,
     .1X,"L=LIST ALL PARENTS",/,
     .1X,"X= END THE PROGRAM")
   25 PRINT 110
  110 FORMAT(1X,"ENTER ACTION TYPE( Y FOR LIST)-- ")
      ICHG=0
      READ(10,120) IKK
  120 FORMAT(A1)
      IF(IKK.EQ.1HY) PRINT 111
      IF(IKK.EQ.1HX) GO TO 25
      IF(IKK.EQ.1HC)GOTO205
      IF(IKK.EQ.1HR) GO TO 500
      IF(IKK.EQ.1HS) GO TO 4000
      IF(IKK.EQ.1HL) GO TO 300
      IF(IKK.EQ.1HX) GO TO 9990
      IF(IKK.EQ.1HA) GO TO 200
      PRINT*,"ACTION TYPE ERROR "
```

Figure E-1. FORCE program code (continued next page).

```
      GO TO 25
 201  FORMAT(A10)
 205  ICHG=1
 250  NUMFOR=1
      PRINT*," ENTER MAJOR FORCE(0 IF DONE)="
 260  I=NUMFOR
      READ(LT,201)  ATOT(I)
      IF(ATOT(I).EQ."0") GO TO 9
      AFOR(1)=AUID(1)=ASPC(1)="SPC"
      CALL OPENM(LFIT,3LI-O)
      CALL OPENM(JFIT,3LI-O)
      DO 6000  I=1,NUMFOR
      AFOR(2) = ATOT(I)
      AFOR (3) = 99999.
      CALL GET(LFIT,AFOR,AFOR(1))
      IF(AFOR(3).EQ.99999.)  GO TO 6000
      CALL OPENM(JFIT,3LI-O)
      DO 4000  J=3,20
      IF(AFOR(J).EQ.0)   GO TO 3999
      AUID(2) =   AFOR(J)
      AUID(3) =   99999.
      CALL GET(JFIT,AUID,AUID(1))
      IF(AUID(3) .EQ. 99999.)  GO TO  3000
      ARRAY(6)=0
      CALL OPENM(KFIT,3LI-O)
      DO 3000  K = 3, 24
      IF(AUID(K).EQ.0)  GO TO 2999
      ASPC(2) =  AUID(K)
      ASPC(3) = 99999.
      CALL GET(KFIT,ASPC,ASPC(1))
      IF(ASPC(3).EQ.99999.) GO TO 2999
      DO 2000  L= 3,45,2
      IF(ASPC(L).EQ.0)  GO TO 1999
      ID = ASPC(L) + 10
      ARRAY(ID) = ARRAY(ID)+ASPC(L+1)
      ARRAY(6)=ARRAY(6)+ASPC(L+1)*EFS(ID-10,AA)
 1999 CONTINUE
 2000 CONTINUE
 2999 CONTINUE
 3000 CONTINUE
      CALL CLOSEM(KFIT)
      ARRAY(1)=ATOT(I)
      ARRAY(2)=AUID(2)
      PRINT 5456,ARRAY(2)
 5456 FORMAT(1X,"ENTER RELATIVE EFFECTIVENESS OF ",1A10)
 3009 READ*,CV
      IF(CV.GT..01.AND.CV.LE.100.)GOTO3100
      PRINT*,"INVALID VALUE--TRY AGAIN "
      GO TO 3009
 3100 DO 3150 II=1,95
      IF(ARRAY(II+10).EQ.0.)GOTO3150
      ARRAY(II+10)=ARRAY(II+10)*CV/100.
 3150 CONTINUE
      ARRAY(5)=CV
      IF(ICHG.EQ.1)GOTO228
      CALL PUT(LFIT,ARRAY,500,ARRAY(1))
      GO IFETCH(LFIT,3LI?C)
```

Figure E-4. FORCE program code (continued).

```
      IF(M.LE.4400) GO TO [illegible]
      [illegible]
 [illegible] CALL [illegible](LFIT,ARRAY,[illegible],ARRAY(11))
 [illegible] DO 9031 IJ=11,[illegible]
9031 ARRAY(IJ)=[illegible]
 [illegible] CONTINUE
 [illegible] CONTINUE
      CALL CLOSEM(JFIT)
 [illegible] CONTINUE
      CALL CLOSEM(IFIT)
      CALL CLOSEM(LFIT)
      IF(ATOT(I).NE.[illegible]) PRINT*,"NEXT="
      IF(ATOT(I).NE.[illegible]) GO TO [illegible]
 [illegible] FORMAT(1A1)
      GO TO [illegible]
 [illegible] PRINT [illegible],ARRAY([illegible]),ARRAY(1),ARRAY(2)
 [illegible] FORMAT(1X, 3(2X,A10),1X,"ALREADY ON FILE")
      GO TO 4000
 [illegible] PRINT 611
 611 FORMAT(1X,"DELETE [illegible] FORCE [illegible] TO EXIT)-- ")
      READ(13,201) ASC[illegible]
      CALL [illegible](LFIT,[illegible])
      IF(ASC [illegible].EQ.3H[illegible]) GO TO 613
 [illegible] CALL GETN(LFIT,ARRAY,ARRAY(1))
      [illegible](LFIT,[illegible])
      IF([illegible]) GO TO [illegible]
      IF(ARRAY(7).NE.AA) GO TO 610
      IF(ARRAY(1).NE.ASC NE) GO TO 611
      CALL DLTE(LFIT,ARRAY(1),[illegible])
      GO TO 612
 612 PRINT*,"ALL DONE WITH THIS DELETE"
 613 CALL CLOSEM(LFIT)
      GO TO [illegible]
 [illegible] PRINT 6001,  ATOT(I)
6001 FORMAT(1X,"[illegible] I.D.",A10," NOT PREVIOUSLY IDENTIFIED",1X,/,1X,
     1"YOU WILL HAVE TO UPDATE THE FORCE FILE IF YOU WANT HIM INCLUDED",
     2/," IN THE BATTLE.")
      GO TO [illegible]
 [illegible] PRINT 7001, AEC(J), ATOT(I)
7001 FORMAT(1X,"UNIT [illegible]",A10," HAS NOT BEEN PREVIOUSLY IDENTIFIED",/,
     11X,"AS BEING INCLUDED IN FORCE ID ",A10,/,
     21X,"THE FORCE IS BEING CREATED BUT THIS UNIT WAS LEFT OUT")
      GO TO [illegible]
 [illegible] PRINT 8001 , AUT(K), AEC(J), ATOT(I)
8001 FORMAT(1X," [illegible] ",A10,1X,"NOT ON [illegible] FILE",/,
     11X,"[illegible] I.D./QTY,[illegible] ASSOCIATED WITH THIS SEC",/,
     21X,"WILL NOT BE INCLUDED IN UNIT ID ",A10,1X,"SEC ID ",A10)
      GO TO [illegible]
 [illegible] PRINT*,"[illegible] LISTED = "
      READ(13,201) ASC NE
      CALL [illegible](LFIT,[illegible])
      ARRAY(1)=ATOT[illegible]
      PRINT 9025,TYP
 [illegible] FORMAT(1X,"TYPE [illegible] = ",A10)
      [illegible]
      CALL [illegible](LFIT)
 [illegible] CONTINUE
```

Figure E-1. FORCE program code (continued).

```
      CALL G TN(LFIT,  AY,ARRAY(11))
      M=IFETCH(LFIT,?LFP)
      IF(M.EQ.1000)  GO TO 9040
      IF(AA.LT.ARRAY(3)) GO TO 9020
      IF(ASP  U .  ,ARRAY(11))     GO TO 9020
      IF(AH.EQ.ARRAY(1)) GO TO 9500
      PRINT 9026,ARRAY(1)
9026  FORMAT(1X,"FORCE ID= ",A10)
9500  PRINT 9027,ARRAY(2)
9027  FORMAT(5X,"UNIT ID:  ",A10,":  LAST CI:  ",A10,":  SECTOR: ",
     1A3,0/26X,"MEN"/24X,"IT          QTY")
      DO 9030  I=11,91
      IF(ARRAY(I). 0.0)   GO TO 9130
      IDD= I-10
      PRINT 9029,IDD,ARRAY(I)
9029  FORMAT(30X,I3,5X,F4.0)
9030  CONTINUE
      AH=ARRAY(1)
      PRINT 9036,ARRAY(2),ARRAY(4)
      GO TO 9020
9040  CONTINUE
      CALL CLOSE (LFIT)
      PRINT*,"DO YOU WANT TO LIST ANOTHER FORCE?  "
      READ(10,9100) IJK
      IF(IJK.EQ.1HY) GO TO 9300
9050  PRINT 9051
9051  FORMAT(1X," ACTION COMPLETE  ",/,
     11X," DO YOU WANT TO END THE PROGRAM")
      READ(10,9000) IJJ
      IF(IJJ.EQ.1HY) GO TO 9300
      GO TO 1
         CALL OPEN (LFIT,?LT-0)
      PRINT 9025,TYPE
      AH=EFFEFF=STEFF=0.
      CALL REWND(LFIT)
250   CONTINUE
      CALL GTN(LFIT,ARRAY,ARRAY(11))
      M=IFETCH(LFIT,?LFP)
      IF(M.EQ.1000) GO TO 254
      IF(AA.NE.ARRAY(3)) GO TO 252
      IF(AH.EQ.ARRAY(1)) GO TO 260
      IF(AH.EQ.0.)GOTO250
      RATEFF=999.
      IF(STEFF.GT..1)RATEFF=EFFEFF/STEFF*100.
      PRINT 275,AH,RATEFF
275   FORMAT(" AVERAGE EFFECTIVENESS OF ",A10," = ",F4.0/)
      EFFEFF=STEFF=0.
      PRINT 9026,ARRAY(1)
      PRINT 9027,ARRAY(2),ARRAY(3),ARRAY(4)
      OFFEF=0.
      DO 267 I=11,91
      IF(ARRAY(I).EQ.0) GO TO 267
      IDD=I-10
      OFFEF=OFFEF+ARRAY(I)*EFF(IDD,JJ)
      EFFEFF=EFFEFF+ARRAY(I)*EFF(IDD,JJ)
      PRINT 9029,IDD,ARRAY(I)
      CONTINUE
```

Figure E-1. FORCE program code (continued).

```
      AH=ARRAY(1)
      UEFF=999.
      IF(ARRAY(6).GE..1)UEFF=UEFF/ARRAY(6)*100.
      PRINT 9036,ARRAY(2),UEFF
9036  FORMAT(" EFFECTIVENESS OF ",A10," = ",F4.0)
      [illegible]=[illegible]+ARRAY(6)
      GO TO 252
 251  PAREFF=999.
      IF(PTEFF.GT..1)PAREFF=[illegible]FF/PTEFF*100.
      PRINT [illegible],AH,PAREFF
      CALL CLOSE[illegible]
      GO TO 1
9998  PRINT 9999
9999  FORMAT(1X,"ALL DONE***JOB HAS ENDED")
9999  STOP
      END
```

Figure E-1. FORCE program code (concluded).

APPENDIX F

OVLYO PROGRAM CODES AND LISTS OF VARIABLES

APPENDIX F

OVLYO PROGRAM CODES AND LISTS OF VARIABLES

This appendix contains the FORTRAN source codes of all the programs, subroutines, and subfunctions of OVLYO. Table F-1 is a list of all common variables used in the Jiffy Game. Table F-2 is a list of the program variables used in SUPER, the Jiffy Game main program. The SUPER source code is given in figure F-1. The initialization subroutine, INIT, source code is presented in figure F-2. Since all the variables used in INIT are common variables, they are defined in table F-1. Table F-3 contains the list of INDEX5, the subfunction used to convert a five subscript variable to a single subscript variable, program variables; and the FORTRAN source code for INDEX5 is given in figure F-3. Table F-4 contains a listing of the program variables used in the LOSS subroutine, which reduces the forces' weapon system arrays by whatever losses have been incurred in a particular type of combat (i.e., indirect fire, armor, etc.). Figure F-4 presents the LOSS program source code. The FORTRAN source code and list of program variables for the DISPLAY subroutine are contained in figure F-5 and table F-5, respectively. The DISPLAY subroutine interactively outputs the quantities and types of weapon systems contained in gamer specified units to the game console during processing.

Table F-1. Jiffy Game common variables. (Continued next page).

Variable	Description
ACI	Critical incident identifier
AH	HISTORY file record array
ALOSS	Weapon loss array
APOS	Attacker tactical deployment factor
ARRAY	FORCE file record array
ASCENE	Critical incident mneumonic
ASECT	Sector number
ATIME	Length of critical incident (HR)
BRRAY	SRC file record array
CFPR	Maneuver firepower ratio
CKILL	Crew kills
CREWS	Number of crewmen killed per weapon system
D	Number of weapons subject to loss apportionment
DPOS	Defender tactical deployment factor
ELMT	Array of weapon systems in sector
FPR	Total firepower ratio
FPS	Array of weapon system firepower scores
FSFPR	Fire support firepower ratio
FSSF	Fire support suppression factor
IA	Index for attacker force

Table F-1. Jiffy Game common variables (concluded).

Variable	Description
ID	Index for defender force
IENGAG	Index for tactical situation
IFIRST	Rate of advance calculation flag
IPIT	File information table for SRC file
IHIST	File information table for HISTORY file
IMOUNT	Index for attacker mobility
IP	Index for tactical situation table
IRUN	Index for type of run
ITERRN	Index for type of terrain
IVIS	Index for visibility
IYBUF	HISTORY file I/O buffer
KEY	Data file random access key
LFIT	File information table for FORCE file
MINES	Minefield flag
MYBUF	FORCE file I/O buffer
NYBUF	SRC file I/O buffer
PACK	Word packing variables
PLT	Infantryman materiel loss rates
PSN	Tactical deployment factor
SF	Suppression factor
SHOTS	Round expenditure array

Table F-2. Program variables for SUPER.

Variable	Description
AKEEP	Temporary storage variable
I	Subscript of firer weapon system
IFLAG	Logic flag
INX	Input response variable
IWP	Index for weapon system
IXNAX	Batch run constant
J	Index for force color
JRUN	Batch run constant
K	Subscript of target weapon system
KIND	Force color
M	File status integer
MM	File status integer
XLOS	Number of weapon systems

```
      OVERLAY(JUEL,0,0)
      PROGRAM JVLYU(INPUT,64,OUTPUT=64,ANSW=64,STATS=64,TAPE6=STATS,
     1 TAPE5=ANSWER,ULOUTAPE4,TAPE3=CDATA)
      COMMON IA,IL,IP,IFDGAG,ITERN,IVIS,IMOUNT,MINES,SFFR,FSFPR,FPR,
     1 AIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),FACK(2),
     3 ELMT(80,2),ALOSS(45,2),SHOTS(37,2),KILL(53,2)
      COMMON/DATA/FPS(60,2),CWFNL(83,2),APOS(12),DPOS(5),
     1 PSN(3,2,2),PL(15),KEY(-1)
      COMMON/ONE/LFIT(35),ARRAY(40),MYBUF(1024),C(80,2),ACI,
     1 ACLFM,ASECT
      COMMON/TWO/IFIT(33),FFFAY(46),NYBUF(1024)
      COMMON/THREE/IHIST(35),AH(91),IYBUF(1024)
      CALL FILEIS(LFIT,3LLFN,6LTAPEE5,2LKA,ARRAY,2LPM,1LR,
     13LEFS,1024,3LFWR,MYBUF)
      CALL FILEIS(IFIT,3LLFN,6LTAPE4,2LKA,ARRAY,2LPM,1LR,
     13LEFS,1024,3LFWR,NYBUF)
      CALL FILEIS(IHIST,3LLFN,6LTAPE3,2LKA,AH,2LPM,1LR,
     13LEFS,1024,3LFWR,IYBUF)
      PRINT 2
    2 FORMAT("1")
    3 FORMAT(" INCORRECT--MUST BE Y OR N--TRY AGAIN   ")
   10 FORMAT(1410)
   11 FORMAT(" ",1410)
      CALL INIT
      IYLAY="N"
      IRUN=3
      PRINT*,"                          C A C D A   J I F F Y   W A R   G A M E"
      GO TO 192
  190 PRINT*," SPECIFY PURPOSE OF THIS RUN"
      READ*,IRUN
      IF(IRUN.EQ.1)WRITE(3,1)IYLAY
      IF(IRUN.EQ.1)WRITE(5,*)IRUN
      IF(IRUN.GE.1.AND.IRUN.LE.3)GO TO 111
      IF(IRUN.EQ."I")GO TO 194
      PRINT11
  194 PRINT*," "
      PRINT*," ENTER 1 = TO CREATE INPUT FILE OF ANSWERS FOR BATCH JOB"
      PRINT*," ENTER 2 = TO GET OUTPUT OF RESULTS INTERACTIVELY"
      GO TO 190
  191 CALL OVERLAY(5HFORCE,10,0,6RECALL)
      GOTO111
  192 PRINT*,"DO YOU WISH TO SEE INSTRUCTIONS? (YES/NO)"
      READ1,INX
      IF(INX.EQ."N")GOTO190
      PRINT*,"ALL USER RESPONSES WILL BE OF TWO GENERAL TYPES:"
      PRINT*,"     1. YES/NO RESPONSES"
      PRINT*,"        A. Y FOR YES"
      PRINT*,"        B. N FOR NO"
      PRINT*,"     2. DATA ENTRY RESPONSES"
      PRINT*,"        A. VALID RESPONSES DISPLAYED FOR SELECTION BY USER"
      PRINT*,"        B. FRACTIONAL RESPONSES (BETWEEN 0 AND 1)"
      PRINT*,"        C. NUMERIC RESPONSES WITHIN SPECIFIED LIMITS"
      PRINT*
      PRINT*,"TO REDUCE INPUT/OUTPUT RESPONSE TIMES, VALID DATA ENTRY RE
     KSPONSES"
      PRINT*,"(2.A. ABOVE) ARE NOT DISPLAYED UNLESS REQUESTED"
```

Figure F-1. SUPER program code. (continued next page)

```
      PRINT*
      PRINT*,"TO REQUEST ADDITIONAL INFO, IF EXISTING, THE USER MUST ENT
     KEY A ""T"" (WITH DOUBLE-QUOTES)"
      GOTO190
    1 FORMAT(1A1)
    5 FORMAT(" ",1A1)
  100 CALL OVERLAY(4HLOFA,1,0,RECALL)
      GOTO111
  101 PRINT*,"ENTER 1 TO LOAD FORCES INTO A SECTOR"
      PRINT*,"      2 TO CALCULATE RATE-OF-ADVANCE"
      PRINT*,"      3 TO ASSESS COMBAT"
      PRINT*,"      4 TO APPORTION CBT LOSSES TO UNITS"
      PRINT*,"      5 TO DISPLAY BATTLE STATISTICS"
      PRINT*,"      6 TO DISPLAY WEAPON ARRAYS"
      PRINT*,"      7 TO ADD SFC'S TO THE FILE"
      PRINT*,"      8 TO RESTART AT A PREVIOUSLY GAMED CI"
      PRINT*,"      9 TO END GAME AND/OR UPDATE HISTORY FILE"
  111 PRINT*,"????????????? D E C I S I O N   P O I N T ?????????????"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(6,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."T")GOTO131
      IF(INX.GE.1.AND.INX.LE.9)GO TO 102
      PRINT11
   11 FORMAT(" INCORRECT RESPONSE - TRY AGAIN")
      GO TO 101
  102 GOTO(1,1,100,150,115,240,500,310,400,900),INX
  150 IF(IFIRST.EQ.1)GOTO0163
      PRINT*,"RATE-OF-ADVANCE MUST BE CALCULATED BEFORE ASSESSMENTS ARE
     RENDERED"
      GOTO101
  163 CALL OVERLAY(6HSUPRTS,8,0,RECALL)
      GOTO0112
  115 DO 240 J=1,2
      DO 240 I=43,58
      IF(ELMT(I,J).GT.0.)GOTO245
  240 CONTINUE
      GOTO0105
  245 CALL OVERLAY(6HCANNON,6,0,RECALL)
  105 IF(MINES.NE.1)GOTO200
      CALL OVERLAY(4HMINE,4,0,RECALL)
  200 DO 205 J=1,2
      DO 205 I=11,30
      IF(ELMT(I,J).GT.0.)GOTO210
  205 CONTINUE
      GOTO215
  210 CALL OVERLAY(6HATGM,7,0,RECALL)
  215 IF(ELMT(3,1).LE.0..AND.ELMT(3,2).LE.0.)GOTO0265
      CALL OVERLAY(6HINFCMB,3,0,RECALL)
  265 DO 270 J=1,2
      DO 270 I=59,62
      IF(ELMT(I,J).GT.0.)GOTO275
  270 CONTINUE
      GOTO0111
  275 CALL OVERLAY(4HRADIO,5,0,RECALL)
      GOTO0111
  112 PRINT*,"DO YOU WISH TO INCLUDE ANY LOSSES DUE TO TACAIR FOR APPORT
```

Figure F-1. SUPER program code (continued).

```
  112 PRINT12"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO113
      IF(INX.EQ."N")GOTO115
      PRINT3
      GOTO112
  113 KIND="BLUE"
      J=1
   20 PRINT21,KIND
   21 FORMAT(" ANY ",A4," LOSSES?     ")
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO25
      IF(INX.EQ."N")GOTO30
      PRINT3
      GOTO20
   25 PRINT*,"ENTER WPN ID,# LOST (0,0 WHEN DONE)--    "
   26 READ*,IWP,XLOS
      IF(IRUN.EQ.1)WRITE(5,*)IWP,XLOS
      IF(IRUN.EQ.3)PRINT*,IWP,XLOS
      IF(IWP.EQ.0)GOTO30
      IF(XLOS.GT.ELMT(IWP,J))GOTO27
      IF(XLOS.LT.0.)GOTO31
      ELMT(IWP,J)=ELMT(IWP,J)-XLOS
      XLOSS(80,IWP)=XLOSS(80,IWP)+IFIX(XLOS*10.)*PACK(J)
   24 PRINT*,"NEXT--      "
      GOTO26
   27 PRINT28,ELMT(IWP,J),IWP,KIND
   28 FORMAT(" OVERKILL!!!     ONLY ",F4.0," OF WEAPON ",I3," REMAIN IN T
     1HE ",A4," FORCE FOR THIS SECTOR.")
   29 PRINT*,"LAST ENTRY IGNORED"
      GOTO24
   31 PRINT*,"# LOST MUST BE ENTERED AS A POSITIVE NUMBER."
      GOTO29
   30 IF(J.EQ.2)GOTO115
      J=2
      KIND="RED"
      GOTO20
  200 IF(IRUN.NE.1)CALL OVERLAY(6HRESULT,9,0,RECALL)
      GOTO111
  500 PRINT*,"       FORCE STRUCTURES"
      PRINT*,"ELEMENT      BLUE      RED"
      DO 510 I=1,80
      IF(ELMT(I,1).EQ.0..AND.ELMT(I,2).EQ.0.)GOTO510
      PRINT5,I,ELMT(I,1),ELMT(I,2)
    5 FORMAT("    ",I2,"        ",F6.0,"    ",F6.0)
  510 CONTINUE
      GOTO111
C
  300 CALL OVERLAY(5HBUILD,12,0,RECALL)
      GOTO111
  400 PRINT*,"ENTER CI PREVIOUSLY GAMED -"
      READ10,ACI
      IF(IRUN.EQ.1)WRITE(5,10)ACI
```

Figure F-1 SUPER program code (continued).

Best Available Copy

```
      IF(I.OR.EQ.3)R=INT1R,ACI
      IF(ACI.EQ."T")GOTO430
C
      CALL OPENM(IHIST,3LI-O,1LR)
      AH(1)=ACI
      AH(4)=90909.
      CALL GET(IHIST,AH,AH(1),0,10)
      IF(AH(4).EQ.90909.)GOTO450
C
      CALL OPENM(LFIT,3LI-O,1LR)
  415 ARRAY(1)=0.
      CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,2LFP)
      IF(M.EQ.100B)GOTO435
      CALL DLTE(LFIT)
      GOTO415
C
  435 AKEEP=AH(5)
      AH(5)=AH(1)
      AH(1)=AH(2)
      AH(2)=AH(3)
      AH(3)=AKEEP
      DO 405 I=1,9
  405 ARRAY(I)=AH(I)
C
      CALL PUT(LFIT,ARRAY,90,ARRAY(1))
  410 CALL GETN(IHIST,AH,AH(1))
      M=IFETCH(IHIST,2LFP)
      IF(M.EQ.100B)GOTO440
      IF(AH(1).NE.ACI)GOTO440
      GOTO435
C
  440 CALL CLOSEM(LFIT)
      CALL CLOSEM(IHIST)
      PRINT445,ACI
  445 FORMAT(" FORCE FILE HAS BEEN RESTARTED AT CI ",A10)
      ACI=0.
      GOTO111
C
  450 CALL CLOSEM(IHIST)
      PRINT455,ACI
  455 FORMAT(" CI ",A10," IS NOT ON HISTORY FILE!")
  460 PRINT*,"CI'S ON HISTORY FILE -"
      AKEEP=90909.
      CALL OPENM(IHIST,3LI-O,1LR)
  470 CALL GETN(IHIST,AH,AH(1))
      M=IFETCH(IHIST,2LFP)
      IF(M.EQ.100B)GOTO480
      IF(AH(1).EQ.AKEEP.OR.AH(1).EQ."CI LOSSES".OR.AH(1).EQ."CI AMMO")GO
     .TO470
      AKEEP=AH(1)
      PRINT475,AKEEP
  475 FORMAT(" ",22X,A10)
      GOTO470
C
  480 CALL CLOSEM(IHIST)
      ACI=0.
```

Figure F-1. SUPER program code (continued).

```
C
  9?3 CALL OVERLAY(HAPFON,11,0,RECALL)
      GOTO111
C
  90? PRINT910,ACI
  910 FORMAT(" HAS THE LAST SECTOR BEEN GAMED FOR CI ",1A10,"?")
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT4,INX
      IF(INX.EQ."Y")GOTO915
      IF(INX.EQ."N")GOTO9?5.
      PRINT3
      GOTO9??
  915 IFLAG=0
      CALL OPENM(IHIST,3LI-C,1LF)
      AH(1)="CI LOSSES"
      AH(2)=ACI
      AH(3)=1.
      AH(4)=99999.
      CALL GET(IHIST,AH,AH(1))
      IF(AH(4).EQ.99999.)GOTO512
      IFLAG=1
      GOTO5??
  55? CALL GETN(IHIST,AH,AH(1))
      M=IFETCH(IHIST,2LFP)
      IF(M.EQ.100?)GOTO514
  55? I=AH(3)
      DO 511 K=1,6)
      ALOSS(I,K)=AH(K+1?)
  511 CONTINUE
      IF(I.EQ.30)GOTO514
      GOTO5?0
  512 PRINT513,ACI
  513 FORMAT(" COMBAT LOSSES FOR CI ",A10," HAVE NOT BEEN APPORTIONED")
  514 CALL CLOSEM(IHIST)
      IF(IFLAG.EQ.0)GOTO53?
      CALL OPENM(IHIST,3LI-C,1LF)
      AH(1)="CI AMMO"
      AH(2)=ACI
      AH(3)=1.
      AH(4)=99999.
      CALL GET(IHIST,AH,AH(1))
      IF(AH(4).EQ.99999.)GOTO520
      IFLAG=IFLAG+1
  527 DO 5?6 I=1,35
  5?6 SHOTS(I,1)=AH(I+10)
      DO 522 I=1,35
  522 SHOTS(I,2)=AH(I+45)
      GOTO533
  520 PRINT524,ACI
  524 FORMAT(" AMMO STATISTICS FOR CI ",A10," HAVE NOT BEEN CALCULATED")
  530 CALL CLOSEM(IHIST)
      IF(IFLAG.NE.2)GOTO521
      WRITE(6,525)ACI
  525 FORMAT(" ",33X,"CUMMULATIVE STATISTICS FOR CI ",1A10)
```

Figure F-1. SUPER program code (continued).

Best Available Copy

```
      CALL OVERLAY(4HMEFULT,9,0,RECALL)
      GOTO 935
525   PRINT 527,ACI
527   FORMAT(" STATISTICS FOR CI ",1A10," ARE NOT ON FILE PROPERLY")
535   PRINT*,"SHOULD FO OR FILE BE ADDED TO HISTORY FILE?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(6,1)INX
      IF(IRUN.EQ.3)PRINT5,INX
      IF(IRUN.EQ.1.AND.(INX.EQ."Y".OR.INX.EQ."N"))GOTO950
      IF(INX.EQ."Y")GOTO920
      IF(INX.EQ."N")GOTO950
      PRINT5
      GOTO415

920   CALL OPENM(IHIST,3LI-O,1LR)
      CALL OPENM(LFIT,3LI-O,1LR)
925   CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=FETCH(LFIT,2LFP)
      IF(M.EQ.100)GOTO940
      IF(ARRAY(2).NE.ACI)ARRAY(2)=0
      ARRAY(2)=ACI
      DO 930 I=1,90
930   AH(I)=ARRAY(I)
      AKEEP=AH(5)
      AH(5)=AH(3)
      AH(3)=AH(2)
      AH(2)=AH(1)
      AH(1)=AKEEP
      CALL PUT(IHIST,AH,900,AH(1))
      GOTO925

940   CALL CLOSEM(LFIT)
      CALL CLOSEM(IHIST)

950   IF(IRUN.EQ.1)PRINT*," ""ANSWER"" FILE HAS BEEN CREATED."

660   STOP 1
      END
```

Figure F-1. SUPER program code (concluded).

```
      SUBROUTINE INIT
      COMMON IA,IB,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,DFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PACK(2),
     3 ELMT(80,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PSN(6,2,2),PLT(15),KEY(41)
      CALL OPENMS(3,KEY,41,0)
      CALL READMS(3,FPS,160,34)
      CALL CLOSMS(3)
      DATA((CREWS(I,J),I=1,53),J=1,2)/2.,2*0.,5*3.,2.,3.,0.,0.,3*2.,3*0.
     1,4.,0.,4.,3*0.,2.,5.,5.,3*0.,4.,5.,7.,5.,5.,9.,14.,10.,13.,14.,6*0
     1.,5*2.,2*4.,2.,2*0.,
     2 5*3.,2.,3.,3.,0.,3*2.,3.,3.,0.,4.,6.,7.,5.,0.,0.,1.,2.,8.,3*0.,
     3 5.,5.,7.,0.,0.,7.,10.,8.,0.,9.,10.,6.,5.,3*0.,5*2.,4.,0./
      DATA (APOS(I),I=1,12)/1.,1.5,2.,.8,1.,1.2,1.,1.2,1.4,1.,1.4,1.6/
      DATA (DPOS(I),I=1,6)/1.,1.,.5,2.,1.5,1.2/
      DATA(((PSN(I,J,K),J=1,2),K=1,2),I=1,6)/2*.33,2*.67,.33,1.,.67,.5,
     1 2*.67,1.,.33,3*(.67,.67,1.,1.)/
C
C          MATERIEL LOSSES PER INFANTRY MAN LOST.
      DATA(PLT(K),K=1,15)/.017,.0,1.,2*.0,1.,.067,.05,.02,.0,.05,.0,.0,
     1 .05,.02/
C
      DO 9 I=1,50
      DO 9 J=1,2
    9 SHOTS(I,J)=0.
      PACK(1)=100000.
      PACK(2)=1.
      RETURN
      END
```

Figure F-2. INIT program code.

Table F-3. Program variables for INDEX5.

Variable	Description
INDEX5	Equivalent single subscript
I1	First parameter subscript
I2	Second parameter subscript
I3	Third parameter subscript
I4	Fourth parameter subscript
I5	Fifth parameter subscript
L1	Length of first parameter array
L2	Length of second parameter array
L3	Length of third parameter array
L4	Length of fourth parameter array

NOTE: All COMMON variables are defined in table F-1.

```
      FUNCTION INDEX5(I1,I2,I3,I4,I5,L1,L2,L3,L4)
C
C     THIS FUNCTION RETURNS THE 1 DIMENSIONAL ELEMENT NUMBER OF AN ARRAY
C     SIMULATING ONE OF 5 DIMENSIONS
C     IT IS ASSUMED THE DATA IS STORED BY COLUMNS AND YOU ARE SEEKING
C     ELEMENT (I1,I2,I3,I4,I5) OF ARRAY (L1,L2,L3,L4,N)
C
      INDEX5=(1+(I1-1+L1*(I2-1+L2*(I3-1+L3*(I4-1+L4*(I5-1))))))
      RETURN
      END
```

Figure F-3. INDEX5 program code.

Table F-4. Program variables for LOSS.

Variable	Description
AKILL	The number of weapons type I kill by all firers.
I	Firing weapon system index
INX	Gamer response variable
ISTART	Variable indexing beginning subscript of firers
ISTOP	Variable indexing ending subscript of firers
J	Force identifier
K	Index for weapon systems lost
KSTART	Variable indexing beginning subscript of weapon systems lost
KSTOP	Variable indexing ending subscript of weapon systems lost

NOTE: All COMMON variables are defined in table F-1.

```
      SUBROUTINE LOSS(ISTART,ISTOP,KSTART,KSTOP)
      COMMON IA,IL,IP,ILNGAG,ITERN,IVIS,IMOUNT,MINES,DFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 D(2),FSF(2),PACK(2),
     3 ELMT(40,2),ALOSS(40,40),SHOTS(35,2),CKILL(43,2)
      COMMON/DATA/FPS(90,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PEN(6,2,2),PLT(15),KEY(41)
    5 PRINT*,"DO YOU WISH TO SUBTRACT LOSSES FROM FORCE STRUCTURES?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT6,INX
      IF(INX.EQ."Y")GOTO10
      IF(INX.EQ."N")GOTO60
      PRINT2
    1 FORMAT(1A1)
    6 FORMAT(" ",1A1)
    2 FORMAT(" INCORRECT - RESPONSE MUST BE Y OR N - TRY AGAIN")
      GOTO 5
   10 DO 50 J=1,2
      DO 50 I=ISTART,ISTOP
      DO 50 K=KSTART,KSTOP
      IF(J.EQ.2)GOTO20
      AKILL=ALOSS(I,K)/PACK(1)/10.
      GOTO25
   20 AKILL=(ALOSS(I,K)-IFIX(ALOSS(I,K)/PACK(1))*PACK(1))/10.
   25 IF(AKILL.LE.0.)GOTO50
      ELMT(K,J)=ELMT(K,J)-AKILL
      IF(ELMT(K,J).LT.0.)ELMT(K,J)=0.
   50 CONTINUE
      GOTO100
   60 DO 90 I=ISTART,ISTOP
      DO 90 K=KSTART,KSTOP
   90 ALOSS(I,K)=0.
  100 RETURN
      END
```

Figure F-4. LOSS program code.

Table F-5. Program variables for DISPLAY.

Variable	Description
AR	Number of weapon systems
CIL	Factor for combat intensity level
I	Unit record word index
ICODE	Weapon system item code
IFLAG	Print flag
INC	Increment counter
INX	Gamer response variable
J	Force identifer
M	File status variable
PARENT	Name of parent unit
REMAIN	Number of particular weapon systems remaining in unit
TFPS	Total firepower score
UEFF	Unit effectiveness
UNIT	Name of unit
XLOST	Number of particular weapon system losses

NOTE: All COMMON variables are defined in table F-1.

```
      SUBROUTINE DISPLAY
      DIMENSION CIL(5),ICODE(10),REMAIN(10),XLOST(10)
      COMMON IA,ID,IP,IENGAG,ITERN,IVIS,IMOUNT,MINES,DFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PACK(2),
     3 FLMT(80,2),ALOSS(80,60),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(N ,2),CREWS(53,2),AFOS(12),DPOS(5),
     1 FS(r,2,2),PLT(1 ),KEY(41)
      COMMON/DMF/LFIT(35),ARRAY(50),MYBUF(1024),C(80,2),ACI,
     $ASTEMF,ASPCT
      DATA(CIL(I),I=1,5)/1000.,5.,4.,2.5,1./
   10 FORMAT(1A10)
   15 FORMAT(" ",1A10)
C
      PRINT*,"ENTER PARENT OF UNIT(S) TO BE DISPLAYED -"
      READ10,PARENT
      IF(IRUN.EQ.1)WRITE(5,10)PARENT
      IF(IRUN.EQ.3)PRINT18,PARENT
C
      TFPS=0.
      UEFF=0.
      CALL OPENM(LFIT,3L1-O,1LF)
      ARRAY(1)=PARENT
      ARRAY(7)=99909.
  100 CALL GET(LFIT,ARRAY,ARRAY(1),3,10)
      IF(ARRAY(7).NE.99909.)GOTO110
      PRINT305,PARENT
      GOTO9900
C
  110 PRINT*,"ENTER UNIT ID (OR ALL) -"
      READ10,UNIT
      IF(IRUN.EQ.1)WRITE(5,10)UNIT
      IF(IRUN.EQ.3)PRINT15,UNIT
C
      IFLAG=0
  113 IF(UNIT.EQ."ALL")GOTO120
      IF(UNIT.EQ.ARRAY(2))GOTO120
  112 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,2LFP)
      IF(M.EQ.100)GOTO200
      IF(ARRAY(1).NE.PARENT)GOTO200
      GOTO113
C
  120 J=ARRAY(3)
      TFPS=TFPS+ARRAY(4)
C
      DO 150 I=1,4
      AR=ARRAY(I+10)-IFIX(ARRAY(I+10)/100000.)*100000.
      IF(AR.LE.0.)GOTO150
      UEFF=UEFF+AR*EFF(I,J)
  150 CONTINUE
      GOTO112
C
  200 CALL CLOSEM(LFIT)
      IF(TFPS.LE.0.)GOTO300
      UEFF=UEFF/TFPS*100.
      PRINT205,PARENT,UEFF
```

Figure F-5. DISPLAY program code.(continued next page).

```
200 FORMAT(" ",//,1X,A10,2X,"EFF=",F4.0,/,1X,67("-"))
    CALL OPENM(LFIT,3LI-O,1LR)
    ARRAY(1)=PARENT
    ARRAY(7)=99999.
    CALL GET(LFIT,ARRAY,ARRAY(1),0,10)
    IF(ARRAY(7).NE.99999.)GOTO155
    PRINT323,PARENT
    GOTO3..

155 IFLAG=0
160 IF(UNIT.EQ."ALL")GOTO200
    IF(UNIT.EQ.ARRAY(2))GOTO200
170 CALL GETN(LFIT,ARRAY,ARRAY(1))
    M=IFETCH(LFIT,2LFP)
    IF(M.NE.100)GOTO500
    IF(ARRAY(1).NE.PARENT)GOTO500
    GOTO160

200 J=ARRAY(3)
    UEFF=0.
    DO 210 I=1,80
    AR=ARRAY(I+10)-IFIX(ARRAY(I+10)/100000.)*100000.
    IF(AR.LE.0.)GOTO210
    UEFF=UEFF+AR*EPS(I,J)
210 CONTINUE
    IF(ARRAY(6).GT.0.)GOTO211
    UEFF=-99.
    GOTO212
211 UEFF=UEFF/ARRAY(6)*100.
212 PRINT215,ARRAY(2),UEFF
215 FORMAT("0",A10,2X,"EFF=",F4.0,/,1X,67("."))

    INC=0
    DO 220 I=1,80
    IF(ARRAY(I+10).EQ.0.)GOTO225
    INC=INC+1
    ICODE(INC)=I
    XLOST(INC)=IFIX(ARRAY(I+10)/100000.)/10.
    REMAIN(INC)=ARRAY(I+10)-XLOST(INC)*100000.
    IF(INC.LT.9)GOTO225

230 PRINT235,(ICODE(INX),INX=1,INC)
235 FORMAT(" ITEM CODE",10(3X,I2,1X))
    PRINT240,(REMAIN(INX),INX=1,INC)
240 FORMAT(" # REMAIN ",10F6.1)
    PRINT245,(XLOST(INX),INX=1,INC)
245 FORMAT(" # LOST     ",10(1X,F5.1))
    PRINT250
250 FORMAT(" ",67("."))
    INC=0
    IFLAG=1
225 IF(INC.NE.0.AND.I.EQ.80)GOTO230
220 CONTINUE

    GOTO170
500 CALL CLOSEM(LFIT)
    GOTO...
```

Figure F-5. DISPLAY program code (continued).

```
  300 PRINT305,UNIT
  305 FORMAT(" UNIT ",A1,," WAS NOT FOUND")
C
  600 RETURN
      END
```

Figure F-5. DISPLAY program code (concluded).

APPENDIX G

OVLY 1 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX G

OVLY 1 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN source code and variable list for OVLY 1 (ROFA). This program sets a number of parameters used throughout the combat assessment routines and calculates the attacker's rate of advance, firepower scores for both forces, and attacker:defender firepower ratios. Table G-1 lists the ROFA program variables; figure G-1 is the FORTRAN source code listing.

Table G-1. Program variables for OVLY1 (ROFA).

Variable	Description
ADIST	Attacker's covered distance
CFPS	Ground combat firepower score
F	Fraction of sector Red force massed
FPRM	Maneuver firepower ratio
FSFPS	Fire support firepower score
I	Weapon system index
IEL	Force index
IFPR	Rate-of-advance firepower ratio index
INX	Input response variable
ISTART	Do-loop index
ITABLE	Engagement type index
J	Force index
JVIS	Rate-of-advance visibility index
K	Weapon system index
KIND	Force color
RATE	Rate-of-advance data array
RMIN	Minimum attacker firepower ratio
ROA	Rate-of-advance (KM/HR)
ROA1 ROA2	Intermediate ROA calculation variable
STALE	Rate of advance index determiner
TFPS	Total firepower score

NOTE: All COMMON variables are defined in table F-1.

```
OVL [illegible](OFA,1, )
PROGRAM OVLY1
COMMON IA,IB,IP,ILFG,G,ITER,IVIC,ICOUNT,MINES,DFPR,FSFPR,FPR,
1 ATIM,IFIRST,IFUN,
2 TF(2),FSTF(2),PACK(2),
3 ELAT(8,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
 COMMON/DATA/FFS(80,2),CPFMS(33,2),APDS(12),CPCS(5),
1 PSN(5,2,2),FLT(11),KEY(-1)
 DIMENSION STALF(11),RATE(1440),DFPS(2),FSFPS(2),TFPS(2)
 DATA (STALF(I),I=1,11)/.8,1.,1.5,2.,2.5,3.,3.5,4.,5.,6.,8./
 DATA (RATE(I),I=1,288)/
1.,.,.,.5,.5,1.3,1.4,1.5,1.7,1.9,2.1,2.2,2.3,0.,0.,.9,1.7,2.0,2.2,2
2.6,2.9,3.1,3.4,3.9,4.6,0.,0.,.3,.5,.8,.9,1.0,1.2,1.3,1.4,1.4,1.6,0
3.,.,.,.4,.8,1.5,1.4,1.6,1.7,1.8,2.0,2.1,2.6,0.,0.,.2,.3,.5,.7,.8,1.
42,1.3,1.4,1.5,1.6,0.,0.,.4,.5,.9,1.0,1.2,1.4,1.6,1.7,1.8,2.0,0.,0.
5,.4,.7,1.0,1.2,1.3,1.4,1.6,1.7,1.8,2.0,0.,0.,.5,1.0,1.6,1.8,2.1,2.
63,2.6,2.7,3.3,3.9,0.,0.,.2,.3,.5,.7,.8,.8,.9,1.0,1.0,1.2,0.,0.,.3,
7.,.,1.0,1.2,1.3,1.4,1.6,1.7,1.8,2.3,0.,0.,.1,.1,.3,.4,.5,.7,.8,.9,.
89,1.0,.,0.,.1,.3,.7,.8,.9,1.0,1.2,1.3,1.4,1.7,0.,0.,.3,.5,.8,.9,1
9.,.,1.2,1.3,1.4,1.6,1.7,0.,0.,.4,.8,1.2,1.2,1.3,1.6,2.0,2.1,2.6,3.3
A,0.,0.,.1,.2,.4,.4,.5,.7,.8,.9,.8,.9,0.,0.,.2,.4,.8,.9,1.0,1.2,1.3
B,1.4,1.5,2.1,0.,0.,.1,.2,.3,.3,.4,.5,.7,.7,.7,.8,0.,0.,.1,.3,.5,.7
C,.,.,.8,1.0,1.2,1.3,1.6,0.,0.,.1,.2,.4,.5,.7,.8,1.0,1.2,1.2,1.3,0.,
D.,.,.2,.4,.6,.9,1.0,1.0,1.2,1.3,1.3,1.4,0.,0.,.1,.2,.3,.4,.5,.5,.6,
E.7,.7,.8,0.,0.,.1,.2,.4,.5,.7,.7,.8,.8,.8,.9,0.,0.,.1,.2,.3,.3,.4,
F.4,.5,.6,.7,.8,0.,0.,.1,.2,.4,.5,.5,.6,.7,.7,.8,.9/
 DATA (RATE(I),I=289,576)/
1.,.,.3,.5,1.0,1.5,1.6,1.8,1.9,2.1,2.4,2.5,2.5,0.,.5,.9,1.7,1.9,2.2,
22.4,2.2,3.0,3.6,4.4,5.1,0.,.2,.3,.6,.9,1.0,1.2,1.3,1.5,1.6,1.6,1.8
3,0.,.3,.5,.9,1.5,1.6,1.8,1.9,2.1,2.2,2.4,2.9,0.,.1,.2,.3,.6,.7,.9,
41.,.1.2,1.3,1.5,1.6,.,.1,.2,.4,.9,1.0,1.2,1.3,1.5,1.6,1.8,2.1,0.,
5.3,.5,.9,1.2,1.3,1.5,1.5,1.8,1.9,2.1,2.4,0.,.3,.5,1.2,1.8,2.1,2.4,
62.6,2.8,3.1,3.7,4.4,0.,.1,.2,.3,.6,.7,.8,.9,1.0,1.2,1.2,1.3,0.,.2,
7.3,.6,1.2,1.3,1.5,1.6,1.8,1.9,2.1,2.6,0.,.1,.1,.1,.3,.4,.6,.8,.9,1
8.0,1.0,1.2,0.,.1,.2,.5,.7,.9,1.0,1.2,1.3,1.5,1.6,1.9,0.,.2,.3,.6,.
99,1.0,1.2,1.3,1.5,1.5,1.7,1.8,0.,.3,.5,.8,1.3,1.5,1.6,1.8,2.2,2.4,
A2.5,2.7,0.,.1,.2,.3,.4,.4,.5,.7,.9,.9,.9,1.0,0.,.1,.2,.4,.9,1.0,1.
32,1.3,1.5,1.6,1.8,2.4,0.,0.,.1,.1,.3,.3,.4,.6,.7,.7,.7,.9,0.,0.,.1
C.,.3,.6,.7,.9,1.0,1.2,1.3,1.5,1.6,0.,0.,.1,.4,.6,.7,.9,1.2,1.3,1
D.3,1.4,0.,0.,0.,.1,.5,1.0,1.2,1.2,1.3,1.5,1.5,1.5,0.,0.,0.,0.,.3,.
F4,.5,.6,.7,.7,.7,.8,0.,0.,0.,0.,.4,.5,.7,.7,.9,.9,.9,1.0,0.,0.,0.,
F5.,.1,.3,.4,.4,.5,.6,.6,.7,0.,0.,0.,0.,.3,.4,.5,.6,.7,.7,.7,.8/
 DATA (RATE(I),I=577,864)/
10.,0.,0.,0.,.2,.2,.3,.3,.3,.4,.4,.5,0.,0.,0.,0.,.4,.4,.5,.5,.6,.6,
2.7,.8,0.,0.,0.,0.,.1,.2,.2,.2,.2,.2,.2,.3,0.,0.,0.,0.,.2,.2,.3,.3,
3.3,.4,.4,.5,0.,0.,0.,0.,.1,.1,.1,.2,.2,.2,.3,.3,0.,0.,0.,0.,.1,.2,
4.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,.2,.2,.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,
5.3,.3,.4,.4,.5,.5,.6,.7,0.,0.,0.,0.,.1,.1,.1,.1,.2,.2,.2,.3,0.,0.,
60.,0.,.2,.2,.2,.2,.3,.3,.4,.4,0.,0.,0.,0.,.1,.1,.1,.1,.1,.2,.2,.3,
70.,0.,0.,0.,.1,.1,.2,.2,.2,.2,.3,.4,0.,0.,0.,0.,.1,.2,.2,.2,.3,.3,
8.4,.4,0.,0.,0.,0.,.2,.2,.3,.3,.4,.4,.5,.6,0.,0.,0.,0.,.1,.1,.1,.1,
9.2,.2,.2,.3,0.,0.,0.,0.,.1,.2,.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,.1,.1,
A.1,.1,.1,.1,.1,.2,0.,0.,0.,0.,.1,.1,.1,.2,.2,.2,.2,.3,0.,0.,0.,0.,
30.,.1,.1,.1,.1,.2,.2,.2,0.,0.,0.,0.,.1,.2,.2,.2,.2,.2,.2,0.,0.,
C0.,0.,0.,0.,.1,.1,.1,.1,.1,.1,0.,0.,0.,0.,0.,0.,.1,.1,.1,.1,.2,.2,
D0.,0.,0.,0.,0.,.1,.1,.1,.1,.1,.1,0.,0.,0.,0.,0.,0.,.1,.1,.1,.1,
E.1,.1/
```

Figure G-1. ÔVLY1 (ROFA) program code.

```
 OVL LAY(ROFA,1, )
 PROGRAM OVLY1
 COMMON IN,ID,IP,ILRG,G,ITERR,IVIS,IHOUNT,FINES,SFPR,FSFPR,FPR,
1 ATIM,IFIRST,IRUN,
2 TF(2),FSTF(2),PACK(2),
3 ELMT( ,2),ALGS(80,80),SHOTS(33,2),CKILL(53,2)
 COMMON/DATA/FFS(80,2),CPFMS(33,2),APOS(12),LPOS(5),
1 PSN( ,2,2),FLT(11),KEY(-1)
 DIMENSION STALF(11),RATE(1440),STFS(2),FSFFS(2),TFPS(2)
 DATA (STALF(I),I=1,11)/.8,1.,1.5,2.,2.5,3.,3.5,4.,5.,6.,8./
 DATA (RATE(I),I=1,288)/
1.,.,.,.5,.9,1.3,1.4,1.6,1.7,1.9,2.1,2.2,2.3,0.,0.,.,.9,1.7,2.0,2.2,2
2.4,2.5,3.1,3.4,3.9,4.6,0.,0.,.3,.5,.8,.9,1.0,1.2,1.3,1.4,1.4,1.6,0
3.,.,.,.4,.8,1.3,1.4,1.6,1.7,1.6,2.0,2.1,2.6,0.,0.,.2,.3,.5,.7,.8,1.
42,1.3,1.4,1.6,1.8,0.,0.,.4,.5,.9,1.0,1.2,1.4,1.6,1.7,1.8,2.0,0.,0.
5.,.4,.8,1.0,1.2,1.3,1.4,1.6,1.7,1.8,2.0,0.,0.,.5,1.0,1.6,1.8,2.1,2.
63,2.6,2.7,3.3,3.9,0.,0.,.2,.3,.5,.7,.8,.8,.9,1.0,1.0,1.2,0.,0.,.3,
7.,.,1.0,1.2,1.3,1.4,1.6,1.7,1.8,2.3,0.,0.,.1,.1,.3,.4,.5,.7,.8,.9,.
89,1.0,.,0.,.1,.3,.7,.8,.9,1.0,1.2,1.3,1.4,1.7,0.,0.,.3,.5,.8,.9,1
9.,.,1.2,1.3,1.4,1.6,1.7,0.,0.,.4,.8,1.2,1.2,1.3,1.6,2.0,2.1,2.6,3.3
A,0.,0.,.1,.2,.4,.4,.5,.7,.5,.8,.8,.9,0.,0.,.2,.4,.8,.9,1.0,1.2,1.3
B,1.4,1.5,2.1,0.,0.,.1,.2,.3,.3,.4,.5,.7,.7,.7,.8,0.,0.,.1,.3,.5,.7
C,.,.,.9,1.0,1.2,1.3,1.6,0.,0.,.1,.2,.4,.5,.7,.8,1.0,1.2,1.2,1.3,0.,
D.,.,.2,.4,.8,.9,1.0,1.0,1.2,1.3,1.3,1.4,0.,0.,.1,.2,.3,.4,.5,.5,.6,
E.7,.7,.8,0.,0.,.1,.2,.4,.5,.7,.7,.8,.8,.8,.9,0.,0.,.1,.2,.3,.3,.4,
F.4,.5,.6,.7,.8,0.,0.,.1,.2,.4,.5,.5,.6,.7,.7,.8,.9/
 DATA (RATE(I),I=289,576)/
1.,.,.3,.5,1.0,1.5,1.6,1.8,1.9,2.1,2.4,2.5,2.5,0.,.5,.9,1.7,1.9,2.2,
22.4,2.2,3.2,3.6,4.4,5.1,0.,.2,.3,.6,.9,1.0,1.2,1.3,1.5,1.6,1.6,1.8
3,0.,.3,.5,.9,1.5,1.6,1.8,1.9,2.1,2.2,2.4,2.9,0.,.1,.2,.3,.6,.7,.9,
41.0,1.2,1.3,1.5,1.6,0.,.1,.2,.4,.9,1.0,1.2,1.3,1.5,1.6,1.8,2.1,0.,
5.3,.5,.8,1.2,1.3,1.5,1.6,1.8,1.9,2.1,2.4,0.,.3,.5,1.2,1.8,2.1,2.4,
62.6,2.4,3.1,3.7,4.4,0.,.1,.2,.3,.5,.7,.9,.9,1.0,1.2,1.2,1.3,0.,.2,
7.3,.5,1.2,1.3,1.5,1.6,1.8,1.9,2.1,2.5,0.,.1,.1,.1,.3,.4,.6,.8,.9,1
8.0,1.0,1.2,0.,.1,.2,.5,.7,.9,1.0,1.2,1.3,1.5,1.6,1.9,0.,.2,.3,.6,.
99,1.0,1.2,1.3,1.5,1.5,1.7,1.9,0.,.3,.5,.9,1.3,1.5,1.6,1.8,2.2,2.4,
A2.5,3.7,0.,.1,.2,.3,.4,.4,.5,.7,.9,.9,.9,1.0,0.,.1,.2,.4,.9,1.0,1.
32,1.3,1.5,1.6,1.8,2.4,0.,0.,.1,.1,.3,.3,.4,.6,.7,.7,.7,.9,0.,0.,.1
C.3,.6,.7,.9,1.0,1.2,1.3,1.5,1.8,0.,0.,0.,.1,.4,.6,.7,.9,1.2,1.3,1
D.3,1.4,0.,0.,0.,.1,.5,1.0,1.2,1.2,1.3,1.5,1.5,1.5,0.,0.,0.,0.,.3,.
E4,.5,.6,.7,.7,.7,.8,0.,0.,0.,0.,.4,.5,.7,.7,.9,.9,.9,1.0,0.,0.,0.,
F0.,.1,.3,.4,.4,.5,.5,.5,.7,0.,0.,0.,0.,.3,.4,.5,.6,.7,.7,.7,.8/
 DATA (RATE(I),I=577,864)/
10.,0.,0.,0.,.2,.2,.3,.3,.3,.4,.4,.5,0.,0.,0.,0.,.4,.4,.5,.5,.6,.6,
2.7,.8,0.,0.,0.,0.,.1,.2,.2,.2,.2,.2,.2,.3,0.,0.,0.,0.,.2,.2,.3,.3,
3.3,.4,.4,.5,0.,0.,0.,0.,.1,.1,.1,.2,.2,.2,.3,.3,0.,0.,0.,0.,.1,.2,
4.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,.2,.2,.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,
5.3,.3,.4,.4,.5,.5,.6,.7,0.,0.,0.,0.,.1,.1,.1,.1,.2,.2,.2,.3,0.,0.,
60.,0.,.2,.2,.2,.2,.3,.3,.4,.4,0.,0.,0.,0.,.1,.1,.1,.1,.1,.2,.2,.3,
70.,0.,0.,0.,.1,.1,.2,.2,.2,.2,.3,.4,0.,0.,0.,0.,.1,.2,.2,.2,.3,.3,
8.4,.4,.5,0.,0.,0.,.2,.2,.3,.3,.4,.4,.5,.5,0.,0.,0.,0.,.1,.1,.1,.1,
9.2,.2,.2,.3,0.,0.,0.,0.,.1,.2,.2,.2,.3,.3,.3,.4,0.,0.,0.,0.,.1,.1,
A.1,.1,.1,.1,.1,.2,0.,0.,0.,0.,.1,.1,.1,.2,.2,.2,.2,.3,0.,0.,0.,0.,
30.,.1,.1,.1,.1,.2,.2,.2,0.,0.,0.,0.,0.,.1,.2,.2,.2,.2,.2,.2,0.,0.,
C0.,0.,0.,0.,.1,.1,.1,.1,.1,.1,0.,0.,0.,0.,0.,0.,.1,.1,.1,.1,.2,.2,
D0.,0.,0.,0.,0.,.1,.1,.1,.1,.1,.1,0.,0.,0.,0.,0.,0.,.1,.1,.1,.1,
E.1,.1/
```

Figure G-1. ÓVLY1 (ROFA) program code.

```
  DATA (RATE(I),I=865,1152)/
1.0,.0,.0,.0,.0,.0,.0,.7,.4,.8,.9,1.0,1.0,.0,.0,.0,.9,.9,1.0,1.2,1.3,1
2.4,1.5,1.7,2.0,.0,.0,.0,.0,.3,.3,.4,.5,.5,.5,.6,.6,.6,.7,0.,0.,0.,.4,.6,
3.,.0,.7,.8,.4,.5,1.0,1.2,0.,0.,0.,.2,.0,.3,.3,.4,.5,.5,.6,.6,0.,0.,0
4.,.2,.3,.4,.0,.0,.0,.0,.7,.8,.0,.0,0.,.4,.5,.6,.6,.6,.7,.8,.8,.9,0
.,.0,.0,.0,.6,.7,.8,.9,1.0,1.2,1.4,1.5,1.7,0.,0.,0.,.2,.2,.3,.3,.3,.4
5.,.5,.6,.0,.0,.3,.5,.6,.6,.7,.8,.9,1.0,0.,0.,0.,.1,.1,.2,.
7.,.5,.5,.4,.4,.4,.5,.6,.0,.2,.3,.3,.5,.5,.6,.7,.8,0.,0.,0.,.3,.
8.,.0,.5,.5,.6,.7,.7,0.,0.,0.,.4,.5,.5,.6,.7,.8,.9,1.2,1.5,0.,0.
9,0.,.1,.2,.2,.2,.3,.3,.3,.3,.4,.0,.0,0.,.2,.3,.4,.5,.5,.6,.6,.7,.9
A,.5,.0,.0,.1,.1,.1,.2,.2,.3,.3,.3,.3,0.,0.,0.,.1,.2,.3,.3,.4,.5,.5
0,.6,.7,.0,.0,.1,.2,.2,.3,.3,.5,.5,.5,.5,0.,0.,0.,.2,.3,.4,.5,.5
C,.0,.0,.0,.0,0.,0.,0.,.1,.2,.2,.2,.3,.3,.3,.3,0.,0.,0.,0.,.2,.2
3,.2,.2,.3,.3,.3,.3,0.,0.,0.,.1,.1,.2,.2,.2,.2,.3,.3,0.,0.,0.,0.
E,.1,.2,.2,.2,.3,.2,.3,.3/
  DATA (RATE(I),I=1153,1440)/
1.0,.0,.7,.8,1.0,1.1,1.2,1.3,1.4,1.6,1.7,1.8,0.,0.,1.2,1.3,1.5,1.7,
22.,.2,.2,.2,.2,.5,.5,.5,.5,.5,.5,.4,.5,.6,.7,.8,.9,1.0,1.1,1.1,1.2,0.
3,0.,.6,.7,1.0,1.1,1.2,1.3,1.4,1.5,1.6,2.0,0.,0.,.2,.3,.4,.5,.6,.7,
4.8,.9,1.0,1.0,.0,.0,.3,.4,.5,.7,.8,.9,1.0,1.1,1.2,1.4,0.,0.,.6,.7,
5.8,.9,1.0,1.1,1.2,1.3,1.4,1.6,0.,0.,.8,1.0,1.2,1.4,1.6,1.8,2.0,2.1
6,2.5,3.0,0.,0.,.2,.3,.4,.5,.5,.5,.7,.8,.8,.9,0.,0.,.4,.5,.8,.9,1.0
7.1.1,1.2,1.3,1.4,1.8,0.,0.,.1,.2,.2,.3,.4,.5,.6,.7,.7,.8,0.,0.,.2,
8.3,.5,.6,.7,.8,.9,1.0,1.1,1.3,0.,0.,.4,.5,.6,.7,.8,.9,1.0,1.1,1.2,
91.0,0.,0.,.6,.7,.9,.9,1.0,1.2,1.5,1.6,2.0,2.5,0.,0.,.1,.2,.3,.3,.4
4,.0,.0,.5,.6,.7,0.,0.,.3,.4,.6,.7,.8,.9,1.0,1.1,1.2,1.6,0.,0.,.1,.
"1,.2,.2,.3,.4,.5,.5,.5,.5,0.,.2,.2,.4,.5,.6,.7,.8,.9,1.0,1.2,0.
5,0.,.1,.2,.3,.4,.5,.6,.8,.9,.9,.9,0.,0.,.2,.4,.5,.7,.8,.8,.9,1.0,1
0.0,1.0,0.,0.,0.,.1,.2,.3,.4,.4,.5,.5,.5,.5,0.,0.,0.,.2,.3,.4,.5,.5
E.,.0,.5,.5,.0,0.,0.,.1,.1,.2,.3,.3,.4,.4,.5,.5,0.,0.,0.,.1,.2,.3
F,.4,.4,.5,.5,.5,.5/
  DO 300 I=1,80
  DO 300 K=1,80
300 DEMSS(I,K)=0.
    DO 310 J=1,2
    DO 310 I=1,30
310 SHOTS(I,J)=0.
  5 PRINT*,"IS BLUE ATTACKING LEFT?"
    READ1,INX
    IF(IRUN.EQ.1)WRITE(5,1)INX
    IF(IRUN.EQ.3)PRINT8,INX
  1 FORMAT(1A1)
  8 FORMAT(" ",1A1)
    IF(INX.EQ."Y")GO TO 10
    IF(INX.EQ."N")GO TO 11
    PRINT2
    GO TO 5
  2 FORMAT(" INCORRECT - RESPONSE MUST BE Y OR N - TRY AGAIN")
 10 IA=1
    IB=2
    GO TO 20
 11 IA=2
    IB=1
 20 PRINT*,"ENTER TYPE OF ENGAGEMENT"
    READ*,IENGAG
    IF(IRUN.EQ.1)WRITE(5,*)IENGAG
    IF(IRUN.EQ.3)PRINT*,IENGAG
```

Figure G-1. OVLY1 (ROFA) program code (continued).

```
      IF(IENGAG.EQ."T")GOTO21
      IF(IENGAG.GE.1.AND.IENGAG.LE.6)GOTO25
      PRINT3
   21 PRINT*,"       FOR MEETING ENGAGEMENT..........ENTER 1"
      PRINT*,"           DELAY.......................ENTER 2"
      PRINT*,"           WITHDRAW....................ENTER 3"
      PRINT*,"           DEFEND FORTIFIED POSITION..ENTER 4"
      PRINT*,"           DEFEND PREPARED POSITION...ENTER 5"
      PRINT*,"           DEFEND HASTY POSITION......ENTER 6"
    3 FORMAT(" INCORRECT ENTRY - TRY AGAIN")
      GO TO 20
   25 IF(IENGAG.GT.3)GO TO 30
      IP=IENGAG
      GO TO 35
   30 PRINT*,"ENTER ATTACKER POSTURE"
      READ*,INX
      IF(IRUN.EQ.1)WRITE(3,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."T")GOTO33
      IF(INX.GE.1.AND.INX.LE.3)GOTO31
      PRINT3
   33 PRINT*,"       FOR FRONTAL ATTACK.....ENTER 1"
      PRINT*,"           SINGLE ENVELOPMENT.ENTER 2"
      PRINT*,"           DOUBLE ENVELOPMENT.ENTER 3"
      GO TO 30
   31 IP=3*IENGAG-3+INX
   35 DO 40 IEL=1,2
      CFPS(IEL)=0.
      DO 40 I=1,30
      CFPS(IEL)=CFPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)
   40 CONTINUE
      ISTART=31
      IEL=1
      KIND="RED"
   42 PRINT45,KIND
   45 FORMAT(" IS THERE A SIGNIFICANT ",A4," AIR THREAT?")
      READ1,INX
      IF(IRUN.EQ.1)WRITE(3,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GO TO 44
      IF(INX.EQ."N")GO TO 43
      PRINT2
      GO TO 42
   43 ISTART=43
   44 FSFPS(IEL)=0.
      DO 50 I=ISTART,80
      FSFPS(IEL)=FSFPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)
   50 CONTINUE
      IF(IEL.EQ.2)GO TO 55
      IEL=2
      ISTART=31
      KIND="BLUE"
      GO TO 42
   55 CFPS(IA)=CFPS(IA)*APOS(IP)
      CFPS(ID)=CFPS(ID)*CPOS(IENGAG)
      IF(CFPS(ID)+FSFPS(ID).GE.1)GOTO54
      PRINT*,"***THERE ARE NO DEFENDERS***"
```

Figure G-1. OVLY1 (ROFA) program code (continued).

```
      PRINT*," FORCE STRUCTURE MUST BE CHANGED"
      IF(IRUN.EQ.1)PRINT*," PROGRAM STOPPED----ATTACH FORCE FILE BEFORE
     1RESTARTING"
      IF(IRUN.EQ.1)STOP
      GOTO39
   54 IFIRST=1
      IF(CFPS(ID).GE.1)GOTO57
      PRINT*,"THERE IS NO MANEUVER FP RATIO"
      CFFR=0.
      GOTO58
   57 CFFR=CFPS(IA)/CFPS(ID)
   58 IF(FSFPS(ID).GE.1)GOTO59
      PRINT*,"THERE IS NO FIRE SUPPORT FP RATIO"
      FSFR=0.
      GOTO 62
   59 FSFR=FSFPS(IA)/FSFPS(ID)
   62 FFR=(CFPS(IA)+FSFPS(IA))/(CFPS(ID)+FSFPS(ID))
C     NOTE-ALL AUTOMATIC WEAPONS WERE CONSIDERED IN MANEUVER FIREPOWER.
   50 PRINT*,"ENTER TERRAIN TYPE"
      READ*,ITERRN
      IF(IRUN.EQ.1)WRITE(5,*)ITERRN
      IF(IRUN.EQ.3)PRINT*,ITERRN
      IF(ITERRN.EQ."T")GOTO051
      IF(ITERRN.GE.1.AND.ITERRN.LE.4)GOTO60
      PRINT3
   51 PRINT*,"     FOR OPEN TERRAIN........ENTER 1"
      PRINT*,"         ROLLING TERRAIN.....ENTER 2"
      PRINT*,"         HILLY TERRAIN.......ENTER 3"
      PRINT*,"         MOUNTAINOUS TERRAIN.ENTER 4"
      GO TO 50
   60 PRINT*,"ENTER VISIBILITY FACTOR"
      READ*,IVIS
      IF(IRUN.EQ.1)WRITE(5,*)IVIS
      IF(IRUN.EQ.3)PRINT*,IVIS
      IF(IVIS.EQ."T")GOTO061
      IF(IVIS.GE.1.AND.IVIS.LE.5)GOTO65
      PRINT3
   61 PRINT*,"     FOR VISIBILITY OF 100% ENTER 1"
      PRINT*,"                        85% ENTER 2"
      PRINT*,"                        65% ENTER 3"
      PRINT*,"                        45% ENTER 4"
      PRINT*,"                        30% ENTER 5"
      GO TO 60
   65 JVIS=(IVIS+1)/2
      PRINT*,"IS ATTACKER MOUNTED"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT1,INX
      IF(INX.EQ."N")IMOUNT=1
      IF(INX.EQ."Y")IMOUNT=2
      IF(INX.EQ."N".OR.INX.EQ."Y")GO TO 66
      PRINT2
      GO TO 65
   66 PRINT*,"ENTER FRACTION OF SECTOR ATTACKER MASSED (MAX=1)"
      READ*,F
      IF(IRUN.EQ.1)WRITE(5,*)F
      IF(IRUN.EQ.3)PRINT*,F
```

Figure G-1. OVLYI (ROFA) program code (continued).

```
      IF(F.GT.0.AND.F.LE.1)GOTO67
      PRINT3
      GOTO5?
   67 RMIN=1.5
      FPRM=(FPR-RMIN*(1.-F))/F
      IF(FPR.LT.RMIN)FPRM=FPR
      IF(IENGAG.LE.3)GO TO 76
      STALE(2)=1.4
      STALE(3)=1.7
   76 DO 75 I=1,11
      IF(FPRM.LT.STALE(I))GO TO 8?
   75 CONTINUE
      I=12
   8? IFPR=I
      ITABLE=IENGAG
      IF(ITABLE.EQ.3)ITABLE=2
      IF(ITABLE.GT.3)ITABLE=ITABLE-1
C     ENTER RATE ARRAY AND READ ROA.
      ROA1=RATE(INDEXS(IFPR,IMOUNT,JVIS,ITERRN,ITABLE,12,2,3,4))
      IF(IFPR.LT.12)GOTO82
      ROA2=ROA1
      GOTO83
   82 ROA2=RATE(INDEXS(IFPR+1,IMOUNT,JVIS,ITERRN,ITABLE,12,2,3,4))
   83 ROA=ROA1+(ROA2-ROA1)*(FPRM-STALE(IFPR-1))/(STALE(IFPR)-STALE(IFPR-
     1 1))
      IF(ROA1.LE.0.)ROA=0.
      IF(IENGAG.EQ.3.AND.IMOUNT.EQ.2)ROA=ROA*1.5
   85 PRINT*,"ARE MINES EMPLOYED IN THIS SECTOR"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT6,INX
      IF(INX.EQ."Y")GO TO 9?
      IF(INX.EQ."N")GO TO 9?
      PRINT2
      GO TO 85
   9? IF(IENGAG.NE.4.AND.IENGAG.NE.?)ROA=.75*ROA
      MINES=1
      GO TO 93
   9? MINES=2
   93 PRINT*,"SHOULD TIME OR DISTANCE BE HELD CONSTANT"
      READ*,INX
      IF(IRUN.EQ.1)WRITE(5,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."?")GOTO91
      IF(INX.EQ.1)GOTO100
      IF(INX.EQ.2)GOTO15?
      PRINT3
   91 PRINT*,"      FOR CONSTANT TIME.....ENTER 1"
      PRINT*,"          CONSTANT DISTANCE.ENTER 2"
      GO TO 93
  100 PRINT*,"ENTER ATTACK TIME IN HOURS (MAX 24)"
      READ*,ATIME
      IF(IRUN.EQ.1)WRITE(5,*)ATIME
      IF(IRUN.EQ.3)PRINT*,ATIME
      IF(ATIME.LE.24.AND.ATIME.GE.1)GO TO 11?
      PRINT4
      GO TO 100
```

Figure G-1. OVLY1 (ROFA) program code (continued).

```
110 ADIST=ROA*ATIME
    GO TO 200
150 PRINT*,"ENTER ATTACK DISTANCE IN METERS (MAX 75000.)"
    READ*,ADIST
    IF(IRUN.EQ.1)WRITE(5,*)ADIST
    IF(IRUN.EQ.3)PRINT*,ADIST
    IF(ADIST.GT.0..AND.ADIST.LE.75000.)GO TO 160
    PRINT3
    GO TO 150
160 IF(ROA.EQ.0.)GO TO 161
    ATIME=(ADIST/1000.)/ROA
    ADIST=ADIST/1000.
    GO TO 200
161 ATIME=0.
    ADIST=0.
200 IF(IRUN.EQ.1)GO TO 98
    PRINT205
205 FORMAT("1")
    TFPS(IA)=(FPS(IA)+FSFPS(IA)
    TFPS(ID)=CFPS(ID)+FSFPS(ID)
    PRINT*,"*-------------------RATE OF ADVANCE---------------------
   1------*"
    PRINT210
210 FORMAT(" I",63X,"I")
215 FORMAT(" I     FP RATIO IN SECTOR'S MAIN ATTACK AREA",5("."),
   C F4.1,13X,"I")
220 FORMAT(" I      TOTAL FP RATIO",25("."),F4.1,13X,"I")
225 FORMAT(" I      MANEUVER FP RATIO",25("."),F4.1,13X,"I")
230 FORMAT(" I      FIRE SUPPORT FP RATIO",21("."),F4.1,13X,"I")
235 FORMAT(" I      RATE-OF-ADVANCE (KPH)",20("."),F5.2,13X,"I")
240 FORMAT(" I      DURATION OF ATTACK (HR)",18("."),F5.1,13X,"I")
245 FORMAT(" I      DISTANCE ADVANCED (KM)",19("."),F5.1,13X,"I")
246 FORMAT(" I      MANEUVER FP SCORE",25("."),F8.0,"/",F8.0,"I")
247 FORMAT(" I      FIRE SUPPORT FP SCORE",21("."),F8.0,"/",F8.0,"I")
248 FORMAT(" I      TOTAL FP SCORE",28("."),F8.0,"/",F8.0,"I")
    PRINT246,CFPS(IA),CFPS(ID)
    PRINT247,FSFPS(IA),FSFPS(ID)
    PRINT248,TFPS(IA),TFPS(ID)
    PRINT215,FPRN
    PRINT220,FPR
    PRINT225,CFPR
    PRINT230,FSFPR
    PRINT210
    PRINT235,ROA
    PRINT240,ATIME
    PRINT245,ADIST
    PRINT210
    PRINT*,"*=======================================================
   1======*"
    PRINT203
98  IF(IRUN.EQ.1)PRINT*," BATTLE CHARACTERISTICS PRINTED HERE"
99  END
```

Figure G-1. OVLY1 (ROFA) program code (concluded).

APPENDIX H

OVLY 2 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX H

OVLY 2 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN source code and variable list for the OVLY 2 (TANK) program. TANK contains the assessment logic for the gaming of combat involving tanks, armored vehicles, and antitank weapons. The program variables are listed in table H-1, and the FORTRAN source code listing is in figure H-1.

Table H-1. Program variables for OVLY2 (TANK).

Variable	Description
A	Acquisition data array
ACQ	Acquisition discriminator
AKILL	Current losses to weapon systems
ATCREW	Number of infantrymen killed per antitank weapon
BLUE	Blue weapon system cumulative losses
CLOST	Crewmen losses
ELMTS	Total number of targets
ELS	Total number of firers
FDF	Fire distribution factor
FIRE	Expected number of completed firings
I	Firer weapon index
IFIND	SSKP data block index
IFLAG	Flag for displaying/suppressing table header
INDX	SSKP data entry index
INX	Input response variable
IPSN	Positioning units index for attacker/defender
ISUP	Suppression degradation factor index
ITYP	Ammunition type index

Table H-1. Program variables for OVLY2 (TANK) (continued).

Variable	Description
J	Firer force index
JJ	Attacker/defender firer force index
JPSN	Positioning units index for contact
K	Target weapon index
KFLAG	Initial contact flag
KIND	Force color
KK	Category type index
KT	Expected number of completed firings firer index
L	Target force index
LL	Attacker/defender target force index
M	Weapon system (firer) index
MAXR	Range index
N	Weapon system (target) index
NBR	SSKP single integer index
OPERA	Weapon system operational availability
PKILL	Target's survival probability against firer
PLOSS	Current losses to weapon systems
RED	Red weapon system cumulative losses
ROUNDS	Ammunition fired per target
SKILL	Loss apportionment factor denominator

Table H-1. Program variables for OVLY2 (TANK)(concluded).

Variable	Description
SS	Defilade SSKP/Final SSKP
SSKP	Weapon system single shot kill probability
SSS	Fully exposed SSKP
SUPDEG	Suppression degradation factor coefficient
TKILL	Targets killed
V	Visibility degradation factors
VICTIM	Firer's target
VISDEG	Visibility degradation factor
WTS	Weapon system category weights
XN	Weapon system engagements

NOTE: All COMMON variables are defined in table F-1.

```
 OVERLAY(TANK,2,0)
 PROGRAM OVLY2
 COMMON IA,JB,IF,ICHGAG,I[illegible],IVIL,IMCNT,MINES,GFPR,FSFPR,FPR,
1 ATIME,IFIRST,IRUN,
2 DF(2),FSSF(2),PACK(2),
3 FLMT(90,2),ALOSS(60,2),SHOTS(30,2),CKILL(53,2)
 COMMON/DATA/FFC(80,2),CREWS(53,2),APSE(12),DPOS(5),
1 MSN[illegible],2,2),PLT(1[illegible]),KEY(41)
 DIMENSION FSF(2),SUPUFS(3),V(5),A(6,2),FIPE(4,6,29),ATCREW(5)
 DIMENSION SSKF(32),ACD(2),GPERS(22,2),PLOSS(32,2,2),WTS(4,2),
1PKILL(29)
 DATA(ATCREW(I),I=1,5)/1.,2.,3.,2.,3./
 DATA((WTS(I,J),J=1,2),I=1,4)/.5,.6,4.3,5.7,5.9,7.4,10.,10./
 DATA (V(I),I=1,5)/1.,.85,.65,.45,.3/
 DATA((A(I,J),I=1,6),J=1,2)/.9,.9,.9,.9,.9,.9,.9,.75,.75,2*.33,.50/
 DATA(SUPUFS(I),I=1,3)/1.,2.35,3.32/
 DATA((FIRE(ITERN,1,K),K=1,29),ITERN=1,4)/.41,.31,.34,2*.41,2*.68
 [illegible]
 [illegible],2*.68,.21,.1[illegible],.17,2*.21,2*.34,.18,.34,.18,2*.26,.18,0.,.34,
 [illegible]2*.27,.17,.27,2*.26,.17,.34,3*[illegible].16,2*.34,.34,.25,.26,2*.34,
 [illegible]2*.32,.27,.[illegible],.27,2*.39,.27,[illegible].52,2*.33,.26,.33,2*.39,.26,.52,
 [illegible].27,2*.32,.28,.2,.21,2*.25,2*.42,.23,.42,.23,2*.32,.23,0.,
 [illegible],2*.25,.21,.25,2*.32,.21,.42,3*0.,.23,2*.42/
 DATA((FIRE(ITERN,2,K),K=1,29),ITERN=1,4)/0.,.37,.65,0.,.55,2*1.3
 [illegible],1.3,[illegible],1.3,2*.71,.65,.71,2*.98,.65,1.3,
 [illegible],2*1.3,[illegible],2*.78,.39,.78,.39,0.,.59,.39,
 [illegible].78,2*.49,.39,[illegible],.39,.78,3*0.,.39,2*.78,0.,.23,.79,0.,
 [illegible],2*1.58,[illegible],.85,[illegible],1.14,.85,[illegible],1.58,2*.99,.79,.99,1.19,1.1
 [illegible],.78,1.58,3*0.,[illegible],0.,.25,.74,0.,.87,2*1.48,.78,1.48,.78,
 [illegible],1.11,.78,0.,1.48,2*.93,.74,.93,2*1.11,.74,1.48,3*0.,.78,2*1.48/
 DATA((FIRE(ITERN,3,K),K=1,29),ITERN=1,4)/2*0.,.65,0.,.05,2*1.3,
 [illegible],.98,.64,[illegible],1.3,2*.71,.65,.71,.98,.64,.65,1.3,3*0.,
 [illegible],2*1.3,2*0.,.24,[illegible],2*.48,.26,[illegible],.26,0.,.36,.26,0.,.48,2*
 [illegible],.24,.3,.39,.26,.24,.48,3*0.,.26,2*.48,2*0.,.97,0.,1.07,2*1.94,
 [illegible],1.94,1.[illegible],1.46,1.03,0.,1.94,2*1.21,.97,1.21,1.46,1.03,
 [illegible],1.94,2*[illegible],1.03,2*1.94,2*0.,.76,0.,.87,2*1.52,.82,1.52,.82,0.,
 [illegible]1.14,.82,0.,1.52,2*.98,.76,[illegible],1.14,.82,.75,1.52,3*0.,.82,2*1.52/
 DATA((FIRE(ITERN,4,K),K=1,29),ITERN=1,4)/2*0.,.46,0.,.01,2*.92,
 [illegible],.48,2*0.,.46,[illegible],2*.46,.46,.58,[illegible],.46,.92,3*0.,.48,
 [illegible],.32,2*0.,.22,[illegible],.22,2*.48,2*.46,.24,2*0.,.24,0.,.46,2*.29,
 [illegible],.24,.23,.46,3*0.,[illegible],2*.46,2*0.,.58,0.,.58,2*1.16,.61,
 [illegible],2*0.,[illegible],1.16,2*.83,.58,.83,0.,.61,.58,1.16,3*0.,.61,
 [illegible],2*0.,.84,[illegible],2*1.08,.[illegible],1.08,.56,2*0.,.56,0.,1.08,2*.68,
 [illegible],.54,1.08,3*[illegible],2*1.08/
 DATA((FIRE(ITERN,5,K),K=1,29),ITERN=1,4)/2*0.,.24,0.,.01,2*.48,
 [illegible],.24,2*0.,2*.3,.24,.3,0.,.24,.24,4*0.,.24,4*0.,
 [illegible],.37,2*.24,.12,.24,.12,2*[illegible],.12,2*[illegible],2*.15,.12,.15,0.,.12,
 [illegible],.12,4*0.,.3,0.,.12,2*.[illegible].31,.[illegible].31,2*0.,.31,2*0.,2*.38,
 [illegible],.33,0.,.31,.3,4*0.,.31,4*0.,.45,0.,.14,2*.9,.47,.9,.47,2*0.,
 [illegible],2*0.,2*[illegible],0.,.47,.45,4*0.,.47,2*0./
 DATA((FIRE(ITERN,6,K),K=1,29),ITERN=1,4)/2*0.,.08,2*0.,2*.16,
 [illegible],.08,2*0.,.[illegible],2*0.,2*.1,.08,.1,0.,.05,.08,4*0.,.06,4*0.,
 [illegible],2*.[illegible],.13,2*0.,.13,2*[illegible],2*.05,.04,.05,0.,
 [illegible],.3,.1[illegible],.3,.15,2*0.,.15,2*0.,2*
 [illegible],.19,[illegible],2*.1[illegible],4*0.,[illegible],2*0.,2*.48,.24,.48,.24,
 [illegible],2*0.,[illegible],.3,.24,[illegible],2*.24,4*0.,.24,2*0./
 DATA ((GPERS(K,J),K=1,22),J=1,2)/.[illegible]1.,2*.72,.67,.78,.67,2*.81,2*1.
```

Figure H-1. OVLY2 (TANK) program code.

```
     1.,1.,.91,.9,3*1.,..,.,*1.,2*.78,1.,.52,1.,.81,3*1.,3*.81,3*1.,2*.85/
 700 PRINT*,"DO YOU WISH TO PROCESS ARMOR/ANTIARMOR ASSESSMENTS?"
     READ1,INX
     IF(IRUN.EQ.1)WRITE(6,1)INX
     IF(IRUN.EQ.3)PRINT6,INX
     IF(INX.EQ."Y")GOTO609
     IF(INX.EQ."N")GOTO230
     PRINT2
     GOTO700
 609 PRINT*,"                ARMOR/ANTIARMOR ASSESSMENTS"
     CALL OPENMS(5,KEY,-1,0)
C
 740 PRINT*,"IS THIS INITIAL COMBAT FOR THIS SECTOR?"
     READ1,INX
     IF(IRUN.EQ.1)WRITE(6,1)INX
     IF(IRUN.EQ.3)PRINT6,INX
     IF(INX.EQ."Y")GOTO750
     IF(INX.EQ."N")GOTO760
     PRINT2
     GOTO740
C
 750 KFLAG=1
     GOTO760
 765 KFLAG=2
C
 760 FOR(1)=0.
     FOR(2)=0.
     DO 770 I=1,32
     DO 770 J=1,2
     DO 770 K=1,2
 770 PLOST(I,J,K)=0.
C
C     INTERVISIBILITY ENTRY
C
     VISRNG=V(IVIS)
     ACQ(IA)=A(IENGAG,2)
     ACQ(ID)=A(IENGAG,1)
  11 PRINT*,"ENTER RANGE INDEX BETWEEN ATTACKER & DEFENDER"
     READ*,MAXR
     IF(IRUN.EQ.1)WRITE(6,*)MAXR
     IF(IRUN.EQ.3)PRINT*,MAXR
     IF(MAXR.EQ."T")GOTO40
     IF(MAXR.EQ.0)GOTO155
     INX=7-IVIS
     IF(INX.LE.3)INX=INX-1
     IF(MAXR.GE.1.AND.MAXR.LE.INX)GOTO15
     PRINT*,"VISIBILITY INSUFFICIENT FOR ENGAGEMENT AT SPECIFIED RANGE"
  40 PRINT*,"     IF RANGE IS BETWEEN:"
     GOTO(41,42,43,45,46),IVIS
  41 PRINT*,"          3000 & 2501 ENTER 6"
  42 PRINT*,"          2500 & 2001 ENTER 5"
  43 PRINT*,"          2000 & 1501 ENTER 4"
     PRINT*,"          1500 & 1001 ENTER 3"
  45 PRINT*,"          1000 &  501 ENTER 2"
  46 PRINT*,"           500 &    0 ENTER 1"
     PRINT*,"          **TO STOP** ENTER 0"
     GOTO11
```

Figure H-1. OVLY2 (TANK) program code (continued).

```
1.  FEF(1)=0.
    FEF(2)=0.

    DO 10 J=1,2
    FEF(J)=0.
    DO 10 K=11,32
    IF(ELMT(K,J)-FLOSS(K,J,1).LT.0.)GOTO10
    IF(J.EQ.1.AND.IOUNT.EQ.2.AND.K.LT.16)GOTO10
    KK=1
    IF(K.GT.16)KK=4
    IF(K.GT.19)KK=3
    [illegible]
    JJ=1
    IF(J.EQ.2)JJ=2
    IF(K.EQ.13)KK=3
    IF(K.EQ.22.AND.J.EQ.1)KK=2
    FEF(J)=FEF(J)+(ELMT(K,J)-FLOSS(K,J,1))*WTS(KK,JJ)*CPERA(K-10,J)
10  CONTINUE

C   ASSESSMENT

1   FORMAT(1A1)
    FORMAT(" ",1A1)
2   FORMAT(" INCORRECT - RESPONSE MUST BE Y OR N - TRY AGAIN")
    DO 100 J=1,2
    KIND="RED"
    IF(J.EQ.2)KIND="BLUE"
    DO 100 K=11,32
    L=1
    IF(J.EQ.1)L=2
    IF(L.EQ.1.AND.IOUNT.EQ.2.AND.K.LT.16)GOTO100
    ELMTS=ELMT(K,L)-FLOSS(K,L,1)
    IF(ELMTS.LT.1.)GOTO100
    IPSN=1
    IF(L.EQ.IO)IPSN=2
    IF(IPSN.EQ.1)JPSN=2
    IF(IPSN.EQ.2)JPSN=1
    VICTIM=ELMTS*CPERA(K-10,L)*VISLG*PSN(IENGAG,IPSV,KFLAG)*ACQ(J)
    IF(VICTIM.LT.1.)GOTO100
    IF(L.EQ.2)GOTO501
    GOTO(604,604,604,604,604,601,601,601,601,602,602,604,100,100,
   1603,603,602,100,100,100,603,100),K-10
501 GOTO(604,604,604,604,604,601,601,601,601,602,602,602,602,100,
   1603,603,603,603,603,100,603,603),K-10
601 N=1
    GOTO605
602 N=2
    GOTO605
603 N=3
    GOTO605
604 N=4
605 KK=1
    IF(K.GT.15)KK=4
    IF(K.EQ.17.OR.K.GT.19)KK=3
    IF(K.GT.24.AND.K.NE.27)KK=2
    LL=1
```

Figure H-1. OVLY2 (TANK) program code (continued).

```
      IF(L.EQ.IA)LL=2
      AKILL=1.
      SKILL=0.
      DO 102 I=11,30
      FKILL(I-10)=1.
      IF(J.EQ.IA.AND.IMOUNT.EQ.2.AND.(I.LT.15.OR.(I.EQ.15.AND.J.EQ.1)))
     1GOTO102
      ELS=FLNT(I,J)-FLOSS(I,J,1)
      IF(ELS.LT.1.)GOTO102
      IF(J.EQ.1)KT=I-10
      IF(J.EQ.2)KT=I
      IF((I.EQ.22.OR.I.EQ.27).AND.J.EQ.1)KT=3
      IF(J.EQ.1.AND.KT.GT.10)GOTO102
      IF(J.EQ.2.AND.KT.EQ.30)GOTO102
      IF(J.EQ.2)GOTO050
      IF(I.LT.16.OR.I.GT.20)GOTO030
      IF(K.GE.15.AND.K.LE.23)GOTO025
      ISUP=1
      ITYP=3
      IF(I.EQ.16.OR.I.EQ.17.OR.I.EQ.19)ITYP=4
      GOTO095
   30 ISUP=2
      IF(I.EQ.22)ISUP=1
      ITYP=6
      IF(I.EQ.12)ITYP=7
      IF(I.EQ.11)ITYP=8
      IF(I.EQ.14)ITYP=9
      IF(I.EQ.15)ITYP=10
      GOTO090
   25 ISUP=1
      ITYP=2
      IF(I.EQ.16.OR.I.EQ.17)ITYP=1
      IF(I.EQ.18.OR.I.EQ.20)ITYP=5
      GOTO090
   50 IF(I.LT.16.OR.I.GT.19)GOTO055
      IF(K.GE.16.OR.K.LE.22)GOTO060
      ISUP=1
      ITYP=2
      IF(I.EQ.19)ITYP=3
      GOTO090
   55 ISUP=2
      ITYP=5
      IF(I.EQ.12)ITYP=4
      IF(I.EQ.22)ITYP=6
      IF(I.EQ.23.OR.I.EQ.13)ITYP=3
      IF(I.EQ.28)ITYP=2
      IF(I.EQ.29)ITYP=7
      IF(I.EQ.21.AND.MAX.LE.2)ITYP=5
      IF(I.EQ.14)ITYP=9
      IF(I.EQ.11)ITYP=10
      GOTO090
   60 ISUP=1
      ITYP=3
      IF(I.EQ.16.OR.I.EQ.17)ITYP=1
   90 IF(I.LE.15)ISUP=3
      YN=ELS*OPERA(I-10,J)*FSN(IENGAG,JPSN,KFLAG)
      IF(K.EQ.22.AND.L.EQ.1)KK=2
```

Figure H-1. OVLY2 (TANK) program code (continued).

```
      ROUNDS=FIFL(ITE,PN,MAX,KT)*(1.-CF(J)*SURFAG(ISUP))*ELMTS*
     1OPERA(K-10,L)*WTS(KK,LL)/FLF(L)
      IF(ROUNDS.LE.0.)GOTO102
      IF(J.EQ.2)GO TO 500
      GOTO(507,506,509,521,513,503,513,502,501,504,102,505,102,102,102,1
     102,505,102,102,102),I-10
  500 GOTO(512,510,511,514,518,522,508,516,515,510,503,517,102,102,102,1
     102,509,519,520,102),I-10
  501 M=1
      GOTO555
  502 M=2
      GOTO555
  503 M=3
      GOTO555
  504 M=4
      GOTO555
  505 M=5
      GOTO555
  506 M=6
      GOTO555
  507 M=7
      GOTO555
  508 M=8
      GOTO555
  509 M=9
      GOTO555
  510 M=10
      GOTO555
  511 M=11
      GOTO555
  512 M=12
      GOTO555
  513 M=13
      GOTO555
  514 M=14
      GOTO555
  515 M=15
      GOTO555
  516 M=16
      GOTO555
  517 M=17
      GOTO555
  518 M=18
      GOTO555
  519 M=19
      GOTO555
  520 M=20
      GOTO555
  521 M=21
      GOTO555
  522 M=22
  555 IF(I.EQ.21.AND.MAXA.LE.2)M=10
      NBR=INDEXF(2,MAXI,N,M,0,2,0,4,0)
      IFIND=(NBR-1)/32+1
      CALL READMS(3,SEKP,32,IFIND)
      INDX=NBR-(NBR/32*32)
      IF(INDX.EQ.0)INDX=32
```

Figure H-1. OVLY2 (TANK) program code (continued).

```
      SS=SSKP(INDX)
      IF(L.EQ.IO)SS=SS*2.
      IF(K.EQ.22.AND.L.EQ.1)N=3
      NBR=INDEXS(1,MAXR,N,M,0,2,K,4,J)
      IFIND=(NBR-1)/32+1
      CALL READMS(3,SSKP,32,IFIND)
      INDX=NBR-(NBR/32*32)
      IF(INDX.EQ.0)INDX=32
      SSS=SSKP(INDX)
      IF(L.EQ.IA)SSS=SSS*2.
      SS=(SS+SSS)/3.
      IF(SS/VICTIM.GT.1.)GOTO102
      PKILL(I-10)=(1.-SS/VICTIM)**(XN*ROUNDS)
      AKILL=AKILL*PKILL(I-10)
      SKILL=SKILL+(1.-PKILL(I-10))
      SHOTS(ITYP,J)=SHOTS(ITYP,J)+ROUNDS
  102 CONTINUE
      TKILL=(1.-AKILL)*VICTIM
      IF(TKILL.LE.0.)GOTO100
      IF(SKILL.LE.0.)GOTO100
      DO 103 I=11,30
      AKILL=TKILL*(1.-PKILL(I-10))/SKILL
      AKILL=IFIX(AKILL*10.+.5)/10.
      ALOSS(I,K)=ALOSS(I,K)+IFIX(AKILL*10.+.001)*PACK(L)
      PLOSS(K,L,2)=PLOSS(K,L,2)+IFIX(AKILL*10.+.001)/10.
      IF(K.GT.15)GOTO108
      ALOSS(I,3)=ALOSS(I,3)+IFIX(AKILL*ATCREW(K-10)*10.+.001)*PACK(L)
      PLOSS(3,L,2)=PLOSS(3,L,2)+IFIX(AKILL*ATCREW(K-10)*10.+.001)/10.
      IF(K.LT.13)GOTO103
  108 ALOSS(I,2)=ALOSS(I,2)+IFIX(AKILL*CREWS(K-12,L)*10.+.001)*PACK(L)
      PLOSS(2,L,2)=PLOSS(2,L,2)+IFIX(AKILL*CREWS(K-12,L)*10.+.001)/10.
      IF((K.NE.21.AND.K.NE.25).OR.IMOUNT.EQ.1.OR.L.EQ.ID)GOTO103
      AKILL=AKILL*6.
      DO 105 KK=3,15
      IF(ELMT(KK,L).LE.0.)GOTO105
      ALOSS(I,KK)=ALOSS(I,KK)+IFIX(AKILL*PLT(KK)*10.+.001)*PACK(L)
      PLOSS(KK,L,2)=PLOSS(KK,L,2)+IFIX(AKILL*PLT(KK)*10.+.001)/10.
  105 CONTINUE
  103 CONTINUE
  100 CONTINUE
      KFLAG=2
      DO 125 J=1,2
      IFLAG=0
      DO 125 I=1,32
      INX=I
      PLOSS(I,J,1)=PLOSS(I,J,2)+PLOSS(I,J,1)
      IF(PLOSS(I,J,1).LE.ELMT(INX,J))GOTO130
      PRINT*,"ALL OF ELMT ",INX," IN FORCE ",J," HAVE BEEN KILLED"
      DO 135 K=11,30
      PLUF=IFIX(ALOSS(K,INX)/PACK(1))/10.
      RED=(ALOSS(K,INX)-IFIX(ALOSS(K,INX)/PACK(1))*PACK(1))/10.
      IF(J.EQ.2)GOTO133
      PLUF=IFIX(PLUF*ELMT(INX,J)/PLOSS(I,J,1)*10.+.5)/10.
      GOTO134
  133 RED=IFIX(RED*ELMT(INX,J)/PLOSS(I,J,1)*10.+.5)/10.
  134 ALOSS(K,INX)=PLUF*PACK(1)*10.+RED*10.
  135 CONTINUE
```

Figure H-1. OVLY2 (TANK) program code (continued).

```
    PLOSS(I,J,1)=FLMT(INX,J)
130 PLOSS(I,J,2)=0.
    IF(PLOSS(I,J,1).LT..1)GOTO125
    IF(IFLAG.EQ.1)GOTO123
    PRINT*
    KIND="RED"
    IF(J.EQ.1)KIND="BLUE"
    IF(IFUN.NE.1)PRINT124,KIND
124 FORMAT(" ",13X,A4," LOSSES TO THIS POINT",/,16X,"ITEM      # LOST")
123 TKILL=PLOSS(I,J,1)
    IF(IFUN.NE.1)PRINT126,INX,TKILL
126 FORMAT(" ",16X,I2,5X,F6.1)
    IFLAG=1
125 CONTINUE
    GOTO11
C        OUTPUT RESULTS.
150 IF(IFUN.EQ.1)GO TO 229
    PRINT153
153 FORMAT("1")
    PRINT*,"*--------------------ARMOR ASSESSMENTS--------------------*"
    PRINT140
140 FORMAT(" I",55X,"I")
    DO 225 J=1,2
    L=1
    IF(J.EQ.L)L=2
    IFLAG=0
    DO 220 K=1,32
    AKILL=PLOSS(K,L,1)
    CLOST=0.
    IF(K.LT.16.AND.K.NE.13)GOTO195
    CLOST=AKILL*CHFMS(K-12,L)
195 IF(AKILL.LT..1.AND.CLOST.LT..1)GOTO220
    IF(IFLAG.EQ.1)GOTO210
    IF(J.EQ.2)GOTO203
    PRINT*,"I                    TOTAL RED LOSSES                     I"
    GOTO205
203 PRINT*,"I                    TOTAL BLUE LOSSES                    I"
205 IFLAG=1
    PRINT*,"I               ITEM     # LOST    CREW LOST              I"
    PRINT140
210 IF(K.LT.16.AND.K.NE.13)GOTO211
    PRINT213,K,AKILL,CLOST
    GOTO220
211 PRINT212,K,AKILL
212 FORMAT(" I",15X,I2,5X,F6.1,27X,"I")
213 FORMAT(" I",15X,I2,5X,F5.1,5X,F6.1,15X,"I")
220 CONTINUE
    PRINT140
    IF(J.EQ.2)GOTO225
    PRINT*,"*---------------------------------------------------------*"
    PRINT140
225 CONTINUE
    PRINT*,"*********************************************************"
    PRINT153
229 IF(IFUN.EQ.1)PRINT*," ARMOR ASSESSMENT PRINTED HERE"
    CALL FLUSH(1)
    CALL ...(11,25,1,50)
230 ...
```

Figure H-1. OVLY2 (TANK) program code (concluded).

APPENDIX I

OVLY 3 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX I

OVLY 3 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN code and variable list for the OVLY 3 (INFANT) program. INFANT is the routine that assesses dismounted infantry combat between the opposing forces. Table I-1 lists the program variables, and figure I-1 is the FORTRAN source code listing.

Table I-1. Program variables for OVLY3 (INFANT). (Continued next page.)

Variable	Description
A	Ambush personnel casualty rate
AIL	Infantry attacker losses
AT	Personnel allocated to infantry attack
ATRIT	Personnel casualties for ambushed unit
D	Defender's personnel casualty rate
DIL	Infantry defender losses
DT	Personnel allocated to infantry defense
DTRIT	Personnel casualties for ambushing unit
F	Fraction of maneuver forces committed
FAC	Casualty rate resolution factor
GFPR	Ground combat firepower ratio
GFPS	Ground combat firepower scores
HR	Hours of combat for assessment
HRC	Hours of conventional combat
I	Target weapon index
IAA	Attacker index in ambush
IEL	Defender index
IFLAG	Logic flag
INDEX	Target weapon flag

Table I-1. Program variables for OVLY3 (INFANT) (Concluded).

Variable	Description
INX	Hours of infantry attack
J	Force index
KIND	Force color
L	Target force index
STALE	Casualty rate index determiner array
TABLE	Ground combat personnel casualty rate
TABLE3	Ambush personnel casualties

NOTE: All COMMON Variables are defined in table F-1.

```
      OVERLAY(INFANT,3,0)
      PROGRAM OVLY3
      COMMON IA,IE,IF,IANGLG,ITERRN,IVIS,IMOUNT,MINES,DFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 DF(2),FSSF(2),FACK(2),
     3 ELMT(80,2),ALOSS(80,80),SHOTS(33,2),CKILL(53,2)
      COMMON/DATA/FPS(83,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PUN(6,2,2),PLI(15),KEY(41)
      DIMENSION GFPS(2),TABLE(7,2,4),STALE(5),FAC(2)
     1,TABLE3(5,2)
      DATA (((TABLE(I,J,K),J=1,2),I=1,7),K=1,4)/
     1.12,.05,.08,.06,.05,.06,.05,.08,.05,.08,.05,.03,.04,.10,.09,.06,.0
     23,.05,.04,.07,.04,.07,.03,.09,.13,.10,.02,.11,.09,.05,.05,.05,.04,
     3.05,.03,.06,.03,.08,.02,.05,.02,.10,.08,.02,.05,.03,.04,.03,.03,.0
     4,.03,.05,.02,.08,.02,.08/
      DATA (STALE(I),I=1,4)/1.,2.,2.5,3.1/
      DATA ((TABLE3(I,J),J=1,2),I=1,5)/10.,20.,20.,15.,35.,10.,50.,5.,
     1 75.,2./
710   PRINT*,"DO YOU WISH TO PROCESS INFANTRY ASSESSMENTS?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT6,INX
      IF(INX.EQ."Y")GOTO711
      IF(INX.EQ."N")GOTO600
      PRINT2
      GOTO710
711   PRINT*,"               INFANTRY ASSESSMENTS"
      PRINT*," "
15    PRINT*,"ENTER THE FRACTION OF MANEUVER FORCES COMMITTED (MAX 1.)"
      READ*,F
      IF(IRUN.EQ.1)WRITE(5,*)F
      IF(IRUN.EQ.3)PRINT*,F
      IF(F.GT.0..AND.F.LE.1.)GOTO5
      PRINT3
3     FORMAT("INCORRECT RESPONSE - TRY AGAIN")
      GO TO 15
5     PRINT*,"DO TANKS SUPPORT THE DISMOUNTED INFANTRY IN THIS SECTOR"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT6,INX
1     FORMAT(1A1)
6     FORMAT(" ",1A1)
      IF(INX.EQ."Y")GO TO 30
      IF(INX.EQ."N")GO TO 30
      PRINT2
2     FORMAT(" INCORRECT - RESPONSE MUST BE Y OR N - TRY AGAIN")
      GO TO 5
30    INDEX=0
      IF(INX.EQ."Y")INDEX=1
      AIL=0.
      IL=0.
      DO 35 IEL=1,2
      GFPS(IEL)=0.
      FAC(IEL)=1.
      IF(ELMT(3,IEL)*F.GT.72.)FAC(IEL)=2.
      DO 35 I=6,30
      IF(I.EQ.20.OR.I.EQ.29.OR.(INX.EQ."N".AND.I.GE.16.AND.I.LE.19))GOTO
```

Figure I-1. OVLY3 (INFANT) program code.

```
   135
      GFPS(IEL)=GFPS(IEL)+ELMT(I,IEL)*FPS(I,IEL)
   35 CONTINUE
      IF(GFPS(ID).GE.1.)GOTO49
      PRINT*,"THERE ARE NO DEFENDERS--ASSESSMENTS CANNOT BE MADE."
      GOTO500
   49 PRINT*,"ENTER # HOURS OF INFANTRY ATTACK (MAX = 6.)."
      READ*,HR
      IF(IRUN.EQ.1)WRITE(5,*)HR
      IF(IRUN.EQ.3)PRINT*,HR
      HRL=HR
      IF(HR.GT.0..AND.HR.LE.6.)GOTO36
      PRINT3
GOTO49
   36 PRINT*,"ARE AMBUSH TACTICS BEING EMPLOYED"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GO TO 50
      IF(INX.EQ."N")GO TO 40
      PRINT2
      GO TO 36
   40 GFFR=GFPS(IA)*APOS(IP)/(GFPS(ID)*DPOS(IENGAG))
      DO 45 I=1,6
      IF(GFFR.LE.I*.5)GO TO 46
   45 CONTINUE
      I=7
   46 INX=I
      GO TO(43,44,44,41,41,42),IENGAG
   41 I=1
      GO TO 47
   42 I=2
      GO TO 47
   43 I=3
      GO TO 47
   44 I=4
   47 HR=HR-1
      IF(HR.LE.0.)GOTO105
      IFLAG=0
      ATFIT=TABLE(INX,1,I)/FAC(IA)
      DTFIT=TABLE(INX,2,I)/FAC(ID)
   51 IF(IA.EQ.1)GO TO 48
      AT=ELMT(3,2)*F-AIL
      DT=ELMT(3,1)*F-DIL
      GO TO 52
   48 AT=ELMT(3,1)*F-AIL
      DT=ELMT(3,2)*F-DIL
   52 AIL=AIL+AT*(1.-(1.-ATFIT)**HR)
      DIL=DIL+DT*(1.-(1.-DTFIT)**HR)
      IF(IFLAG.NE.0)GOTO65
      GO TO 135
   50 HR=HR-1.
      IF(HR.GT.1.)HR=1.
      PRINT*,"IS DEF AMBUSHING REG"
      READ11,IFLAG
      IF(IRUN.EQ.1)WRITE(5,11)IFLAG
      IF(IRUN.EQ.3)PRINT8,IFLAG
```

Figure I-1. OVLY3 (INFANT) program code (continued).

```
      IF(IFLAG.EQ."Y")GO TO 55
      IF(IFLAG.EQ."N")GO TO 60
      PRINT8
      GO TO 50
   55 GFPR=4.5*GFPS(1)/GFPS(2)
      IAA=1
      GO TO 65
   60 GFPR=4.5*GFPS(2)/GFPS(1)
      IAA=2
   65 DO 70 I=1,4
      IF(GFPR.LT.STALF(I))GO TO 75
   70 CONTINUE
      I=5
   75 A=TABLE3(I,2)/100.
      D=TABLE3(I,1)/100.
      IF(IAA.NE.IA)GO TO 84
      AT+IT=A
      DT+IT=D
      GO TO 81
   84 AT+IT=D
      DT+IT=A
      GO TO 81
  100 IF(IRUN.EQ.1)GO TO 599
      PRINT*,"*------------------INFANTRY ASSESSMENTS------------------*"
  105 FORMAT(" I",55X,"I")
      DO 2.0 J=1,2
      PRINT 105
      IFLAG=0
      KIND="BLUE"
      L=1
      IF(L.NE.J)GOTO205
      KIND="RED"
      L=2
  205 DO 200 I=1,INDEX
      IF(ALMT(I,J).LE.0.)GOTO200
      A=FIL
      IF(L.EQ.1D)A=DIL
      IF(1.EQ.2.OR.I.EQ.4.OR.I.EQ.5)GOTO200
      IF(I.EQ.7.AND.L.EQ.2)GOTO200
      A=A*PLT(I)
      A=IFIX(A*10.+.5)/10.
      ALOSS(3,I)=ALOSS(3,I)+IFIX(A*1 .+.001)*PACK(L)
      IF(A.LT..1)GOTO200
      IF(IFLAG.EQ.1)GOTO21.
      PRINT202,KIND
  202 FORMAT(" I",18X,A4,1X,"INFANTRY LOSSES",17X,"I")
      PRINT*,"I                    ITEM        #LOST                    I"
      IFLAG=1
  210 PRINT215,I,A
  215 FORMAT(" I",20X,I2,7X,F6.1,20X,"I")
  200 CONTINUE
      PRINT105
      PRINT*,"*=========================================================*"
  599 IF(IRUN.EQ.1)PRINT*," TOTAL INFANTRY LOSSES PRINTED HERE"
      CALL LOSS(3,3,1,15)
  600 ...
```

Figure I-1. OVLY3 (INFANT) program code (concluded).

APPENDIX J

OVLY 4 PROGRAM CODES AND LISTS OF VARIABLES

APPENDIX J

OVLY 4 PROGRAM CODES AND LISTS OF VARIABLES

This appendix contains the FORTRAN listings and variable lists for the main program, MINE, and the subroutine, FASCAM, of the OVLY 4 program. The MINE routine assess attacker force losses to conventional minefields, and the FASCAM subroutine makes the assessments for FASCAM minefields. The MINE program variables are listed in table J-1, with the FORTRAN source code listing in figure J-1. For the FASCAM subroutine, table J-2 and figure M-2 give the program variable list and the FORTRAN code, respectively.

Table J-1. Program variables for MINE (continued next page).

Variable	Description
AFRONT	Minefield frontage input variable
AKILL	Attacker weapon system kills
ATDEN	Antitank minefield (MF) density per square meter
ATFAC	Percent tank losses by antitank mines
BMPL	Mine planter platoons
CLOST	Crewmen losses for productive time lost due to enemy
FROBY	Minefield frontage bypassed by attacker
FRONT	Potential minefield frontage
HOURS	Hours required to lay MF strip
HRMAN	Man-hours available for emplacement of mines
HRREQ	Man-hours required to manually emplace mines
IND	Type of mine employment index
INX	Input response variable
J	Antitank mine density index
K	Target weapon system index

Table J-1. Program variables for MINE (concluded).

Variable	Description
KIND	Force color
KK	Infantry weapon system index
NUMEN	Number men to emplace mines
P	Percent of force entering minefield
PERCAS	Percent AP mines personnel casualties
PERCOV	Percent of unit's front covered by mines
PHR	Man hours available
PLOSS	Total victims killed
PMFNBY	Percent of MF not bypassed by attacker
RNMPH	Mine planter hours available
STRIPW	Minefield strip width
TRZONE	Terrain trafficable by armor
WDEGF	Work degradation factor
X	Mine density input variable

NOTE; All COMMON variables are defined in table F-1.

```
      OVERLAY(MINE,4,0)
      PROGRAM OVLY4
      COMMON IA,ID,IP,IENGAG,ITERN,IVIS,IMOUNT,MINES,SFPP,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSCF(2),PACK(2),
     3 ELMT(80,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),OPENS(53,2),APOS(12),DPOS(5),
     1 PSN(5,2,2),PLT(15),KEY(41)
      DIMENSION ATDEN(5),ATFAC(5),HRREQ(5),PLOSS(32,2,2)
      DATA(ATDEN(J),J=1,5)/.2,.5,1.,2.,3./
      DATA(ATFAC(J),J=1,5)/.1,.3,.6,.4,.9/
      DATA(HRREQ(J),J=1,5)/3*234.,279.,323./
    3 FORMAT(" INCORRECT - RESPONSE MUST BE Y OR N - TRY AGAIN")
    4 FORMAT(" NUMBER NOT WITHIN DOCTRINES BOUNDARY - TRY AGAIN")
    7 FORMAT(1A1)
    8 FORMAT(" ",1A1)
    5 PRINT*,"DO YOU WISH TO PROCESS MINE ASSESSMENTS?"
      READ7,INX
      IF(IRUN.EQ.1)WRITE(5,7)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y") GO TO 556
      IF(INX.EQ."N") GO TO 1000
      PRINT3
      GO TO 5
  556 PRINT*,"SELECT TYPE OF MINE EMPLOYMENT"
      READ*,IND
      IF(IRUN.EQ.1)WRITE(5,*)IND
      IF(IRUN.EQ.3)PRINT*,IND
      IF(IND.EQ.1)GOTO32
      IF(IND.EQ.2)GOTO33
      IF(IND.EQ.3)GOTO999
      IF(IND.EQ."T")GOTO11
      PRINT4
   11 PRINT*,"FOR CONVENTIONAL MINES............ENTER 1"
      PRINT*,"    FASCAM MINES..................ENTER 2"
      PRINT*,"    ***TO END***..................ENTER 3"
      GOTO556
C
C        ASSUMES DEFENDER EMPLACED MINFIELDS
C
   33 CALL FASCAM(PLOSS)
      GOTO556
   32 KIND="BLUE"
      IF(ID.EQ.2)KIND="RED"
      PRINT*,"ARE MINES LAID PRIOR TO COMMENCEMENT OF HOSTILITIES?"
      READ7,INX
      IF(IRUN.EQ.1)WRITE(5,7)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO29
      IF(INX.EQ."N")GOTO22
      PRINT3
      GOTO32
   29 MINSF=.9
      GOTO558
   22 MINSF=.7
  558 PRINT558,KIND
  558 FORMAT(" WILL ",A4," HAVE THE CAPABILITY TO EMPLOY MECHANICAL",
```

Figure J-1. MINE program code.

```
    1 " MINE PLANTERS?")
      READ7,INX
      IF(IRUN.EQ.1)WRITE(5,7)INX
      IF(IRUN.EQ.3)PRINT6,INX
      IF(INX.EQ."Y") GO TO 515
      IF(INX.EQ."N") GO TO 10
      PRINT3
      GO TO 558
  515 PRINT*,"ENTER NUMBER OF MECHANICAL MINE PLANTER PLATOONS (MAX 30)"
C              MECHANICAL EMPLACEMENT OF MINEFIELD
      READ*,BMPL
      IF(IRUN.EQ.1)WRITE(5,*)BMPL
      IF(IRUN.EQ.3)PRINT*,BMPL
      IF(BMPL.GE.1.AND.BMPL.LE.30) GO TO 52
      PRINT4
      GO TO 515
   52 PRINT*,"ENTER NUMBER OF AVAILABLE MINE PLANTER HOURS (MAX 300)"
      READ*,RNMPH
      IF(IRUN.EQ.1)WRITE(5,*)RNMPH
      IF(IRUN.EQ.3)PRINT*,RNMPH
      IF(RNMPH.GE.1.AND.RNMPH.LE.300)GOTO53
      PRINT4
      GO TO 52
   53 IF(KIND.EQ."REG")GOTO54
      HOURS=6.
      STRIPW=2300.
      GOTO55
   54 HOURS=2.
      STRIPW=1000.
   55 FRONT=(BMPL*RNMPH*WDEGF)/HOURS*STRIPW
      J=2
      GOTO56
   10 PRINT*,"ENTER NUMBER OF MEN USED TO EMPLACE MINES (MAX 1000)"
      READ*,NUMEN
      IF(IRUN.EQ.1)WRITE(5,*)NUMEN
      IF(IRUN.EQ.3)PRINT*,NUMEN
      IF(NUMEN.GE.1.AND.NUMEN.LE.1000)GOTO14
      PRINT4
      GO TO 10
   14 PRINT*,"ENTER HOURS AVAILABLE FOR EMPLACEMENT OF MINES (MAX 300)"
      READ*,HRMAN
      IF(IRUN.EQ.1)WRITE(5,*)HRMAN
      IF(IRUN.EQ.3)PRINT*,HRMAN
      IF(HRMAN.GE.1.AND.HRMAN.LE.300) GO TO 13
      PRINT4
      GO TO 14
   13 PH=NUMEN*HRMAN*WDEGF
   17 PRINT*,"SELECT MINEFIELD DENSITY"
      READ*,J
      IF(IRUN.EQ.1)WRITE(5,*)J
      IF(IRUN.EQ.3)PRINT*,J
      IF(J.EQ."T")GOTO16
      IF(J.GE.1.AND.J.LE.5)GOTO16
      PRINT4
   16 PRINT*,"FOR DENSITY .0013 MINE/SQ METER.......ENTER 1"
      PRINT*,"              .0033 MINE/SQ METER.......ENTER 2"
      PRINT*,"              .0066 MINE/SQ METER.......ENTER 3"
```

Figure J-1. MINE program code (continued).

```
      PRINT*,"              .015  MINE/SQ METER........ENTER 4"
      PRINT*,"              .0200 MINE/SQ METER........ENTER 5"
      GO TO 17
   18 FRONT=PHR/HR*EQ(J)*100.
      FRONT=IFIX(FRONT+.5)
   65 PRINT44,FRONT
   44 FORMAT(" POTENTIAL MINEFIELD FRONTAGE IS ",F8.0)
      PRINT*," "
      PRINT*,"ENTER ACTUAL MF FRONTAGE (MAX=POTENTIAL)"
      READ*,AFRONT
      IF(IRUN.EQ.1)WRITE(5,*)AFRONT
      IF(IRUN.EQ.3)PRINT*,AFRONT
      IF(AFRONT.LE.FRONT)GOTO70
      PRINT4
      GOTO65
   70 FRONT=AFRONT
   19 PRINT*,"ENTER FRACTION OF MINE FIELD NOT BYPASSED BY ATTACKER (MAX
     1=1.0)"
      READ*,PMFNBY
      IF(IRUN.EQ.1)WRITE(5,*)PMFNBY
      IF(IRUN.EQ.3)PRINT*,PMFNBY
      IF(PMFNBY.EQ."T")GOTO21
      IF(PMFNBY.GE.0..AND.PMFNBY.LE.1.)GOTO20
      PRINT4
   21 PRINT*,"FOR EXAMPLE: 0. MEANS ALL OF THE MF CAN BE BYPASSED"
      PRINT*,"OR,            1. MEANS NONE OF THE MF CAN BE BYPASSED"
      GO TO 19
   20 FRONBY=FRONT*PMFNBY
      PRINT770,FRONBY
  770 FORMAT(" ENTER AMOUNT OF TRAFFICABLE TERRAIN (",F8.0,"-100000. M)"
     1)
      READ*,TRZONE
      IF(IRUN.EQ.1)WRITE(5,*)TRZONE
      IF(IRUN.EQ.3)PRINT*,TRZONE
      IF(TRZONE.GE.FRONBY.AND.TRZONE.LE.100000) GO TO 40
      PRINT4
      GO TO 20
   40 PRINT*,"ENTER AP MINE DENSITY(SQ METER) - (MIN=.013-MAX=.160)"
      READ*,X
      IF(IRUN.EQ.1)WRITE(5,*)X
      IF(IRUN.EQ.3)PRINT*,X
      IF(X.GE..013.AND.X.LE..160)GOTO42
      PRINT4
      GOTO40
   42 [illegible]=X*150.
      [illegible]
      [illegible]
      [illegible]
      IF([illegible].LE.1) GO TO 39
      PRINT*,"[illegible] OF ZONE [illegible] MUST BE BETWEEN ZERO(0) AND"
      PRINT*,"ONE(1) - CHECK THE TRAFFICABLE ZONE DEFENDED TO SEE"
      PRINT*,"IF IT IS LARGER THEN THE RESULT OF MULTIPLYING MF"
      PRINT*,"FRONTAGE TIMES PERCENT OF MF NOT BYPASSED"
      GO TO 20
   39 PRINT*,"ENTER [illegible] (DECIMAL) OF FORCES ENTERING MF(MAX=.5)"
      READ*,P
      IF(IRUN.EQ.1)WRITE(5,*)P
```

Figure J-1. MINE program code (continued).

```
      IF(IMOUNT.EQ.3)PRINT*,P
      IF(P.GE.0..AND.P.LE..5)GOTO26
      PRINT4
      GOTO39
   26 DO 120 K=1,32
      IF(ELMT(K,IA)-PLOSS(K,IA,2).LE.0.)GOTO120
      IF(K.LT.16.AND.K.NE.3)GOTO120
      IF(K.EQ.3.AND.IMOUNT.EQ.2)GOTO120
      IF(K.NE.3)GO TO 101
      AKILL=PERCOV*(ELMT(3,IA)-PLOSS(3,IA,2))*P*PERCAS
      AKILL=IFIX(AKILL*10.+.5)/10.
      GO TO 110
  101 AKILL=PERCOV*(ELMT(K,IA)-PLOSS(K,IA,2))*P*ATFAC(J)
      AKILL=IFIX(AKILL*10.+.5)/10.
      ALOSS(5,K)=ALOSS(5,K)+IFIX(AKILL*10.+.001)*PACK(IA)
      PLOSS(K,IA,1)=PLOSS(K,IA,1)+IFIX(AKILL*10.+.001)/10.
      IF(K.EQ.1)GOTO100
      CLOST=AKILL*CREWS(K-12,IA)
      GOTO105
  100 CLOST=AKILL*2.
  105 ALOSS(5,2)=ALOSS(5,2)+IFIX(CLOST*10.+.001)*PACK(IA)
      PLOSS(2,IA,1)=PLOSS(2,IA,1)+IFIX(CLOST*10.+.001)/10.
      IF(IMOUNT.EQ.1.OR.(K.NE.21.AND.K.NE.25))GOTO120
      AKILL=IFIX(AKILL*60.+.5)/10.
  110 DO 115 KK=3,15
      IF(ELMT(KK,IA)-PLOSS(KK,IA,2).LE.0.)GOTO115
      ALOSS(5,KK)=ALOSS(5,KK)+IFIX(AKILL*PLT(KK)*10.+.001)*PACK(IA)
      PLOSS(KK,IA,1)=PLOSS(KK,IA,1)+IFIX(AKILL*PLT(KK)*10.+.001)/10.
  115 CONTINUE
  120 CONTINUE
      IF(IPRNT.EQ.1)GOTO999
      INN=1
      DO 180 K=1,32
      AKILL=PLOSS(K,IA,1)
      IF(AKILL.LT..1)GOTO180
      IF(K.GE.2.AND.K.LE.15)GOTO 140
      IF(K.EQ.1)GOTO135
      CLOST=AKILL*CREWS(K-12,IA)
      GOTO140
  135 CLOST=AKILL*2.
  140 IF(INN.EQ.1)GOTO155
      PRINT145
  145 FORMAT("1")
      PRINT*,"*---------------------MINEFIELD ASSESSMENTS---------------------*"
      PRINT150
  150 FORMAT(" I",[illegible],"I")
      PRINT*,"I                      ATTACKED LOSSES                      I"
      PRINT*,"I              ITEM      # LOST    CREW LOST               I"
      PRINT150
      INN=1
  155 IF(K.GE.2.AND.K.LE.15)GOTO170
      PRINT160,K,AKILL,CLOST
  160 FORMAT(" I",17X,I2,3X,F5.1,5X,F5.1,16X,"I")
      GOTO180
  170 PRINT175,K,AKILL
  175 FORMAT(" I",17X,I2,3X,F5.1,26X,"I")
  180 CONTINUE
```

Figure J-1. MINE program code (continued).

```
      IF(INX.EQ.0)GO TO 998
      PRINT150
      PRINT*,"*=========================================================*"
      PRINT145
  998 IF(IRUN.EQ.1)PRINT*," LOSSES TO MINEFIELD PRINTED HERE"
      DO 900 K=1,32
      DO 900 J=1,2
      PLOSS(K,J,2)=PLOSS(K,J,2)+PLOSS(K,J,1)
      PLOSS(K,J,1)=0.
  900 CONTINUE
      GOTO950
  999 CALL LOSS(5,5,16,30)
 1000 END
```

Figure J-1. MINE program code (concluded).

Table J-2. Program variables for FASCAM.

Variable	Description
AKILL	Attacker weapon system kills
CLOST	Crewmen lost
FATCAS	Percent tank casualties by FASCAM mines
FPCAS	Percent personnel casualties by FASCAM mines
FROBY	Minefield frontage bypassed by attacker
FRONT	Minefield frontage
II	Type of FASCAM delivery system
INX	Input response variable
J	Force index
K	Target weapon index
KK	Target weapon index
P	Percent of force entering minefield
PERCOV	Percent of units front covered by mines
PLOSS	Total victims killed
PMFNBY	Percent of MF not bypassed by attacker
TRZONE	Terrain trafficable by armor

NOTE: All COMMON variables are defined in table F-1.

```
      SUBROUTINE FASCAM(FLOSS)
      COMMON IA,IU,IF,IDAGAG,ITERRN,IVIS,IMOUNT,MINES,SFPR,FSFPK,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PICF(2),
     3 FLMT(50,2),FLOSS(53,83),SHOTS(55,2),CKILL(53,2)
      COMMON/DATA/FPS(90,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PLNE(6,2,2),PLT(15),KEY(41)
      DIMENSION FATCAS(3),FPCAS(3),PLOSS(32,2,2)
      DATA(FATCAS(I),I=1,3)/.2,.6,.2/
      DATA(FPCAS(I),I=1,3)/.08,.24,.0/
    4 FORMAT(" NUMBER NOT WITHIN SELECTION BOUNDARY - TRY AGAIN")
   10 PRINT*,"SELECT TYPE OF FASCAM DELIVERY"
      READ*,II
      IF(IRUN.EQ.1)WRITE(5,*)II
      IF(IRUN.EQ.3)PRINT*,II
      IF(II.EQ."T")GOTO12
      IF(II.GE.1.AND.II.LE.3)GOTO14
      PRINT4
   12 PRINT*,"FOR  ARTILLERY.........................ENTER 1"
      PRINT*,"     GEMMS.............................ENTER 2"
      PRINT*,"     SLUMINES..........................ENTER 3"
      GOTO10
   14 PRINT*,"ENTER MINEFIELD FRONTAGE (MAX= 100000)"
      READ*,FRONT
      IF(IRUN.EQ.1)WRITE(5,*)FRONT
      IF(IRUN.EQ.3)PRINT*,FRONT
      IF(FRONT.GE.10...AND.FRONT.LE.100000.)GOTO16
      PRINT4
      GOTO14
   16 PRINT*,"ENTER FRACTION OF MF NOT BYPASSED BY ATTACKER (MAX=1.)"
      READ*,PMFNBY
      IF(IRUN.EQ.1)WRITE(5,*)PMFNBY
      IF(IRUN.EQ.3)PRINT*,PMFNBY
      IF(PMFNBY.EQ."T")GOTO18
      IF(PMFNBY.GE..5.AND.PMFNBY.LE.1.)GOTO20
      PRINT4
   18 PRINT*,"FOR EXAMPLE: 0. MEANS ALL OF MF CAN BE BYPASSED"
      PRINT*,"OR              1. MEANS NONE OF MF CAN BE BYPASSED"
      GOTO16
   20 FRONBY=FRONT*PMFNBY
      PRINT22,FRONBY
   22 FORMAT(" ENTER AMOUNT OF TRAFFICABLE TERRAIN (",F8.0,"-100000. M)"
     1)
      READ*,TRZONE
      IF(IRUN.EQ.1)WRITE(5,*)TRZONE
      IF(IRUN.EQ.3)PRINT*,TRZONE
      IF(TRZONE.GE.FRONBY.AND.TRZONE.LE.100000.)GOTO24
      PRINT4
      GOTO20
   24 PRINT*,"ENTER PERCENT(DECIMAL) OF FORCE ENTERING MF (MAX=.5)"
      READ*,P
      IF(IRUN.EQ.1)WRITE(5,*)P
      IF(IRUN.EQ.3)PRINT*,P
      IF(P.GT.0..AND.P.LE..5)GOTO26
      PRINT4
      GOTO24
   26 FRACOV=FRONBY/TRZONE
```

Figure J-2. FASCAM program code (continued next page).

```
      DO 120 K=1,32
      IF(ELMT(K,IA)-PLOSS(K,IA,2).LE.0.)GOTO120
      IF(K.LT.16.AND.K.NE.3)GOTO120
      IF(K.EQ.3.AND.IMOUNT.EQ.2)GOTO120
      IF(K.NE.3)GO TO 103
      AKILL=PERCOV*(ELMT(3,IA)-PLOSS(3,IA,2))*P*FPCAS(II)
      AKILL=IFIX(AKILL*10.+.5)/10.
      GOTO110
  103 AKILL=PERCOV*(ELMT(K,IA)-PLOSS(K,IA,2))*P*FATCAS(II)
      AKILL=IFIX(AKILL*10.+.5)/10.
      ALOSS(?,K)=ALOSS(?,K)+IFIX(AKILL*10.+.001)*PACK(IA)
      PLOSS(K,IA,1)=PLOSS(K,IA,1)+IFIX(AKILL*10.+.001)/10.
      IF(K.EQ.1)GOTO1..
      CLOST=AKILL*CREWS(K-12,IA)
      GOTO105
  100 CLOST=AKILL*2.
  105 ALOSS(?,2)=ALOSS(?,2)+IFIX(CLOST*10.+.001)*PACK(IA)
      PLOSS(2,IA,1)=PLOSS(2,IA,1)+IFIX(CLOST*10.+.001)/10.
      IF(IMOUNT.EQ.1.OR.(K.NE.21.AND.K.NE.25))GOTO120
      AKILL=IFIX(AKILL*10.+.5)/1..
  110 DO 115 KK=3,15
      IF(ELMT(KK,IA)-PLOSS(KK,IA,2).LE.0.)GOTO115
      ALOSS(?,KK)=ALOSS(?,KK)+IFIX(AKILL*PLT(KK)*10.+.001)*PACK(IA)
      PLOSS(KK,IA,1)=PLOSS(KK,IA,1)+IFIX(AKILL*PLT(KK)*10.+.001)/10.
  115 CONTINUE
  120 CONTINUE
      IF(I-U..EQ.1)GOTO..6
      I..Y=0
      DO 180 K=1,32
      AKILL=PLOSS(K,IA,1)
      IF(AKILL.LT..1)GOTO180
      IF(K.GE.?.AND.K.LE.15)GOTO1..
      IF(K.EQ.1)GOTO13.
      CLOST=AKILL*CREWS(K-12,IA)
      GOTO14.
  135 CLOST=AKILL*2.
  140 IF(IN..EQ.1)GOTO155
      PRINT149
  149 FORMAT("1")
      PRINT*,"*--------------------FASCAM ASSESSMENTS-----------------*"
      PRINT150
  150 FORMAT(" I",9.X,"I")
      PRINT*,"I                          ATTACKER LOSSES                           I"
      PRINT*,"I                     ITEM     # LOST   CREW LOST                    I"
      PRINT150
      IN..=1
  155 IF(K.GE.2.AND..K.LE.1.)GOTO170
      PRINT16.,.,AKILL,CLOST
  16. FORMAT(" I",17X,I2,.X,F5.1,9X,F5.1,16X,"I")
      GOTO18.
  170 PRINT175,K,AKILL
  175 FORMAT(" I",17X,I2,.X,F5.1,25X,"I")
  180 CONTINUE
      IF(IN...E...) TO ...
      PRINT150
      PRINT*,"*=====================================================*"
      PRINT149
```

Figure J-2. FASCAM program code (continued).

```
999 IF(IPUN.EQ.1)PRINT*," LOSSES TO MINEFIELD PRINTED HERE"
    DO 900 K=1,32
    DO 900 J=1,2
    PLOSS(K,J,2)=PLOSS(K,J,2)+PLOSS(K,J,1)
    PLOSS(K,J,1)=0.
900 CONTINUE
    RETURN
    END
```

Figure J-2. FASCAM program code (concluded).

APPENDIX K

OVLY 5 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX K

OVLY 5 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN source code listing and variable list for the OVLY 5 (AHAD) program. The AHAD routine processes assessments for combat in which attack helicopters are firing at ground maneuver units while being engaged by air defense weapons. The FORTRAN source code listing of AHAD is given in figure K-1; the program variables are listed in table K-1.

Table K-1. Program variables for OVLY5 (AHAD) (continued next page).

Variable	Description
AC	Number of helicopters entered in cell
ACAV	Helicopter operational availabilities
ACCREW	Helicopter crewmen losses
ACKILL	Mission helicopter losses
ACLOST	Popup helicopter losses
AHKILL	Cumulative probability of survival against helicopters
AIRLOSS	Total helicopter losses
AKILL	Helicopter survival probability against all AD
AOFA	Number of avenues of approach
APOP	Helicopter ordnance success rates of fire
CELL	Helicopter attack cell configuration array
CLOST	Ground weapons crewmen losses
EXP	Total number of helicopter exposures to AD fire
FDF	Fire distribution factor
FRAC	Loss apportionment factor
GFKILL	Mission ground force losses
GIS	Mounted/dismounted infantry materiel loss factor
GNDLOS	Total ground force losses

Table K-1. Program variables for OVLY5 (AHAD)(continued).

Variable	Description
H	Hours of flying time for helicopters
HELI	Number of helicopters remaining in a force
I	Ground weapon system index
IABORT	Mission abort flag
IECM	Electronic countermeasure index
IEL	Infantry weapon loss calculation index
IFLAG	Display header flag
II	Ground weapon system index
IN	AD firer index
INX	Input response variable
ITGT	Ground weapon target type index
ITYP	Ordnance type index
IWP	Infantry weapon index
J	Ground force index
JFLAG	Helicopter crew loss display flag
JPSN	Positioning units index for contact
K	Helicopter type index
KIND	Force color
KK	Helicopter type index
KTRL	AD weapon control status factor index

Table K-1. Program variables for OVLY5 (AHAD)(continued).

Variable	Description
L	Helicopter force index
N	Cell popup index
NN	Cell popup counter index
NPOP	Number of helicopter popups per sortie
NPOPUP	Number of cell popups
OPAV	Weapon system operational availability
ORD	Helicopter ordnance loads
ORDEXP	Helicopter ordnance expenditure
PA	Helicopter percent acquisition factor
PHKILL	Weapon system survival probability against helicopter
PK	Helicopter probability of kill array
PKILL	Helicopter survival probability against AD weapon
POPORD	Helicopter per popup ordnance expenditure
PROB	Helicopter averaged PK against target
PROB1	Helicopter PK against target in defilade
PROB2	Helicopter PK against target in open
ROUNDS	Total helicopter rounds fired.
S	AD weapons suppression factor
SA	Helicopter sorties available
SFACT	Suppression factor coefficient

Table K-1. Program variables for OVLY5 (AHAD)(concluded).

Variable	Description
SH	Helicopter suppression factor
SHKILL	Loss apportionment denominator for helicopters
SKILL	Loss apportionment denominator for AD weapons
SSK	AD single engagement kill probabilities
TMASK	Terrain masking factors
TNOW	Current total number of helicopters in cell
TSTART	Initial total number of helicopters in cell
V	Visibility degradation factor
VICTIM	Total ground weapon system targets for helicopters
VKILL	Ground weapon systems killed by helicopters
WEAPC	AD weapon control status factors
WEIGHT	AD target weighting factor

NOTE: All COMMON variables are defined in table F-1.

```
  OVERLAY(AHAD,5,0)
  PROGRAM OVLY5
  COMMON IA,IO,IP,IEDGAG,ITERN,IVIS,IMOUNT,MINES,OFPR,FSFPR,FPR,
 1 ATIME,IFIRST,IRUN,
 2 SF(2),FSFF(2),PACK(2),
 3 FLMT(82,2),ALOSS(80,82),SHOTS(35,2),CKILL(53,2)
  COMMON/DATA/FPS(3,2),CREWS(53,2),APOS(12),OPOS(5),
 1 FUN(6,2,2),PLT(12),KEY(41)
  DIMENSION GRD(7,2,4),PK(96),SSK(12,2,2),V(5),KIND(2),WEAPC(3),
 1SFACT(12,2),PKILL(12),PHKILL(7),TMASK(4),APOP(4,2),SA(7),CELL(7),
 2NPOP(7,2),POPORD(7,2,4),GNDLOS(8,42,2),AIFLOS(13,7,2),ACKILL(7),
 3DPAV(42,2),GFKILL(32),ACLOST(7),ACCREW(2),ACAV(7,2)
  DATA(((GRD(I,J,K),K=1,4),I=1,7),J=1,2)/0.,14.,0.,4.,0.,52.,6*0.,
 18.,15*0.,128.,1.,4*0.,4.,124.,1.,13*0./
  DATA((SFACT(I,J),I=1,12),J=1,2)/2.86,0.,3.52,3*0.,3.52,2.86,3.52,
 12.86,3.52,2.86,4*3.52,2*0.,3.52,2.86,0.,2.86,3.52,2.86/
  DATA((DPAV(I,J),I=1,42),J=1,2)/9*1.,0.,.95,.81,.93,.95,.91,
 12*.72,.67,.78,.67,2*.81,2*0.,.9,.81,.9,3*0.,.6,0.,.6,3*0.,
 2.84,2*.88,3*.78,8*1.,0.,2*1.,0.,2*.95,2*.91,.85,
 33*.78,.62,4*.81,0.,3*.81,0.,4*.85,2*0.,2*.83,0.,3*.85/
  DATA((ACAV(I,J),I=1,7),J=1,2)/2*.65,0.,.81,.77,.75,.63,.65,2*0.,
 1.81,0.,.7,.63/
  DATA(V(I),I=1,5)/1.,.85,.65,.45,.3/
  DATA(KIND(I),I=1,2)/"BLUE","RED"/
  DATA(WEAPC(I),I=1,3)/.8,.6,.1/
  DATA(TMASK(I),I=1,4)/4*.69/
  DATA((APOP(I,J),I=1,4),J=1,2)/.3,10.,2.,.8,.8,25.,2.,.8/
  DATA(((AIFLOS(K,I,J),K=1,13),I=1,7),J=1,2)/182*1./
  DATA(((GNDLOS(K,I,J),K=1,8),I=1,42),J=1,2)/672*0./
  DATA(ACCREW(J),J=1,2)/2*0./
  DO 999 J=1,2
  DO 999 I=1,7
  NPOP(I,J)=0
  DO 998 K=1,4
  POPORD(I,J,K)=0.
998 NPOP(I,J)=NPOP(I,J)+GRD(I,J,K)/APOP(K,J)
  IF(NPOP(I,J).EQ.0)GOTO999
  DO 997 K=1,4
997 POPORD(I,J,K)=GRD(I,J,K)/NPOP(I,J)
999 CONTINUE
  CALL OPENMS(3,KEY,41,0)
  CALL READMS(3,PK,96,35)
  CALL READMS(3,SSK,48,39)
  CALL CLOSMS(3)
 1 FORMAT(1A1)
 2 FORMAT(" ",1A1)
 3 FORMAT(" INCORRECT ENTRY--TRY AGAIN")
 4 FORMAT(" INCORRECT--MUST BE Y OR N--TRY AGAIN")
 6 PRINT*,"DO YOU WISH TO PROCESS AIR DEFENSE/ARMED HELICOPTER ASSESS
 1MENTS?"
  READ1,INX
  IF(IRUN.EQ.1)WRITE(5,1)INX
  IF(IRUN.EQ.3)PRINT2,INX
  IF(INX.EQ."Y")GOTO6
  IF(INX.EQ."N")GOTO9000
  PRINT4
  GOTO6
```

Figure K-1. OVLY5 (AHAD) program code.
(Continued next page)

```
      PRINT*,"                  AIR DEFENSE/ARMED HELICOPTER ASSESSMENTS"
      DO 1000 J=1,2
      L=1
      IF(J.EQ.L)L=2
      SH=1.-FSSF(L)*2.8)
      JPSN=1
      IF(J.EQ.IE)JPSN=2
      PA=.7
      IF(L.EQ.IE)PA=.5
      DO 900 K=59,65
      IF(ELMT(K,L).GT.0.)GOTO20
  900 CONTINUE
      GOTO500
   20 PRINT25,KIND(J),KIND(L)
   25 FORMAT(" DO YOU WISH TO GAME ",A4," ADA AND ",A4," A/C?")
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT2,INX
      IF(INX.EQ."Y")GOTO30
      IF(INX.EQ."N")GOTO1000
      PRINT4
      GOTO20
C       SET AD ENVIRONMENT
   30 PRINT32,KIND(J)
   32 FORMAT(//" THE FOLLOWING SETS PARAMETERS FOR ",A4," AD WEAPONS"/)
   37 PRINT35,KIND(J)
   35 FORMAT(" ENTER ",A4," WEAPON CONTROL (STATUS) FACTOR")
      READ*,KTRL
      IF(IRUN.EQ.1)WRITE(5,*)KTRL
      IF(IRUN.EQ.3)PRINT*,KTRL
      IF(KTRL.GE.1.AND.KTRL.LE.3)GOTO40
      IF(KTRL.EQ."T")GOTO31
      PRINT3
   31 PRINT*,"                 FOR WEAPON FREE............ENTER 1"
      PRINT*,"                     WEAPON TIGHT...........ENTER 2"
      PRINT*,"                     WEAPON HOLD............ENTER 3"
      GOTO37
   40 PRINT45,KIND(J)
   45 FORMAT(" ENTER ECM ENVIRONMENT FOR ",A4," DEPLOYED SYSTEMS.")
      READ*,IECM
      IF(IRUN.EQ.1)WRITE(5,*)IECM
      IF(IRUN.EQ.3)PRINT*,IECM
      IF(IECM.GE.1.AND.IECM.LE.2)GOTO50
      IF(IECM.EQ."T")GOTO41
      PRINT3
   41 PRINT*,"                 FOR CLEAR.................ENTER 1"
      PRINT*,"                     COUNTERMEASURES.......ENTER 2"
      GOTO40
   50 PRINT*,"ENTER NUMBER OF AVENUES OF APPROACH (MAX=5)."
      READ*,AOFA
      IF(IRUN.EQ.1)WRITE(5,*)AOFA
      IF(IRUN.EQ.3)PRINT*,AOFA
      IF(AOFA.GE.1..AND.AOFA.LE.5.)GOTO60
      PRINT3
      GOTO50
   60 PRINT65,KIND(J)
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
   65 FORMAT(" ENTER PRIORITY WEIGHTING FACTOR FOR ",A4," AOA TARGETS (M
     1AX=10).")
      READ*,WEIGHT
      IF(IRUN.EQ.1)WRITE(5,*)WEIGHT
      IF(IRUN.EQ.3)PRINT*,WEIGHT
      IF(WEIGHT.GE.1..AND.WEIGHT.LE.10.)GOTO70
      PRINT3
      GOTO50
   70 DO 990 I=31,32
      FPS(I,J)=FPS(I,J)*WEIGHT
  990 CONTINUE
C     SET FLYING TIME AND # SORTIES PER HOUR
      PRINT995,KIND(L)
  995 FORMAT(//" THE FOLLOWING SETS PARAMETERS FOR ",A4," HELICOPTERS"/)
   80 PRINT85,KIND(L),ATIME
   85 FORMAT(" ENTER TOTAL FLYING TIME FOR ",A4," A/C THIS CI (MAX= ",F4
     1.1," HOURS)"/)
      READ*,H
      IF(IRUN.EQ.1)WRITE(5,*)H
      IF(IRUN.EQ.3)PRINT*,H
      IF(H.LE.ATIME.AND.H.GE.0.)GOTO90
      PRINT3
      GOTO80
   90 PRINT95,KIND(L)
   95 FORMAT(" ENTER SORTIES PER HOUR FOR THE FOLLOWING ",A4," A/C (MAX=
     13.)")
      DO 96 K=59,65
      IF(ELMT(K,L).LE.0.)GOTO96
      KK=K-58
   97 PRINT*,"TYPE ",K,":    "
      READ*,SA(KK)
      IF(IRUN.EQ.1)WRITE(5,*)SA(KK)
      IF(IRUN.EQ.3)PRINT*,SA(KK)
      IF(SA(KK).GT.0..AND.SA(KK).LE.3.)GOTO96
      PRINT3
      GOTO97
   96 CONTINUE
C      COMPUTE # SORTIES AVAILABLE
      DO 100 K=59,65
      KK=K-58
  100 SA(KK)=SA(KK)*H*ELMT(K,L)*ACAV(KK,L)
C      BUILD CELL
      PRINT102,KIND(L),KIND(J)
  102 FORMAT(//" BEGIN BUILDING CELLS OF ",A4," A/C TO FLY AGAINST ",
     1A4," GROUND FORCES"/)
      PRINT 115,KIND(L)
  115 FORMAT(" TOTAL ",A4," A/C AND SORTIES AVAILABLE THIS CI")
      GOTO120
  130 PRINT 135,KIND(L)
  135 FORMAT(" ",A4," A/C AND SORTIES REMAINING THIS CI")
  120 PRINT*,"      A/C TYPE       # A/C       # SORTIES"
      DO 140 K=59,65
      KK=K-58
      CELL(KK)=0.
      FLYCLL(KK)=0.
      HELI=IFIX(ELMT(K,L)*ACAV(KK,L)-AIRLOS(13,KK,L))
      IF(SA(KK).LE.0..OR.HELI.LE.0.)GOTO140
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
      PRINT145,K,NCELL,SA(KK)
  145 FORMAT(9X,I2,9X,F4.0,9X,F5.0)
  140 CONTINUE
      PRINT*,"ENTER A/C ELMT #,NO. ADDED(+ OR -) TO CELL--0,0 TO STOP"
  190 READ*,K,AC
      IF(IRUN.EQ.1)WRITE(5,*)K,AC
      IF(IRUN.EQ.3)PRINT*,K,AC
      IF(K.EQ.0)GOTO200
      IF(K.LT.59.OR.K.GT.65)GOTO150
      KK=K-58
      IF(CELL(KK)+AC.LT.0.)GOTO180
      IF(CELL(KK)+AC.GT.ELMT(K,L)*ACAV(KK,L)-AIRLOS(13,KK,L))GOTO170
      IF(CELL(KK)+AC.GT.SA(KK))GOTO160
      CELL(KK)=CELL(KK)+AC
  195 PRINT*,"NEXT ENTRY"
      GOTO190
  150 PRINT*,"INVALID A/C ELMT #--ENTRY IGNORED"
      GOTO195
  160 IF(IRUN.EQ.3)GOTO161
      PRINT*,"# A/C ENTERED REQUIRES MORE SORTIES THAN ARE AVAILABLE"
      PRINT*,"ENTRY IGNORED"
      GOTO195
  170 IF(IRUN.EQ.3)GOTO171
      PRINT*,"# A/C ENTERED EXCEEDS # AVAILABLE--ENTRY IGNORED"
      GOTO195
  180 IF(IRUN.EQ.3)GOTO181
      PRINT*,"# A/C FLYING CANNOT BE NEGATIVE--ENTRY IGNORED"
      GOTO195
  161 CELL(KK)=SA(KK)
      GOTO194
  171 CELL(KK)=ELMT(K,L)*ACAV(KK,L)-AIRLOS(13,KK,L)
      IF(CELL(KK).GT.SA(KK))CELL(KK)=SA(KK)
      GOTO194
  181 CELL(KK)=0.
  194 PRINT*,"ENTRY ADJUSTED FOR TYPE ",K," A/C"
      GOTO195
  200 PRINT*,"WILL THIS CELL PENETRATE FEBA?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT2,INX
      IF(INX.EQ."Y")IL=12
      IF(INX.EQ."N")IL=0
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO210
      PRINT*
      GOTO200
  210 DO 211 I=1,32
  211 SFKILL(I)=0.
      TSTART=0.
      NPOPUP=0
      DO 215 KK=1,7
      IF(CELL(KK).GT.0..AND.NPOP(KK).GT.NPOPUP)NPOPUP=NPOP(KK)
      ACKILL(KK)=0.
      SA(KK)=SA(KK)-CELL(KK)
  215 TSTART=TSTART+CELL(KK)
      TNCR=TSTART
C      BEGIN LOOP TO FLY THE CELL
C
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
      IF(MOD(M,[illegible]).EQ.0)[illegible]=1
      DO 232 M=1,[illegible]
      FDI=0.
      DO 232 I=1,32
  232 FDI=FDI+(ELMT(I,J)*OPAV(I,J)-GNDLOS(8,I,J))*FPS(I,J)
C     AD ASSESSMENT AGAINST A/C
      IF(TNOW.LE.0.)GOTO315
      DO 240 KK=1,7
      IF((CELL(KK)-ACKILL(KK)).LE.0.)GOTO240
      AKILL=1.
      SKILL=0.
      K=KK+12
      DO 250 II=1,IN
      I=II+36
      PKILL(II)=1.
      IF(ELMT(I,J)*OPAV(I,J)-GNDLOS(9,I,J).LE.0.)GOTO250
      IF(II.EQ.9.AND.I.EQ.37)GOTO250
      S=1.-FSSF(J)*SFACT(II,J)
      EXP=(ELMT(I,J)*OPAV(I,J)-GNDLOS(8,I,J))*TMASK(ITERFN)/AOFA
      IF(I.EQ.37.OR.(I.EQ.38.AND.J.EQ.2))EXP=EXP*.7
      IF(I.EQ.37.AND.J.EQ.IA.AND.IMOUNT.EQ.2)EXP=EXP/3.
      EXP=EXP*WEAPO(KTRL)*S*.9*V(IVIS)*((CELL(KK)-ACKILL(KK))/TNOW)
      IF(SSK(II,J,IFCM)/(CELL(KK)-ACKILL(KK)).GE.1.)GOTO250
      PKILL(II)=(1.-SSK(II,J,IFCM)/(CELL(KK)-ACKILL(KK)))**EXP
      SKILL=SKILL+(1.-PKILL(II))
  250 AKILL=AKILL*PKILL(II)
      ACLOST(KK)=(1.-AKILL)*(CELL(KK)-ACKILL(KK))
      IF(ACKILL(KK)+ACLOST(KK).GT.CELL(KK))ACLOST(KK)=CELL(KK)-
     1ACKILL(KK)
C     APPORTION LOSSES
      DO 260 II=1,IN
      IF(PKILL(II).GE.1.)GOTO260
      I=II+36
      FRAC=(1.-PKILL(II))/SKILL
      AIRLOS(II,KK,L)=AIRLOS(II,KK,L)+ACLOST(KK)*FRAC
      AIRLOS(13,KK,L)=AIRLOS(13,KK,L)+ACLOST(KK)*FRAC
      ACCREW(L)=ACCREW(L)+ACLOST(KK)*CREWS(K-12,L)*FRAC
  260 CONTINUE
  240 CONTINUE
C     A/C ASSESSMENTS AGAINST GROUND FORCES
      DO 270 I=1,32
      IF(ELMT(I,J)*OPAV(I,J)-GNDLOS(8,I,J).LE.0.)GOTO270
      IF(J.EQ.IA.AND.IMOUNT.EQ.2.AND.I.LT.16)GOTO270
      IF(J.EQ.2)GOTO275
      GOTO(270,270,15,270,270,270,270,270,270,270,270,270,15,270,270,11
     1,11,11,11,12,12,12,270,270,13,13,13,270,270,270,14,270),I
  275 GOTO(270,270,15,270,270,270,270,270,270,270,270,270,15,270,270,
     111,11,11,11,12,12,11,11,270,12,12,12,11,11,270,14,14),I
   11 ITGT=1
      GOTO280
   12 ITGT=2
      GOTO280
   13 ITGT=3
      GOTO280
   14 ITGT=4
      GOTO280
   15 ITGT=5
```

Figure K-1. CVLYS (AHAD) program code (continued).

```
      GOTO280
   11 IIGT=6
  280 AHKILL=1.
      SHKILL=0.
      VICTIM=ELMT(I,J)*OPAV(I,J)-GNDLOS(8,I,J)
      VICTIM=VICTIM*PA*V(IVIS)*FSN(IENGAG,JPSN,2)
      DO 290 KK=1,7
      PHKILL(KK)=1.
      IF((CELL(KK)-ACKILL(KK)).LE.0..OR.N.GT.NPOP(KK))GOTO290
      DO 300 ITYP=1,4
      IF(POPORD(KK,L,ITYP).LE.0.)GOTO300
      PROB1=PK(INDEXS(1,ITGT,L,ITYP,J,2,6,2,J))
      PROB2=PK(INDEXS(2,ITGT,L,ITYP,0,2,6,2,J))
      IF(J.EQ.IC)PROB1=PROB1*2.
      IF(J.EQ.IA)PROB2=PROB2*2.
      PROB=(PROB1+PROB2)/3.
      IF(PROB/VICTIM.GT.1.)GOTO300
      ORDEXP=(CELL(KK)-ACKILL(KK))*POPORD(KK,L,ITYP)*SH
      ROUNDS=ORDEXP*(ELMT(I,J)*OPAV(I,J)-GNDLOS(8,I,J))*FPS(I,J)/FOF
      PHKILL(KK)=PHKILL(KK)*(1.-PROB/VICTIM)**ROUNDS
      SHOTS(ITYP+11,L)=SHOTS(ITYP+11,L)+ROUNDS
  300 CONTINUE
      AHKILL=AHKILL*PHKILL(KK)
      SHKILL=SHKILL+(1.-PHKILL(KK))
  290 CONTINUE
      VKILL=(1.-AHKILL)*VICTIM
      IF(GNDLOS(8,I,J)+VKILL.GT.ELMT(I,J)*OPAV(I,J))VKILL=ELMT(I,J)*
     1OPAV(I,J)-GNDLOS(8,I,J)
      IF(VKILL.LE..1)GOTO270
C     APPORTION LOSSES
      DO 310 KK=1,7
      IF(PHKILL(KK).GE.1.)GOTO310
      GIS=1.
      FRAC=(1.-PHKILL(KK))/SHKILL
      IEL=1
      IF(I.EQ.3)GOTO311
      GNDLOS(KK,I,J)=GNDLOS(KK,I,J)+VKILL*FRAC
      GNDLOS(8,I,J)=GNDLOS(8,I,J)+VKILL*FRAC
      GFKILL(I)=GFKILL(I)+VKILL*FRAC
      IF(I.LT.13)GOTO310
      GNDLOS(KK,2,J)=GNDLOS(KK,2,J)+VKILL*CREWS(I-12,J)*FRAC
      GNDLOS(8,2,J)=GNDLOS(8,2,J)+VKILL*CREWS(I-12,J)*FRAC
      GFKILL(2)=GFKILL(2)+VKILL*CREWS(I-12,J)*FRAC
      IF((I.NE.21.AND.I.NE.25).OR.IMOUNT.EQ.1.OR.J.EQ.ID)GOTO0310
      GIS=5.
      IEL=3
  311 DO 312 IWP=IEL,12
      GNDLOS(KK,IWP,J)=GNDLOS(KK,IWP,J)+VKILL*GIS*PLT(IWP)*FRAC
      GNDLOS(8,IWP,J)=GNDLOS(8,IWP,J)+VKILL*GIS*PLT(IWP)*FRAC
      GFKILL(IWP)=GFKILL(IWP)+VKILL*GIS*PLT(IWP)*FRAC
  312 CONTINUE
  310 CONTINUE
  270 CONTINUE
      TNEW=0.
      DO 315 KK=1,7
      ACKILL(KK)=ACKILL(KK)+ACLOST(KK)
      ACLOST(KK)=0.
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
311 TNON=TNON+(CELL(KK)*ACKILL(KK))
C    CHECK A/C LOSSES FOR ABORT
315 NN=N
    IABORT=0
    IF(TNON.GE.TSTART*.7.AND.NN.NE.NPOPUP)GOTO230
    IF(NN.EQ.NPOPUP)GOTO51.
    PRINT*,"LOSSES EXCEED 30% AFTER ",NN," POPUPS"
    IF(IRUN.NE.2)GOTO..3
    IABORT=1
    GOTO=1
50  PRINT*,"SORTIE ABORTED"
    GOTO515
51  PRINT*,"SORTIE COMPLETED"
515 PRINT*,"DO YOU WISH TO SEE LOSSES?"
    READ1,INX
    IF(IRUN.EQ.1)WRITE(5,1)INX
    IF(IRUN.EQ.3)PRINT2,INX
    IF(INX.EQ."Y")GOTO520
    IF(INX.EQ."N")GOTO349
    PRINT4
    GOTO515
520 PRINT325,KIND(L)
325 FORMAT(//" ",A4," HELICOPTERS KILLED"/"   TYPE        # KILLED")
    DO 326 KK=1,7
    IF(ACKILL(KK).LE..1)GOTO326
    K=KK+5
    PRINT327,K,ACKILL(KK)
327 FORMAT(" ",3X,I2,10X,F5.1)
326 CONTINUE
    PRINT328,KIND(J)
328 FORMAT(//" ",A4," GROUND FORCES KILLED"/"   TYPE        # KILLED")
    DO 329 I=16,32
    IF(GFKILL(I).LE..1)GOTO329
    PRINT331,I,GFKILL(I)
331 FORMAT(" ",3X,I2,11X,F5.1)
329 CONTINUE
349 IF(IABORT.EQ..)GOTO350
229 PRINT*,"DO YOU WISH TO ABORT THIS SORTIE?"
    READ1,INX
    IF(INX.EQ."Y")GOTO350
    IF(INX.EQ."N")GOTO230
    PRINT4
    GOTO229
230 CONTINUE
350 PRINT355,KIND(L)
355 FORMAT(" DO YOU WISH TO FLY ANOTHER CELL OF ",A4," A/C?")
    READ1,INX
    IF(IRUN.EQ.1)WRITE(5,1)INX
    IF(IRUN.EQ.3)PRINT2,INX
    IF(INX.EQ."Y")GOTO130
    IF(INX.EQ."N")GOTO1001
    PRINT4
    GOTO350
1001 DO 1002 I=31,32
1002 FPS(I,J)=FPS(I,J)/WEIGHT
    DO 1003 I=1,15
    IF(GS.LD3(5,I,J).LE.ELMT(I,J)*DPAV(I,J))GOTO1003
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
      IF(GNDLOS(8,I,J).LT..1)GOTO1003
      DO 1004 KK=1,7
1004  GNDLOS(KK,I,J)=(GNDLOS(KK,I,J)/GNDLOS(8,I,J))*ELMT(I,J)*OPAV(I,J)
      GNDLOS(8,I,J)=ELMT(I,J)*OPAV(I,J)
1003  CONTINUE
      DO 1005 I=1,42
      DO 1005 K=59,65
      KK=K-58
      IF(I.GT.32)GOTO1006
      GNDLOS(KK,I,J)=IFIX(GNDLOS(KK,I,J)*10.+.5)/10.
      ALOSS(K,1)=ALOSS(K,1)+IFIX(GNDLOS(KK,I,J)*10.+.001)*PACK(J)
      IF(I.LT.31)GOTO1005
1006  II=I-31
      AIRLOS(II,KK,L)=IFIX(AIRLOS(II,KK,L)*10.+.5)/10.
      ALOSS(I,K)=ALOSS(I,K)+IFIX(AIRLOS(II,KK,L)*10.+.001)*PACK(L)
      ALOSS(I,2)=ALOSS(I,2)+IFIX(AIRLOS(II,KK,L)*CREWS(K-12,L)*10.+.001)
     1*PACK(L)
1005  CONTINUE
      IF(ACCREW(L).GT.ELMT(2,L))ACCREW(L)=ELMT(2,L)
      GOTO1500
 500  PRINT555,KIND(L)
 555  FORMAT(" THE ",A4," FORCE HAS NO HELICOPTERS")
1000  CONTINUE
      IF(I-UN.EQ.1)GOTO3000
      PRINT1010
1010  FORMAT("1")
      PRINT*,"*--------------ARMED HELICOPTER ASSESSMENTS---------------*"
      PRINT1020
1020  FORMAT(" I",65X,"I")
      DO 1220 L=1,2
      IFLAG=0
      J=1
      IF(J.EQ.L)J=2
      DO 2000 I=1,32
      CLOST=0.
      IF(I.LT.16.AND.I.NE.13)GOTO2010
      CLOST=GNDLOS(8,I,J)*CREWS(I-12,J)
2010  AKILL=IFIX(GNDLOS(8,I,J)*10.+.5)/10.
      IF(AKILL.LT..1.AND.CLOST.LT..1)GOTO2000
      IF(IFLAG.GT.0)GOTO2020
      IFLAG=1
      IF(J.EQ.2)GOTO2030
      PRINT*,"I                             TOTAL BLUE LOSSES                     I"
      GOTO2040
2030  PRINT*,"I                              TOTAL RED LOSSES                     I"
2040  PRINT*,"I                      ITEM      # LOST     CREW LOST               I"
      PRINT1020
2020  IF(I.LT.16.AND.I.NE.13)GOTO2050
      PRINT2025,I,AKILL,CLOST
2025  FORMAT(" I",15X,I2,5X,F6.1,5X,F5.1,15X,"I")
      GOTO2000
2050  PRINT2055,I,AKILL
2055  FORMAT(" I",15X,I2,5X,F6.1,27X,"I")
2000  CONTINUE
      IF(IFLAG.EQ.0)PRINT1020
      IF(L.EQ.2)GOTO1030
      PRINT*,"*---------------------------------------------------------*"
```

Figure K-1. OVLY5 (AHAD) program code (continued).

```
      PRINT1020
 1030 CONTINUE
      PRINT*,"*============================================================*"
 3000 IF(IRUN.EQ.1)PRINT*,"ARMED HELICOPTER ASSESSMENTS PRINTED HERE"
      PRINT1015
 1015 FORMAT(////" FOR GROUND FORCES KILLED BY HELICOPTERS:")
      CALL LOSS(59,65,1,32)
      IF(IRUN.EQ.1)GOTO5000
      PRINT1010
      PRINT*,"*----------------AIR DEFENSE ASSESSMENTS----------------*"
      PRINT1020
      DO 4000 J=1,2
      L=1
      IF(L.EQ.J)L=2
      JFLAG=0
      IFLAG=0
      DO 4010 K=59,65
      IF(ACCREW(L).GE..5.AND.IFLAG.EQ.0)GOTO4020
 4070 CLOST=AIRLOS(13,K-58,L)*CREWS(K-12,L)
      AKILL=IFIX(AIRLOS(13,K-58,L)*10.+.5)/10.
      JFLAG=1
      IF(CLOST.LT..1.AND.AKILL.LT..1)GOTO4010
      IF(IFLAG.GT.0)GOTO4030
 4020 IFLAG=1
      IF(J.EQ.2)GOTO4040
      PRINT*,"I                        TOTAL RED LOSSES                       I"
      GOTO4050
 4040 PRINT*,"I                       TOTAL BLUE LOSSES                       I"
 4050 PRINT*,"I               ITEM     # LOST     CREW KILLED                 I"
      PRINT1020
      IF(JFLAG.NE.1)GOTO4060
 4030 PRINT4035,K,AKILL,CLOST
 4035 FORMAT(" I",15X,I2,5X,F6.1,5X,F5.1,15X,"I")
      GOTO4010
 4060 AKILL=ACCREW(L)
      PRINT4065,AKILL
 4065 FORMAT(" I",15X,"2",5X,F6.1,27X,"I")
      GOTO4070
 4010 CONTINUE
      IF(IFLAG.NE.0)PRINT1020
      IF(J.EQ.2)GOTO4000
      PRINT*,"*-----------------------------------------------------------*"
      PRINT1020
 4000 CONTINUE
      PRINT*,"*===========================================================*"
 5000 IF(IRUN.EQ.1)PRINT*,"AIR DEFENSE ASSESSMENTS PRINTED HERE"
      PRINT5005
 5005 FORMAT(////" FOR HELICOPTERS KILLED BY AIR DEFENSE:")
      CALL LOSS(31,42,59,65)
      PRINT1010
 9000 END
```

Figure K-1. OVLY5 (AHAD) program code (concluded).

APPENDIX L

OVLY 6 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX L

OVLY 6 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN source code listings and variable lists for the main program, CANNON, and subroutine, CLGP, of the OVLY 6 overlay. OVLY 6 is the routine that assesses indirect fire combat losses. CANNON contains the logic for assessing all true indirect fire missions; subroutine CLGP assesses only cannon launched guided projectile (CLGP) missions. Table L-1 is the program variable list for CANNON; table L-2 is the list for CLGP. The FORTRAN source code is given in figure L-1 for CANNON and figure L-2 for CLGP.

Table L-1. Program variables for CANNON (Continued next page).

Variable	Description
ACQ	Acquisition factor
ADSF	Air defense suppression mission flag
AKILL	Target survival probability against all firers
AT	Number of homogeneous area targets.
BMT	Battery missions per tube
CBTLEV	IDF combat level required
CLEV	IDF combat level
CLOST	Crewmen lost
CM	Total CLGP missions fired
CM50	CLGP missions fired by weapon 50
CM53	CLGP missions fired by weapon 53
DE	Fraction of IDF systems deployed
F	Fraction of missions which are targeted
FAC	Fire allocation constant
FDF	Fire distribution factor
FDT	Fractional damage table
FLAG	Flag for type of IDF mission
FPF	Length of final protective fires (minutes)
HOURS	Length of IDF mission (hours)
HR	Length of IDF support (hours)

Table L-1. Program variables for CANNON (continued)

Variable	Description
HRARTY	Length of artillery support (hours)
I	Firer weapon integer index
ICAT	IDF weapon category index
ICB	Counterbattery mission flag
ICS	Close support mission flag
IFLAG	Mission flag index
II	Firer weapon mapping index
INX	Input response variable
IOP	Suppression factor index
IPOINT	Output header flag
IS	AMMO array index
ISHOT	Ammunition expenditure index
ITP	Dual purpose ICM flag
I50	Weapon 50 CLGP fire flag
I53	Weapon 53 CLGP fire flag
J	Firer's force integer index
K	Victim weapon integer index
KIND	Force color
KK	Target mapping index
L	Victim's force integer index

Table L-1. Program variables for CANNON (concluded).

Variable	Description
MAP	IDF target mapping array
MW	Military worth array
MWTH	Military worth
OPERA	Operational availability
PERSF	Personnel fire missions flag
PKILL	Target survival probability against firer
PLOSS	Total victims killed
PDK	Percent of knowledge
PREP	Lengths of prep/counter-prep fires (minutes)
ROF	Rate of fire
RPM	Rounds per mission
S	Suppression factor
SKILL	Loss apportionment factor denominator
SUPR	Weapon suppression constants
TBAT	Tubes per battery
TGT	Elements per target
TKILL	Target losses to IDF systems

NOTE: All COMMON variables are defined in table F-1.

```
      OVERLAY(CANNON,6,0)
      PROGRAM OVLY5
      COMMON IA,ID,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,CFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PACK(2),
     3 ELMT(90,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PSN(6,2,2),PLT(15),KEY(41)
      DIMENSION TBAT(13,2),SUPR(4),ISHOT(13,2),FDF(5,2),PREP(2),MAP(55),
     1 PLOSS(55,2,2),HR(2),PERSF(2),ROF(13,3,2),RPM(2),PKILL(13),
     2 CUTLEV(6),TGT(17,2),OPERA(17,2),POK(55,2),FDT(15,17,2),MW(17),
     3 ILB(2),ICS(2),CLEV(2),ADSF(2)
      REAL MW,MWTH
C
C          # OF ELEMENTS PER TARGET.
      DATA((TGT(K,L),K=1,17),L=1,2)/49.,4*10.,3.,1.,6.,10.,3*6.,49.,2
     1.,3*10.,31.,4*10.,3.,1.,6.,10.,4.,2*6.,31.,2.,3*10./
C
C          DATA MAPPING INDEX.
      DATA(MAP(I),I=1,55)/6,0,1,6*0,2*2,2*1,2*4,3,4,11*5,9,8,4*8,1,8,8,3
     1*7,3*10,5,5,2*11,6*12/
C
C          INDIRECT FIRE WEAPON SUPPRESSION COFFICIENTS.
      DATA(SUPR(I),I=1,4)/3.52,3*2.86/
C
C          # IDF TUBES PER BATTERY
      DATA((TBAT(I,J),I=1,13),J=1,2)/2*3.,2*4.,3.,6.,0.,6.,2*4.,6.,6.,
     10.,3*6.,2*0.,3*6.,6.,4*6./
C
C          IDF ROUND EXPENDITURE INDEX
      DATA((ISHOT(I,J),I=1,13),J=1,2)/17,19,2*21,19,23,0,26,31,32,26,
     135,0,17,19,21,2*0,23,26,23,26,25,27,30,31/
C
      CALL OPENMS(3,KEY,41,0)
C          POK TABLE
      CALL READMS(3,POK,110,36)
C
C          IDF FRACTIONAL DAMAGE TABLE.
      CALL READMS(3,FDT,510,37)
C
C          FRACTION OF ARTY PER SUPPORT LEVEL.
      DATA(CUTLEV(I),I=1,6)/.35,.67,1.,1.67,2.5,4./
C
C          IDF SYSTEM OPERATIONAL AVAILABILITIES.
      DATA((OPERA(I,J),I=1,17),J=1,2)/1.,.93,.67,.72,.74,.9,.83,2*.6,
     1 .92,.86,.7,1.,.93,2*.72,.74,1.,.91,2*.7,.81,.9,.83,2*.85,2*.86,
     2 .7,1.,.91,2*.7,.81/
C
C          IDF RATES OF FIRE.
      CALL READMS(3,ROF,78,38)
      CALL CLOSMS(3)
C
C          MILITARY WORTH.
      DATA(MW(I),I=1,17)/8.36,5.47,2*12.86,10.79,2.56,2*4.05,10.79,6.71,
     1 2*10.12,8.36,5.47,2*12.86,10.79/
C
   10 PRINT*,"DO YOU WISH TO PROCESS INDIRECT FIRE ASSESSMENTS?"
```

Figure L-1. CANNON program code. (Continued next page.)

```
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
    1 FORMAT(1A1)
    8 FORMAT(" ",1A1)
      IF(INX.EQ."Y")GOTO15
      IF(INX.EQ."N")GOTO1000
      PRINT3
    3 FORMAT(" INCORRECT ENTRY - TRY AGAIN")
      GOTO10
C
C
   15 PRINT*,"              INDIRECT FIRE ASSESSMENTS"
C
C                             *** INITIALIZATION ***
C
      FLAG=0.
      CM30=0.
      CM53=0.
C
      DO 18 J=1,2
   18 PLASF(J)=0.
C
      DO 20 K=1,15
      DO 20 J=1,2
      DO 20 L=1,2
   20 PLOSS(K,J,L)=0.
C
C         SET AVG. ROUND PER MISSION.
      RPM(1)=6.
      RPM(2)=6.
C
C                          - INTERACTIVE INPUTS -
C
   25 PRINT*,"IS DUAL PURPOSE ICM BEING USED?"
      READ1,ITP
      IF(IRUN.EQ.1)WRITE(5,1)ITP
      IF(IRUN.EQ.3)PRINT8,ITP
      IF(ITP.EQ."Y".OR.ITP.EQ."N")GOTO30
      PRINT3
      GOTO25
C
   30 DO 38 J=1,2
      KIND="BLUE"
      IF(J.EQ.2)KIND="REL"
      PRINT33,KIND
   33 FORMAT(" ENTER LEVEL OF ",A4," ARTY SUPPORT -")
      READ*,INX
      IF(IRUN.EQ.1)WRITE(5,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.GE.1.AND.INX.LE.6)GOTO38
      IF(INX.EQ."T")GOTO35
      PRINT3
   35 PRINT*,"ENTER 1 FOR LIGHT INTERMITTENT FIRES"
      PRINT*,"      2 FOR FIRES BASED ON 2/3 BASIC LOAD"
      PRINT*,"      3 FOR FIRES BASED ON TOTAL BASIC LOAD"
      PRINT*,"      4 FOR FIRES BASED ON 2/3 DAILY RESUPPLY RATE"
```

Figure L-1. CANNON program code (continued).

```
      PRINT*,"        2 FOR FIRES BASED ON TOTAL DAILY RESUPPLY RATE"
      PRINT*,"        3 FOR APPROX. SUSTAINED RATE OF FIRE"
      GOTO30
C
C         SET ARTY COMBAT LEVEL.
   35 CLEV(J)=SETLEV(INX)
C
   40 PRINT*,"ENTER # HOURS OF ARTY SUPPORT (0-",ATIME,")"
      READ*,HRARTY
      IF(IRUN.EQ.1)WRITE(5,*)HRARTY
      IF(IRUN.EQ.3)PRINT*,HRARTY
      IF(HRARTY.GE.0.AND.HRARTY.LE.ATIME)GOTO50
      PRINT3
      GOTO40
C
   50 PRINT*,"ENTER # MINUTES OF PREP FIRE (0-60)"
      READ*,PREP(IA)
      IF(IRUN.EQ.1)WRITE(5,*)PREP(IA)
      IF(IRUN.EQ.3)PRINT*,PREP(IA)
      IF(PREP(IA).EQ.0.)GOTO70
      IF(PREP(IA).GT.0..AND.PREP(IA).LE.60.)GOTO60
      PRINT3
      GOTO50
C
   60 PRINT*,"ENTER # MINUTES OF COUNTER-PREP FIRES (0-60)"
      READ*,PREP(ID)
      IF(IRUN.EQ.1)WRITE(5,*)PREP(ID)
      IF(IRUN.EQ.3)PRINT*,PREP(ID)
      IF(PREP(ID).GE.0..AND.PREP(ID).LE.60.)GOTO70
      PRINT3
      GOTO60
C
   70 PRINT*,"ENTER # MINUTES OF FINAL PROTECTIVE FIRE (0-60)"
      READ*,FPF
      IF(IRUN.EQ.1)WRITE(5,*)FPF
      IF(IRUN.EQ.3)PRINT*,FPF
      IF(FPF.GE.0..AND.FPF.LE.60.)GOTO80
      PRINT3
      GOTO70
C
C         CALC. ACTUAL # HOURS OF IDF SUPPORT
C         ATTACKING FORCE
   80 HOURS=HRARTY
      HR(IA)=HOURS-(PREP(IA)/60.)
C         DEFENDING FORCE
      HR(ID)=HOURS-(PREP(ID)/60.)-(FPF/60.)
C
C
C         SPECIAL MISSION LOGIC FLAGS
C
      PRASF(IA)=1.
      IF(IMOUNT.EQ.2)GOTO90
      PRASF(ID)=1.
      GOTO91
C
   90 PRINT*,"WILL ATTACKER DISMOUNT INFANTRY DURING THIS CI?"
      READ*,INX
```

Figure L-1. CANNON program code (continued).

```
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")PERSF(IF)=1./HR(I)
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO91
      PRINT3
      GOTO90
   91 DO 95 J=1,2
      KIND="BLUE"
      IF(J.EQ.2)KIND="RED"
      PRINT92,KIND
   92 FORMAT(" SPECIFY THE TYPES OF IDF MISSIONS THE ",A4," FORCE WILL F
     CIRE")
   96 PRINT*,"COUNTER-BATTERY?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      ICB(J)=1
      IF(INX.EQ."N")ICB(J)=0
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO93
      PRINT2
    2 FORMAT(" INCORRECT ENTRY - RESPONSE MUST BE Y OR N")
      GOTO96
   93 PRINT*,"CLOSE SUPPORT?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      ICS(J)=1
      IF(INX.EQ."N")ICS(J)=0
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO94
      PRINT2
      GOTO93
   94 PRINT*,"AD SUPPRESSION?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      ADSF(J)=1.
      IF(INX.EQ."N")ADSF(J)=0.
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO95
      PRINT2
      GOTO94
   95 CONTINUE
C
C                       *** FIRE DISTRIBUTION FACTOR (FDF) ***
C
  100 DO 99 INX=1,2
      DO 99 J=1,2
   99 FDF(INX,J)=0.
C
      IFLAG=IFLAG+1
C         CYCLE ROUTINE FOR BOTH FORCES.
C
      DO 250 J=1,2
      L=1
      IF(L.EQ.J)L=2
      IF(IFLAG.EQ.2.AND.J.EQ.IA)GOTO250
C
C          BRANCH FOR FP FIRES
```

Figure L-1. CANNON program code (continued).

```
      IF(FLAG.EQ.2)GOTO110
      IF(FLAG.EQ.3..AND.PERF(J).EQ.0.)GOTO120
C
C          FOR DENOMINATOR FOR STANDARD IDF MISSIONS.
C
C          ITERATE FOR ALL POSSIBLE IDF TARGETS.
      DO 106 K=1,59
      IF(ELMT(K,L)-PLOSS(K,L,2).LE.0.)GOTO106
C           TEST FOR DISMOUNTED INFANTRY DURING PREP/C-PREP FIRES.
      IF(K.EQ.3.AND.FLAG.EQ.3..AND.PERSF(J).NE.1.)GOTO106
C
C          SET ACQ FACTOR.
      KK=2
      INX=2
      IF(L.EQ.IA)INX=1
      ACQ=POK(K,L)
      IF((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47))ACQ=POK(K,L)*
     1 PSN(IENGAG,INX,KK)+.5*(1.-PSN(IENGAG,INX,KK))
C
C          CALCULATE PRESENTED TARGET AREAS.
      FAC=1.
      IF(K.EQ.3)FAC=PERSF(J)
      IF(K.GE.33.AND.K.LE.42)FAC=ADSF(J)
      IF(K.LT.33.AND.ICS(J).EQ.0)GOTO106
      IF(K.GE.43.AND.K.LE.59.AND.ICB(J).EQ.0)GOTO106
      KK=MAP(K)
      IF(KK.LE.5.AND.L.EQ.ID)KK=KK+12
      AT=ACQ*(ELMT(K,L)-PLOSS(K,L,2))*OPERA(KK,L)/TGT(KK,L)
      MWTH=MW(KK)
      IF(J.EQ.2.OR.FLAG.EQ.3.)MWTH=1.
      FAC=AT*MWTH*FAC
C
C          FILTER OUT INAPPROPRIATE SPECIAL MISSIONS
      IF(FAC.LE.0)GOTO106
C
C          BRANCH AS PER TARGETING SCHEME
C
      IF(K.EQ.1.OR.K.EQ.48.OR.K.EQ.59)GOTO103
      IF(K.EQ.3.OR.K.EQ.12.OR.K.EQ.13)GOTO101
      IF(K.LT.16.OR.K.EQ.37.OR.K.EQ.42)GOTO106
      IF(K.GE.25.AND.K.LE.36)GOTO102
      IF(K.GE.40.AND.K.LE.41)GOTO105
      IF(K.GE.43.AND.K.LE.47)GOTO101
      GOTO104
C
C          LT. MORTARS
  101 FOF(1,J)=FOF(1,J)+FAC
C
C          HVY. MORTARS
  102 FOF(2,J)=FOF(2,J)+FAC
C
C          LT. ARTY
  103 FOF(3,J)=FOF(3,J)+FAC
C
C          MED. ARTY
  104 FOF(4,J)=FOF(4,J)+FAC
C
```

Figure L-1. CANNON program code (continued).

```
C          HVY. ARTY
  1.5 IF(K.GE.46.AND.K.LE.55)FAC=FAC*2.
      FOF(L,J)=FOF(L,J)+FAC
  106 CONTINUE
      GOTO120
C
C
C          FOF DETERMINATOR CALC. FOR FP FIRES.
C
C          TEST FOR SPECIAL IDF MISSIONS
  110 IF(FLAG.EQ.2..AND.(J.EQ.IA.OR.FPF.EQ.0.))GOTO120
C
C          ONLY FORWARD WEAPON SYSTEMS ARE SPECIAL MISSION TARGETS.
      DO 115 K=1,32
      IF(ELMT(K,L)-PLOSS(K,L,2).LE.0.)GOTO115
C
C          SET ACQ FACTOR.
      KK=2
      IF(FLAG.EQ.1.)KK=1
      INX=2
      IF(L.EQ.IA)INX=1
      ACQ=POK(K,L)
      IF((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47))ACQ=POK(K,L)*
     1 PSN(IENGAG,INX,KK)+.5*(1.-PSN(IENGAG,INX,KK))
C          CALCULATE PRESENTED TARGET AREAS.
C
C
      FAC=1.
      IF(K.EQ.3)FAC=FLOOF(J)
      KK=MAF(K)
      IF(KK.LE.5.AND.L.EQ.IE)KK=KK+12
      A1=ACQ*(ELMT(K,L)-PLOSS(K,L,2))*OPERA(KK,L)/TGT(KK,L)
      FAC=A1*FAC
C          FILTER OUT INAPPROPRIATE SPECIAL MISSIONS.
      IF(FAC.LE.0.)GOTO115
C
C          FILTER OUT INAPPROPRIATE SPECIAL MISSION TARGETS.
      IF(K.LT.16.OR.(K.GT.19.AND.K.NE.21.AND.K.NE.25))GOTO115
      DO 111 INX=1,5
      FOF(INX,J)=FOF(INX,J)+FAC
  111 CONTINUE
  115 CONTINUE
C
C
C                        *** IDF ASSESSMENTS ***
C
C
C          ITERATE FOR ALL IDF TARGETS.
  120 INJ=0
      INP=0
      DO 200 K=1,3
      SKILL=.
      SKILL=1.
      IF(ELMT(K,L)-PLOSS(K,L,2).LE.0.)GOTO200
C
C          SET ACQ PARMS.
      KK=2
```

Figure L-1. CANNON program code (continued).

```
      IF(FLAG.EQ.2.)KK=1
C
C          DETERMINE AGO FACTOR.
      INX=2
      IF(L.EQ.IA)INX=1
      AGO=POK(K,L)
      IF((K.GE.3.AND.K.LE.32).OR.(K.GE.43.AND.K.LE.47))AGO=POK(K,L)*
     1 PSN(IENGAG,INX,KK)+.5*(1.-PSN(IENGAG,INX,KK))
C          CALCULATE PRESENTED TARGET AREAS.
      FAC=1.
      IF(K.LT.33)FAC=TOS(J)
      IF(K.EQ.3.AND.FAC.EQ.1)FAC=PERSF(J)
      IF(K.GE.33.AND.K.LE.42)FAC=AOSF(J)
      IF(K.GE.43.AND.K.LE.55)FAC=ICB(J)
      KK=MAP(K)
      IF(KK.LE.5.AND.L.EQ.IE)KK=KK+12
      AT=AGO*(ELMT(K,L)-PLOSS(K,L,2))*OPERA(KK,L)/TGT(KK,L)
C          SET MILITARY WORTH OF TARGETS
      MWTH=MW(KK)
      IF(J.EQ.2.OR.FLAG.NE.1.)MWTH=1.
      IF(K.GE.48.AND.K.LE.55.AND.ICAT.EQ.5)MWTH=MWTH*2.
C          ITERATE FOR ALL IDF WEAPON SYSTEMS.
      DO 220 I=43,55
      PKILL(I-42)=1.
      IF(ELMT(I,J)-PLOSS(I,J,2).LE.0.)GOTO220
C
C          DETERMINE CATEGORY OF IDF WEAPON
      ICAT=4
      IF(I.EQ.43)ICAT=1
      IF(I.GE.44.AND.I.LE.47)ICAT=2
      IF(I.EQ.48.OR.I.EQ.55)ICAT=3
      IF(I.EQ.52.OR.(I.EQ.53.AND.J.EQ.2))ICAT=5
      IF(FDF(ICAT,J).EQ.0.)GOTO220
C
C          SET # HOURS OF IDF SUPPORT.
      HOURS=HR(J)
      IF(FLAG.EQ.0.)HOURS=PREF(J)/60.
      IF(FLAG.EQ.2.)HOURS=FPF/60.
      IF(FLAG.EQ.2..AND.J.EQ.IA)HOURS=0.
      IF(HOURS.LT.0)HOURS=0.
C          SET FRACTION OF MISSIONS FIRED AT TARGETED OBJECTIVES.
      F=.87
      IF(ICAT.LT.3)F=.47
C
C          CALC. SUPPRESSION FACTOR
      IOP=1
      IF(J.EQ.1.AND.I.GE.46)IOP=2
      S=1.-FSSF(J)*SUPR(IOP)
      IF(FLAG.EQ.2)S=1.
C
C          SET THE FRACTION OF MORTARS ACTIVE.
      OF=PSN(IENGAG,J,2)
      IF(FLAG.EQ.0.)OF=PSN(IENGAG,J,1)
C
C          ALL ARTY IS ACTIVE.
      IF(ICAT.GT.2)OF=1.
      IF(ICAT.GT.2.AND.FLAG.EQ.1.)OF=OLEV(J)
```

Figure L-1. CANNON program code (continued).

Best Available Copy

```
C
C       SET UPPING INDEX
      II=MAP(I)
C       BRANCH FOR PREP/COUNTER-PREP OR FP FIRES.
      IF(FLAG.EQ.2)GOTO150
C
C       FILTER OUT INAPPROPRIATE TARGETS FOR STANDARD IDF MISSIONS.
      IF(K.EQ.1.OR.K.EQ.48.OR.K.EQ.53)GOTO123
      IF(K.EQ.3.OR.K.EQ.12.OR.K.EQ.13)GOTO121
      IF(K.LT.16.OR.K.EQ.37.OR.K.EQ.42)GOTO200
      IF(K.GE.29.AND.K.LE.36)GOTO122
      IF(K.GE.40.AND.K.LE.41)GOTO125
      IF(K.GE.43.AND.K.LE.47)GOTO121
      GOTO124
C
  121 IF(ICAT.EQ.1)GOTO170
C
  122 IF(ICAT.EQ.2)GOTO170
C
  123 IF(ICAT.EQ.3)GOTO170
C
  124 IF(ICAT.EQ.4)GOTO170
C
  125 IF(ICAT.EQ.5)GOTO170
      GOTO225
C
C       FILTER OUT INAPPROPRIATE TARGETS FOR SPECIAL IDF MISSIONS
  150 IF(K.GT.32)GOTO250
      IF(K.EQ.3.AND.FLAG.EQ.0..AND.PERSF(J).NE.1.)GOTO200
      IF(K.LT.16.AND.K.NE.3.AND.K.NE.12.AND.K.NE.13)GOTO200
      IF(FLAG.EQ.2.AND.(K.LT.16.OR.(K.GT.19.AND.K.NE.23.AND.K.NE.25)))GO
     1TO200
C
C       CALCULATE BATTERY MISSIONS PER TUBE FOR THIS TYPE MISSION.
  170 IF(TBAT(I-42,J).LE.0.)GOTO225
      BMT=(CLMT(I,J)-PLUSS(I,J,2))/TBAT(I-42,J)*OPERA(II,J)*S*DE*F*HOURS
     1 *FDF(I-42,IFLAG,J)/TFM(J)*FAC*AT*MNTH/FDF(ICAT,J)
C
      IF(FLAG.NE.1..OR.J.EQ.2)GOTO175
      IF(I.NE.50.AND.I.NE.53)GOTO175
      IF(I.EQ.50.AND.I50.EQ.1)GOTO175
      IF(I.EQ.53.AND.I53.EQ.1)GOTO175
      IF(FAC*AT*FDF(ICAT,J).LE.0)GOTO174
      BMT=BMT*FDF(ICAT,J)/(FAC*AT*MNTH)
      CALL CLGP(BMT,I,PLUSS,CM)
      BMT=BMT*FAC*AT*MNTH/FDF(ICAT,J)
      IF(I.EQ.50)I50=1
      IF(I.EQ.50)CM50=CM
      IF(I.EQ.53)I53=1
      IF(I.EQ.53)CM53=CM
C
C       CALC. AMMO EXPENDITURES
  175 IF(J.EQ.2.OR.FLAG.NE.1.OR.(I.NE.50.AND.I.NE.53))GOTO174
      IF(I.EQ.50)CM=CM50
      IF(I.EQ.53)CM=CM53
      BMT=BMT-(CM/3.*FAC*AT*MNTH/FDF(ICAT,J))
  174 IT=ITBAT(I-42,J)
```

Figure L-1. CANNON program code (continued).

```
      IF(J.EQ.2)GOTO169
      IF(ICAT.GT.2)SHOTS(IS,J)=SHOTS(IS,J)+6.*BMT*.1/F*TBAT(I-42,J)
      IF(J.EQ.1.AND.ITP.EQ."Y".AND.(I.EQ.50.OR.I.EQ.51.OR.I.EQ.53))GOTO1
     C76
      GOTO177
  176 IS=IS+3
      GOTO178
  177 IF(K.EQ.3.AND.ICAT.GT.2)IS=IS+2
  178 SHOTS(IS,J)=SHOTS(IS,J)+6.*BMT*TBAT(I-42,J)
      GOTO183
C
  180 IF(I.NE.48)GOTO181
      SHOTS(IS+1,J)=SHOTS(IS+1,J)+6.*BMT*(1.-F)/F*TBAT(I-42,J)
      GOTO182
  181 SHOTS(IS,J)=SHOTS(IS,J)+6.*BMT*(1.-F)/F*TBAT(I-42,J)
  182 SHOTS(IS,J)=SHOTS(IS,J)+6.*BMT*TBAT(I-42,J)
C
C          CALC. LOSSES
  183 IF(FOT(I-42,KK,J)/AT.GT.1.)GOTO200
      IF(J.EQ.1.AND.ITP.EQ."Y".AND.(I.EQ.50.OR.I.EQ.51.OR.I.EQ.53))
     GOGOTO225
      IF(FOT(INX,KK,J)/AT.GE.1.)GOTO220
      PKILL(I-42)=(1.-FOT(I-42,KK,J)/AT)**BMT
      GOTO222
  225 INX=14
      IF(I.EQ.51)INX=15
      IF(FOT(INX,KK,J)/AT.GE.1.)GOTO220
      PKILL(I-42)=(1.-FOT(INX,KK,J)/AT)**BMT
  222 AKILL=AKILL*PKILL(I-42)
      SKILL=SKILL+(1.-PKILL(I-42))
  220 CONTINUE
      TKILL=(1.-AKILL)*AT*TGT(KK,L)
C
C          DISTRIBUTE LOSSES
      IF(SKILL.LE.0.)GOTO260
      DO 205 I=43,55
      IF(ELMT(I,J)-FLOSS(I,J,2).LE.0.)GOTO205
      AKILL=TKILL*(1.-PKILL(I-42))/SKILL
      AKILL=IFIX(AKILL*10.+.5)/10.
C
C          BRANCH FOR PERSONNEL TARGETS
      IF(K.EQ.3)GOTO190
      ALOSS(I,K)=ALOSS(I,K)+IFIX(AKILL*10.+.001)*PACK(L)
      FLOSS(K,L,1)=FLOSS(K,L,1)+IFIX(AKILL*10.+.001)/10.
C
C          ACCESS CREW KILLS.
      IF(K.GT.12)ALOSS(I,2)=ALOSS(I,2)+IFIX(AKILL*CREWS(K-12,L)*10.+.001
     1 )*PACK(L)
      IF(K.GT.12)FLOSS(2,L,1)=FLOSS(2,L,1)+IFIX(AKILL*CREWS(K-12,L)*10.
     C+.001)/10.
      IF((K.GE.21.AND.K.LE.25).OR.IMOUNT.EQ.1)GOTO205
      IF(L.EQ.IC)GOTO205
      AKILL=IFIX(AKILL*10.+.5)/10.
C
C          ASSESSMENT OF DISMOUNTED INFANTRY WEAPONS.
  190 DO 191 KK=3,11
      IF(ELMT(KK,L)-FLOSS(KK,L,2).LE.0.)GOTO191
```

Figure L-1. CANNON program code (continued).

```
      ALOSS(I,KK)=ALOSS(I,KK)+IFIX(AKILL*PLT(KK)*10.+.001)*PACK(L)
      PLOSS(KK,L,1)=PLOSS(KK,L,1)+IFIX(AKILL*PLT(KK)*10.+.001)/10.
  191 CONTINUE
C
  200 CONTINUE
C
  220 CONTINUE
C
  250 CONTINUE
C
      IF(IRUN.EQ.1)GOTO230
      PRINT303
      IF(IFLAG-2)260,270,280
  260 PRINT*,"*----------------PREP/C-PREP ASSESSMENTS----------------*"
      GOTO290
  270 PRINT*,"*------------STANDARD IOF MISSION ASSESSMENTS------------*"
      GOTO230
  280 PRINT*,"*--------------------FPF ASSESSMENTS--------------------*"
      GOTO290
C
C         SUM LOSSES DUE TO LAST TYPE OF MISSION.
  230 DO 210 K=1,55
      DO 210 L=1,2
      PLOSS(K,L,2)=PLOSS(K,L,2)+PLOSS(K,L,1)
  210 PLOSS(K,L,1)=0.
      FLAG=FLAG+1
C
C         BRANCH FOR NEXT TYPE OF MISSION
      IF(FLAG.LT.3)GOTO100
C
C         OUTPUT RESULTS
      IF(IRUN.EQ.1)GO TO 999
      PRINT303
  303 FORMAT(1H1)
      PRINT*,"*----------------INDIRECT FIRE ASSESSMENTS----------------*"
  290 DO 390 J=1,2
      IPCINT=1
      L=1
      IF(L.EQ.J)L=2
      DO 340 K=1,55
      TKILL=0.
      CLOST=0.
      IF(FLAG.EQ.3)GOTO300
      AKILL=PLOSS(K,L,1)
      IF(K.LT.16.AND.K.NE.13)GOTO344
      CLOST=AKILL*CREWS(K-12,L)
  344 TKILL=AKILL
      GOTO345
  300 DO 340 I=43,55
      IF(J.EQ.2)GOTO310
      AKILL=(ALOSS(I,K)-IFIX(ALOSS(I,K)/PACK(1))*PACK(1))/10.
      GOTO320
  310 AKILL=IFIX(ALOSS(I,K)/PACK(1))/10.
  320 IF(K.LT.16.AND.K.NE.13)GOTO330
      CLOST=CLOST+AKILL*CREWS(K-12,L)
  330 TKILL=TKILL+AKILL
  340 CONTINUE
```

Figure L-1. CANNON program code (continued).

```
  345 IF(TKILL.LT..1.AND.CLOST.LT..1)GOTO390
      IF(IPOINT.EQ.1)GOTO380
      IF(J.EQ.2)GOTO360
      PRINT350
  350 FORMAT(" I",57X,"I")
      PRINT*,"I                         RED LOSSES TO BLUE                I"
      GOTO370
  360 PRINT350
      PRINT*,"*-------------------------------------------------------*"
      PRINT350
      PRINT*,"I                         BLUE LOSSES TO RED                I"
  370 IPOINT=1
      PRINT*,"I                 ITEM       # LOST     CREW LOST           I"
      PRINT350
  380 IF(K.LT.16.AND.K.NE.13)GOTO388
      PRINT385,K,TKILL,CLOST
      GOTO390
  385 FORMAT(" I",14X,I2,6X,F5.1,5X,F6.1,15X,"I")
  388 PRINT389,K,TKILL
  389 FORMAT(" I",14X,I2,6X,F6.1,27X,"I")
  390 CONTINUE
      PRINT350
      PRINT*,"*=======================================================*"
      IF(FLAG.NE.3)GOTO230
      PRINT305
  999 IF(IRUN.EQ.1)PRINT*," INDIRECT FIRE ASSESSMENTS PRINTED HERE"
      CALL LOSS(43,58,1,38)
 1000 END
```

Figure L-1. CANNON program code (concluded).

Table L-2. Program variables for CLGP.

Variable	Description
AKILL	Target losses to CLGP fire
BMT	Battery missions per tube
CM	Total CLGP missions fired
FD	Fraction damage
FDF	Fire distribution factor
I	Firer weapon index
IPSN	Positioning units index for contact
J	Firer force index
K	Target weapon index
KK	Infantry weapon index
L	Target for index
MAX	Maximum CLGP mission to fire
OA	Operational availability
PLOSS	Total victims killed
PREC	CLGP SSKPs and GLLD suppression factor
R	Number of CLGP rounds fired
T	Number of targets available to CLGP firer

NOTE: All COMMON variables are defined in table F-1.

```
      SUBROUTINE CLGP(CMT,I,PLOSS,CM)
      COMMON IA,ID,IP,IENGAG,ITERON,IVIS,IMOUNT,MINES,OFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PACK(2),
     3 ELMT(60,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPE(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PSN(6,2,2),PLT(15),KEY(41)
      DIMENSION PLOSS(55,2,2),PREC(5),OA(15)
      DATA(OA(I),I=1,15)/3*.78,.62,4*.81,0.,5*.81,0./
      L=2
      J=1
      MAX=CMT
   10 PRINT*,"ENTER # CLGP MISSIONS TO FIRE (MAX=",MAX,") -"
      READ*,CM
      IF(IRUN.EQ.3.AND.CM.GT.MAX)CM=MAX
      IF(IRUN.EQ.1)WRITE(5,*)CM
      IF(IRUN.EQ.3)PRINT*,CM
      IF(CM.EQ.0.)RETURN
      IF(CM.GT.0.AND.CM.LE.MAX)GOTO20
      PRINT*,"INCORRECT ENTRY - TRY AGAIN"
      GOTO10
C
   20 FLF=0.
      DO 25 K=16,30
   25 FLF=FLF+ELMT(K,L)*OA(K-15)
C
      CALL OPENMS(3,KEY,41,0)
      CALL READMS(3,PREC,6,40)
      CALL CLOSMS(3)
      DO 30 K=16,30
      FD=PREC(5)
      IF(K.GT.19)FD=PREC(6)
      IPSN=1
      IF(L.EQ.ID)IPSN=2
C
      IF(ELMT(K,L)*OA(K-15).LE.0.)GOTO30
      R=CM*ELMT(K,L)*OA(K-15)/FLF
      T=ELMT(K,L)*OA(K-15)*PSN(IENGAG,IPSN+2)
      IF(FD/T.GT.1.)GOTO35
      AKILL=(1.-(1.-FD/T)**(2.*R))*T*PREC(4)
      AKILL=IFIX(AKILL*10.+.5)/10.
C
C          BRANCH FOR PERSONNEL TARGETS
      IF(K.EQ.3)GOTO100
      ALOSS(I,K)=ALOSS(I,K)+IFIX(AKILL*10.+.001)*PACK(L)
      PLOSS(K,L,1)=PLOSS(K,L,1)+IFIX(AKILL*10.+.001)/10.
C
C          ASSESS CREW KILLS.
      IF(K.GT.12)ALOSS(I,2)=ALOSS(I,2)+IFIX(AKILL*CREWS(K-12,L)*10.+.001
     1 )*PACK(L)
      IF(K.GT.12)PLOSS(2,L,1)=PLOSS(2,L,1)+IFIX(AKILL*CREWS(K-12,L)*10.
     C+.001)/10.
      IF((K.NE.21.AND.K.NE.25).OR.IMOUNT.EQ.1)GOTO30
      IF(L.EQ.ID)GOTO30
      AKILL=IFIX(AKILL*10.+.5)/10.
C
C          ADJUSTMENT OF DISMOUNTED INFANTRY WEAPONS.
```

Figure L-2. CLGP program code. (Continued next page.)

```
190 DO 191 KK=3,15
    IF(ELNT(KK,L)-PLOSS(KK,L,2).LE.0.)GOTO191
    ALOSS(I,KK)=ALOSS(I,KK)+IFIX(AKILL*PLT(KK)*10.+.001)*PACK(L)
    PLOSS(KK,L,1)=PLOSS(KK,L,1)+IFIX(AKILL*PLT(KK)*10.+.001)/10.
191 CONTINUE
 30 CONTINUE
C
    SHOTS(35,J)=SHOTS(35,J)+2.*CM
C
    RETURN
    END
```

Figure L-2. CLGP program code (concluded).

APPENDIX M

OVLY 8 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX M

OVLY 8 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN program source code and list of program variables for OVLY 8 (SUPRES), the suppression overlay. SUPRES determines the overall and fire support suppression factors for both the attacking and defending forces. Table M-1 contains a list of the SUPRES program variables. Figure M-1 is the program source code.

Table M-1. Program variables for SUPRES.

Variable	Description
FACT	Suppression factor data array
I	Array index
STALE	Limits of firepower ratios which index suppression factor data

NOTE: All COMMON variables are defined in table F-1.

```
      OVERLAY(SUPRES,10,0)
      PROGRAM OVLY8
      COMMON IA,ID,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,SFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 SF(2),FSSF(2),PACK(2),
     3 ELMT(30,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PSN(6,2,2),PLT(15),KEY(41)
      DIMENSION FACT(12,6,2),STALE(11)
      DATA (STALE(I),I=1,11)/.6,1.,1.5,2.,2.5,3.,3.5,4.,5.,6.,8./
      DATA (((FACT(I,J,K),K=1,2),J=1,6),I=1,12)/
     12.1,8.3,1.2,7.4,.8,3.7,1.4,22.5,2.7,15.0,3.0,11.1,3.8,3.8,1.8,4.4,
     21.2,2.2,1.8,14.0,3.6,9.3,3.9,6.6,4.5,3.2,2.7,3.0,1.8,1.5,2.4,9.9,4
     3.6,5.6,4.8,4.5,4.8,3.0,3.6,2.4,2.4,1.2,3.0,8.1,6.0,5.4,6.0,3.6,5.6
     4,2.5,4.2,2.0,2.8,1.0,3.5,7.2,6.9,4.8,6.9,3.0,6.5,2.3,5.1,1.8,3.4,.
     59,3.6,6.3,7.8,4.5,7.8,2.7,7.3,2.1,5.9,1.7,3.9,.9,4.5,6.5,8.9,4.3,8
     6.7,2.6,8.1,2.0,6.6,1.6,4.4,.8,5.0,6.3,9.9,4.2,9.5,2.4,9.8,1.7,8.1,
     71.4,5.4,.7,6.0,5.4,12.0,3.6,11.4,2.1,11.3,1.6,9.9,1.3,6.2,.7,6.8,5
     8.3,13.5,3.5,12.6,2.0,14.4,1.5,12.0,1.2,8.3,.6,9.0,5.0,18.0,3.3,16.
     93,1.5/
      DO 10 I=1,10
      IF(FPR.LE.STALE(I))GO TO 100
   10 CONTINUE
      I=11
  100 SF(IA)=FACT(I,IENGAG,2)/100.
      SF(ID)=FACT(I,IENGAG,1)/100.
      DO 1000 I=1,10
      IF(FSFPR.LE.STALE(I))GO TO 10000
 1000 CONTINUE
      I=11
10000 FSSF(IA)=FACT(I,IENGAG,2)/100.
      FSSF(ID)=FACT(I,IENGAG,1)/100.
      END
```

Figure M-1. OVLY8 (SUPRES) program code.

APPENDIX N

OVLY 9 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX N

OVLY 9 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN code listing and variable list for the OVLY 9 (RESULT) program. RESULT is the program used to record the overall results of a sector or critical incident battle. The OVLY 9 program variables are listed in table N-1; the FORTRAN source code listing is in figure N-1.

Table N-1. Program variables for OVLY9 (RESULT).
(Continued next page.)

Variable	Description
AKILL	Unpacked ALOSS killer/victim variable
COSCOM	Equipment repairable at Corp level
DIV	Equipment repairable at Division level
FOS	Percent of repairable equipment at Corp Level
I	Killer weapon index
II	Victim index for killer/victim matrix output
IK	Weapon code index counter
IKILL	Data indices for killer
INX	Victim equipment recoverability category index
IPP	Combat posture index for recoverability data
IT	Victim weapon index
ITH	Killer equipment recoverability category index
IVICT	Data indices for victim
J	Victim force index
K	Loss category index
KAT	Killer category index
KIND	Force color

Table N-1. Program variables for OVLY9 (RESULT) (continued).

Variable	Description
L	Killer weapon index beginning
LL	Victim weapon index beginning
M	Killer weapon index end
MAP	Index to aggregate killer weapon systems
MI	Victim nomenclature index
MM	Victim weapon index end
PREC	Percent (function) of Red systems recoverable
RECOV	Percent of recoverable equipment
RECV	Total recoverable equipment
REP	Total recoverable equipment
REPAIR	Total recoverable equipment
REP10	Equipment repairable in ten (10) days
REP2	Equipment repairable in two (2) days
REP5	Equipment repairable in five (5) days
SUM	Victim losses cumulated over all killers
TABLE7	Losses by category of killer
THEA	Equipment repairable at Theater level
THER	Percent of equipment repairable at Theater level

Table N-1. Program variables for OVLY9 (RESULT) (concluded).

Variable	Description
TKILL	Sum of losses recoverable/nonrecoverable
TLOSS	Total losses to killer category KAT
TOTAL	Sum of all losses incurred by unit
VCLASS	Victim nomenclature
XKILL	Weapon system losses
XNREP	Total nonrepairable equipment
XREP	Total repairable equipment

NOTE: All COMMON variables are defined in table F-1.

```
      PROGRAM OVLY9
      COMMON IA,ID,IP,IFDBAG,ITERRN,IVIS,IMOUNT,MINES,CFPR,FSFPR,FPR,
     1 ATIME,IFIRST,IRUN,
     2 CF(2),FSSF(2),PACK(2),
     3 ECMT(80,2),ALOSS(60,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 [illegible](6,2,2),PLT(15),KEY(41)
      COMMON/ONE/LFIT(30),ARRAY(90),MYBUF(1024),P(80,2),ACI,
     1ACLINE,ASECT
      DIMENSION MAP(80),TABLE7(60,8),RECOV(2,2,3),THER(2,2,3),FOS(2,2,3)
     1 ,[illegible](4),VCLASS(6),IKILL(6,2),IVICT(6,2),XKILL(65,20,2),PREC(6)
      DATA(VCLASS(I),I=1,6)/"ARMOR","INFANTRY","MINES","ARTILLERY",
     1"HELICOPTER","ADA"/
      DATA((IKILL(I,J),I=1,6),J=1,2)/11,3,5,43,59,31,[illegible],3,5,58,65,42/
      DATA((IVICT(I,J),I=1,6),J=1,2)/1,1,16,1,1,59,32,15,30,58,32,65/
      DATA(MAP(I),I=1,80)/4*1,6,4*1,[illegible],5*4,4*3,3*4,4*3,4,3*0,
     14*[illegible],2*[illegible],[illegible],[illegible],[illegible]7,18*6/
      DATA(((RECOV(I,J,K),I=1,2),J=1,2),K=1,3)/.92,.[illegible],.89,.56,.90,.67,
     1 .87,.[illegible],.92,.87,.[illegible],.[illegible]/
      DATA(((THER(I,J,K),I=1,2),J=1,2),K=1,3)/.11,.15,.47,.48,.22,
     1.25,.[illegible],.51,.1[illegible],.2[illegible],2*.3/
      DATA(((FOS(I,J,K),I=1,2),J=1,2),K=1,3)/.33,.49,.21,.28,.26,.5,
     1.1[illegible],.2[illegible],.32,.4[illegible],.[illegible],.[illegible]/
      CALL OPENMS(3,KEY,41,0)
      CALL READMS(3,P,FC,[illegible],40)
      CALL CLOSMS(3)
    1 FORMAT("1")
      WRITE(6,111)
  111 FORMAT("1************************ BATTLE STATISTICS*****************
     1*************")
      WRITE(6,2)
    2 FORMAT(" ")
      WRITE(6,2)
      DO [illegible] J=1,2
      KIND="[illegible]"
      IF(J.EQ.1)KIND="[illegible]"
      WRITE(6,2)
      WRITE(6,10) KIND
   10 FORMAT(" ",1[illegible],A4," LOSS & DAMAGE DISTRIBUTION")
      IF(J.EQ.2)GOTO12
      WRITE(6,4)
    4 FORMAT("            DAMAGE/                          EXCEED THEATER   COSCOM
     1    DIV")
      WRITE(6,5)
    5 FORMAT(" TYPE    LOSSES    [illegible]     RECV        REPAIR        REPAIR
     1    REPAIR")
      GOTO13
   12 WRITE(6,6)
    6 FORMAT("            DAMAGE/                   TOTAL               REPAIRA
     1BLE")
      WRITE(6,7)
    7 FORMAT(" TYPE    LOSSES    [illegible]    REPAIRED        2 DAYS   5 DAY
     1S 10 DAYS")
   13 DO 14 I=1,80
      DO 14 K=1,8
   14 TABLE7(I,K)=0.
```

Figure N-1. OVLY9 (RESULT) program code (continued next page).

Best Available Copy

```
      DO 40 K=1,80
      REPAIR=0.
      XNREP=0.
      DO 14 KAT=1,6
   14 TLOSS(KAT)=0.
      TKILL=0.
      DO 20 I=1,80
      IF(J.EQ.2)GOTO16
      AKILL=IFIX(ALOSS(I,K)/PACK(I))/10.
      GOTO18
   16 AKILL=(ALOSS(I,K)-IFIX(ALOSS(I,K)/PACK(I))*PACK(I))/10.
   18 AKILL=IFIX(AKILL+.5)
      IF(AKILL.EQ.0.)GOTO20
      IF(MAP(I).EQ.0)GOTO20
      KAT=MAP(I)
      TLOSS(KAT)=TLOSS(KAT)+AKILL
   20 CONTINUE
      IPP=2
      IF(I1.EQ.1)IPP=1
      DO 30 KAT=1,6
      ITH=2
      IF(KAT.EQ.2)ITH=1
      INX=0
      IF(K.GE.16.AND.K.LE.19)INX=1
      IF(K.GE.20.AND.K.LE.29)INX=2
      IF(K.GE.31.AND.K.LE.57)INX=3
      TKILL=TKILL+TLOSS(KAT)
      TABLE7(K,KAT)=TABLE7(K,KAT)+TLOSS(KAT)
      IF(INX.EQ.0)GOTO30
      IF(J.EQ.1)GOTO31
      REPAIR=IFIX(REPAIR+TLOSS(KAT)+.5)
      GOTO30
   31 REPAIR=REPAIR+IFIX(RECOV(IPP,ITH,INX)*TLOSS(KAT)+.5)
   30 CONTINUE
      IF(TKILL.LT..5)GOTO40
      IF(J.EQ.2)GOTO34
      RECV=REPAIR
      XNREP=TKILL-RECV
      THEA=IFIX(RECV*THER(IPP,ITH,INX)+.5)
      COSCOM=IFIX(RECV*FOS(IPP,ITH,INX)+.5)
      DIV=TKILL-(XNREP+THEA+COSCOM)
      WRITE(6,32) K,TKILL,XNREP,RECV,THEA,COSCOM,DIV
   32 FORMAT(" ",1X,I2,3X,F6.0,3X,F6.0,2X,F6.0,7X,F6.0,7X,F6.0,2X,F6.0)
      GOTO40
C
C              REP REPAIRABLE ITEMS
C
   34 REP2=IFIX(REPAIR*PREC(1)+.5)
      REP10=IFIX(REPAIR*PREC(3)+.5)
      REP5=IFIX(REPAIR*PREC(2)+.5)
C
C              REP NONREPAIRABLE ITEMS
C
      REP=REP2+REP5+REP10
      XREP=TKILL-REP
      IF(TKILL.EQ..0)GOTO40
      WRITE(6,36) K,TKILL,XREP,REP,REP2,REP5,REP10
```

Figure N-1. OVLY9 (RESULT) program code (continued).

```
 36 FORMAT(" ",1X,I2,2X,F6.0,1X,F6.0,5X,F6.0,5X,F6.0,2X,F6.0,3X,F6.0)
 40 CONTINUE
    WRITE(6,55)
 55 FORMAT("1")
    WRITE(6,42) KIND
 42 FORMAT(" ",20X,A4," LOSSES BY SOURCE OF LOSS")
    WRITE(6,3)
  3 FORMAT(" TYPE  INF CBT  INF FIRE  TANK  ATGM  ADA  MINES  A/HEL  T
   1ACAIR  TOTAL")
    DO 48 I=1,F0
    TOTAL=TABLE7(I,1)+TABLE7(I,2)+TABLE7(I,3)+TABLE7(I,4)+TABLE7(I,5)+
   1TABLE7(I,6)+TABLE7(I,7)+TABLE7(I,8)
    IF(TOTAL.LT..5)GOTO48
    WRITE(6,44) I,(TABLE7(I,M),M=1,8),TOTAL
 44 FORMAT(" ",1X,I2,4X,F6.0,4X,F6.0,1X,F5.0,1X,F5.0,1X,
   1F6.0,1X,F6.0,1X,F6.0,2X,F6.0,1X,F6.0)
 48 CONTINUE
    WRITE(6,55)
 50 CONTINUE
    WRITE(6,55)
C
C              AMMUNITION EXPENDITURES
C
    WRITE(6,9)
  9 FORMAT("            AMMUNITION EXPENDITURE")
    WRITE(6,21)
 21 FORMAT("    BLUE                              RED")
    WRITE(6,22)
 22 FORMAT(" TYPE-NUMBER                          TYPE-NUMBER")
    DO 60 I=1,35
    IF(SHOTS(I,1).EQ.0..AND.SHOTS(I,2).EQ.0.)GOTO60
    WRITE(6,58) I,SHOTS(I,1),I,SHOTS(I,2)
 58 FORMAT(" ",I3,F8.0,16X,I3,F8.0)
 60 CONTINUE
C
C         KILLER-VICTIM MATRIX
C
    WRITE(6,1)
    DO 92 MI=1,6
    L=IKILL(MI,1)
    M=IKILL(MI,2)
    LL=IVICT(MI,1)
    MM=IVICT(MI,2)
    DO 90 J=1,2
    TOTAL=0.
    IT=0
    DO 502 I=L,M
    IT=IT+1
    DO 501 K=LL,MM
    IF(J.EQ.1)GOTO131
    XKILL(K,IT,J)=(ALOSS(I,K)-IFIX(ALOSS(I,K)/PACK(1))*PACK(1))/10.
    GOTO503
131 XKILL(K,IT,J)=IFIX(ALOSS(I,K)/PACK(1))/10.
503 TOTAL=TOTAL+XKILL(K,IT,J)
501 CONTINUE
502 CONTINUE
    IF(TOTAL.EQ.0.)GOTO90
```

Figure N-1. OVLY9 (RESULT) program code (continued).

```
      KIND=" RED"
      IF(J.EQ.2)KIND="BLUE"
      WRITE(6,120)
  120 FORMAT(" ***************************************************KILLER-V
     1ICTIM MATRIX****************************************************
     1******")
      WRITE(6,121)KIND,VCLASS(MI)
  121 FORMAT(8X,"*",45X,A4,1X,A10)
      WRITE(6,122)
  122 FORMAT(" VICTIM *",47X,"KILLER")
      WRITE(6,128)(IK,IK=L,M)
  128 FORMAT(8X,"*",5X,20(I2,4X))
      WRITE(6,123)
  123 FORMAT("*************************************************************
     1*****************************************************************
     2******")
      DO 80 I=LL,MM
      SUM=0.
      DO 82 II=1,IT
      SUM=SUM+XKILL(I,II,J)
   82 CONTINUE
      IF(SUM.LE.0.)GOTO80
      WRITE(6,161)I,(XKILL(I,II,J),II=1,IT)
  161 FORMAT(3X,I2,3X,"* ",20(F6.1))
      WRITE(6,125)
  125 FORMAT(" ")
   80 CONTINUE
      WRITE(6,123)
      WRITE(6,1)
   90 CONTINUE
   92 CONTINUE
      END
```

Figure N-1. OVLY9 (RESULT) program code (concluded).

APPENDIX O

OVLY 10 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX O

OVLY 10 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN program source code of FORCE, the force manipulation overlay (OVLY 10), and a list of the program variables used in the overlay. The FORCE program variable list is given in table O-1, and the program source code is presented in figure O-1.

Table O-1. List of program variables for FORCE.

Variable	Description
AJ	Quantity of weapons to adjust
CIL	Combat intensity level factors
CV	Relative effectiveness
I	Do-loop index
IFLAG	Logic flag
IJ	SRC record weapon index
INX	Gamer response variable
J	Force designator
JJ	FORCE record weapon index
KIND	Force color
M	Indexed-sequential file status variable
MM	Type of weapons to adjust
PAR	Parent unit ID
PARENT	Parent unit ID
TFPS	Total firepower score
UEFF	Unit effectiveness
UNIT	Unit ID
XCI	Critical incident name
XSECT	Sector number

NOTE: All COMMON variables are defined in table F-1.

```
      OVERLAY(FORCE,12,0)
      PROGRAM OVLY10
      COMMON IA,ID,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,CFPR,FSFPR,FPR,
     1ATIME,IFIRST,IRUN,
     2SF(2),FSSF(2),PACK(2),
     3ELMT(80,2),ALOSS(80,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FFS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PEN(6,2,2),PLT(15),KEY(41)
      COMMON/ONE/LFIT(35),ARRAY(50),MYBUF(1024),D(80,2),ACI,
     ,ASCENE,ASECT
      COMMON/TWO/IFIT(35),DERAY(46),NYBUF(1024)
      DIMENSION CIL(5)
      DATA(CIL(I),I=1,5)/1000.,5.,4.,2.5,1./
      DO 6601 J=1,2
      DO 6601 I=1,80
      D(I,J)=0.
 6601 ELMT(I,J)=0.0
      IF(ACI.NE.0)GOTO5
C
      PRINT*,"ENTER CI MNEMONIC -"
      READ10,ACI
   10 FORMAT(1A10)
      IF(IRUN.EQ.1)WRITE(5,10)ACI
      IF(IRUN.EQ.3)PRINT18,ACI
   18 FORMAT(" ",1A10)
    1 FORMAT(1A1)
    8 FORMAT(" ",1A1)
C
    5 PRINT*,"ENTER SECTOR NUMBER -"
      READ*,ASECT
      IF(IRUN.EQ.1)WRITE(5,*)ASECT
      IF(IRUN.EQ.3)PRINT*,ASECT
      IF(ASECT.GE.1.AND.ASECT.LE.25)GOTO20
      PRINT*,"INVALID SECTOR ENTERED!"
      GOTO5
   20 PRINT*,"??? O P T I O N ???"
      READ*,INX
      IF(IRUN.EQ.1)WRITE(5,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."T")GOTO101
      IF(INX.GE.0.AND.INX.LE.7)GOTO22
      PRINT*,"INVALID OPTION ENTERED!"
  101 PRINT*,"ENTER 0 TO PROCEDE WITH ASSESSMENTS"
      PRINT*,"        1 TO LOAD UNITS INTO SECTOR"
      PRINT*,"        2 TO REMOVE UNITS FROM SECTOR"
      PRINT*,"        3 TO CREATE A NEW UNIT"
      PRINT*,"        4 TO ADJUST WPNS IN A UNIT"
      PRINT*,"        5 TO ATTACH A UNIT TO A NEW PARENT"
      PRINT*,"        6 TO DISPLAY A UNIT"
      PRINT*,"        7 TO DELETE A UNIT FROM FORCE FILE"
      GOTO20
   22 IF(INX.EQ.0)GOTO400
      GOTO(100,500,700,200,300,600,800),INX
C
  100 PRINT*,"ENTER PARENT OF UNIT(S) TO BE LOADED INTO SECTOR -"
      READ10,ARRAY(1)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(1)
```

Figure O-1. OVLY10 (FORCE) program code.

```
      IF(IRUN.EQ.3)PRINT18,ARRAY(1)
      PARENT=ARRAY(1)
      ARRAY(7)=90909.
      CALL OPENM(LFIT,3LI-O,1LR)
      CALL GET(LFIT,ARRAY,ARRAY(1),0,10)
      IF(ARRAY(7).NE.90909.)GOTO110
      PRINT115,PARENT
  115 FORMAT(" UNIT ",A10," IS NOT ON FORCE FILE!")
      GOTO160
C
C
  110 PRINT*,"ENTER UNIT ID (OR ALL) -"
      READ10,UNIT
      IF(IRUN.EQ.1)WRITE(5,10)UNIT
      IF(IRUN.EQ.3)PRINT18,UNIT
C
      IFLAG=0
  113 IF(UNIT.EQ."ALL")GOTO150
      IF(ARRAY(2).EQ.UNIT)GOTO150
  112 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,2LFP)
      IF(M.EQ.100B)GOTO130
      IF(ARRAY(1).NE.PARENT)GOTO130
      GOTO113
C
  150 ARRAY(4)=ASEGT
      ARRAY(5)=ACI
      IFLAG=1
      CALL REPLC(LFIT,ARRAY,900,ARRAY(1))
      GOTO112
C
  130 IF(IFLAG.EQ.0)PRINT115,UNIT
  160 CALL CLOSEM(LFIT)
  165 PRINT*,"LOAD ANOTHER UNIT?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO100
      IF(INX.EQ."N")GOTO20
      PRINT2
      GOTO165
C
C
  200 PRINT*,"ENTER PARENT OF UNIT TO BE ADJUSTED -"
      READ10,ARRAY(1)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(1)
      IF(IRUN.EQ.3)PRINT18,ARRAY(1)
C
      PRINT*,"ENTER UNIT ID-"
      READ10,ARRAY(2)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(2)
      IF(IRUN.EQ.3)PRINT18,ARRAY(2)
C
      ARRAY(7)=90909.
      CALL OPENM(LFIT,3LI-O,1LR)
      CALL GET(LFIT,ARRAY,ARRAY(1))
      IF(ARRAY(7).EQ.90909.)GOTO55
```

Figure O-1. OVLY10 (FORCE) program code (continued).

```
      PRINT*,"ENTER ITEM ID,NEW QTY--0,0 WHEN DONE  "
   50 READ*,MM,AJ
      IF(IRUN.EQ.1)WRITE(5,*)MM,AJ
      IF(IRUN.EQ.3)PRINT*,MM,AJ
      IF(MM.EQ.0) GO TO 58
      IF(MM.LT.0.OR.MM.GT.80)GOTO57
      IF(ARRAY(MM+10)+AJ.LT.0.)GOTO56
      ARRAY(MM+10)=ARRAY(MM+10)+AJ
      PRINT*,"NEXT-"
      GO TO 50
   56 PRINT*,"ENTRY REDUCES # WEAPONS IN UNIT BELOWZERO - ENTRY IGNORED!
     ."
      GOTO50
   57 PRINT*,"INVALID ITEM CODE - ENTRY IGNORED!"
      GOTO50
   58 IF(IRUN.EQ.1)GOTO52
      J=ARRAY(3)
      UEFF=0.
      DO 59 I=1,80
      IF(ARRAY(I+10).LE.0.)GOTO59
      UEFF=UEFF+ARRAY(I+10)*EPS(I,J)
   59 CONTINUE
      UEFF=UEFF/ARRAY(5)*100.
      ARRAY(8)=UEFF
      CALL REPLC(LFIT,ARRAY,900,ARRAY(1))
   52 CALL CLOSEM(LFIT)
  210 PRINT*,"ANYMORE UNITS TO CHANGE? "
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO200
      IF(INX.EQ."N")GOTO20
      PRINT2
      GOTO210
   55 PRINT 115, ARRAY(2)
      ARRAY(7)=0.
      GO TO 52
C
  700 KIND="BLUE"
      ARRAY(1)="SRG"
      J=1
  720 PRINT725,KIND
  725 FORMAT(" ARE THERE ANY ",A4," UNITS TO CREATE?")
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO715
      IF(INX.EQ."N")GOTO790
      PRINT2
      GOTO720
    2 FORMAT(" INCORRECT!  RESPONSE MUST BE YES OR NO - TRY AGAIN")
C
  715 ARRAY(3)=J
      DO 707 INX=11,90
  707 ARRAY(INX)=0
      PRINT*,"ENTER PARENT UNIT ID -"
      READ10,ARRAY(1)
```

Figure 0-1. OVLY10 (FORCE)program code (continued).

```
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(1)
      IF(IRUN.EQ.3)PRINT18,ARRAY(1)
C
      PRINT*,"ENTER UNIT ID -"
      READ10,ARRAY(2)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(2)
      IF(IRUN.EQ.3)PRINT18,ARRAY(2)
  731 PRINT*,"CREATE BY SRC'S?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO728
      IF(INX.EQ."N")GOTO729
      PRINT2
      GOTO731
  729 ARRAY(6)=0
      PRINT*,"ENTER WPN ID,QTY--0,0 WHEN DONE  "
  703 READ*,MM,AJ
      IF(IRUN.EQ.1)WRITE(5,*)MM,AJ
      IF(IRUN.EQ.3)PRINT*,MM,AJ
      IF(MM.EQ.0) GO TO 701
      ARRAY(MM+10)=ARRAY(MM+10)+AJ
      ARRAY(6)=ARRAY(6)+AJ*FPS(MM,J)
      PRINT*,"NEXT -"
      GO TO 703
  701 PRINT*,"ENTER RELATIVE EFFECTIVENESS -"
      READ*,CV
      IF(IRUN.EQ.1)WRITE(5,*)CV
      IF(IRUN.EQ.3)PRINT*,CV
      IF(CV.GE.0..AND.CV.LE.10000.)GOTO702
      PRINT*,"INVALID REL. EFF. - TRY AGAIN!"
      GOTO701
  702 ARRAY(6)=ARRAY(6)*100./CV
      ARRAY(8)=CV
C
      IF(IRUN.EQ.1)GOTO704
      CALL OPENM(LFIT,3LI-O,1LR)
      CALL PUT(LFIT,ARRAY,900,ARRAY(1))
      CALL CLOSEM(LFIT)
  704 DO 710 I=11,90
  710 ARRAY(I)=0
  711 PRINT*,"CREATE ANOTHER UNIT FOR THIS FORCE?    "
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO715
      IF(INX.EQ."N")GOTO790
      PRINT2
      GOTO711
  728 PRINT*,"ENTER SRC--0 WHEN DONE"
      ARRAY(6)=0
  730 READ10,ARRAY(2)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(2)
      IF(IRUN.EQ.3)PRINT18,ARRAY(2)
      IF(ARRAY(2).EQ."0")GOTO701
      ARRAY(3)=99999
      CALL OPENM(IFIT,3LI-O,1LR)
```

Figure O-1. OVLY10 (FORCE) program code (continued).

```
      CALL GETN(IFIT,BRRAY,BRRAY(1))
      CALL CLOSEM(IFIT)
      IF(BRRAY(2).EQ.90909)GOTO737
      DO 736 IJ=1,22
      JJ=IJ*2+1
      INX=BRRAY(JJ)
      IF(INX.EQ.0)GOTO736
      ARRAY(INX+10)=ARRAY(INX+10)+BRRAY(JJ+1)
      ARRAY(6)=ARRAY(6)+BRRAY(JJ+1)*FPS(INX,J)
  736 CONTINUE
      DO 739 I=3,24
  739 BRRAY(I)=0
      PRINT*,"NEXT-"
      GOTO734
  737 PRINT738,BRRAY(2)
      BRRAY(3)=0
  738 FORMAT(" SKC ",1A10," NOT ON FILE")
      PRINT*,"NEXT-"
      GOTO734
C
  790 IF(J.EQ.2)GOTO020
      J=2
      KIND="REJ"
      GOTO720
C
  800 PRINT*,"ENTER PARENT OF UNIT(S) TO BE DELETED -"
      READ10,ARRAY(1)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(1)
      IF(IRUN.EQ.3)PRINT18,ARRAY(1)
      CALL OPENM(LFIT,3LI-O,1LR)
      PARENT=ARRAY(1)
      ARRAY(7)=90909
      CALL GET(LFIT,ARRAY,ARRAY(1),0,10)
      IF(ARRAY(7).NE.90909)GOTO810
      PRINT115,PARENT
      GOTO880
C
  810 PRINT*,"ENTER UNIT ID (OR ALL) -"
      READ10,UNIT
      IF(IRUN.EQ.1)WRITE(5,10)UNIT
      IF(IRUN.EQ.3)PRINT18,UNIT
C
      IFLAG=0
  813 IF(UNIT.EQ."ALL")GOTO820
      IF(UNIT.EQ.ARRAY(2))GOTO820
  812 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,7LFP)
      IF(M.EQ.100B)GOTO855
      IF(ARRAY(1).NE.PARENT)GOTO855
      GOTO813
C
  820 CALL DELETE(LFIT,ARRAY(1))
      IFLAG=1
      GOTO812
C
  855 IF(IFLAG.EQ.0)PRINT115,UNIT
  880 CALL CLOSEM(LFIT)
```

Figure 0-1. OVLY10 (FORCE) program code (continued).

```
885  PRINT*,"ANOTHER UNIT TO DELETE?"
     READ1,INX
     IF(IRUN.EQ.1)WRITE(5,1)INX
     IF(IRUN.EQ.3)PRINT8,INX
     IF(INX.EQ."Y")GOTO800
     IF(INX.EQ."N")GOTO020
     PRINT2
     GOTO885
C
C
500  PRINT*,"ENTER PARENT OF UNIT(S) TO BE REMOVED -"
     READ10,ARRAY(1)
     IF(IRUN.EQ.1)WRITE(5,10)ARRAY(1)
     IF(IRUN.EQ.3)PRINT18,ARRAY(1)
     CALL OPENM(LFIT,3LI-O,1LR)
     PARENT=ARRAY(1)
     ARRAY(7)=99999
     CALL GET(LFIT,ARRAY,ARRAY(1),0,10)
     IF(ARRAY(7).NE.99999)GOTO550
     PRINT115,PARENT
     GOTO510
C
550  PRINT*,"ENTER UNIT ID (OR ALL) -"
     READ10,UNIT
     IF(IRUN.EQ.1)WRITE(5,10)UNIT
     IF(IRUN.EQ.3)PRINT18,UNIT
C
     IFLAG=0
511  IF(UNIT.EQ."ALL")GOTO520
     IF(UNIT.EQ.ARRAY(2))GOTO520
512  CALL GETN(LFIT,ARRAY,ARRAY(1))
     M=IFETCH(LFIT,2LFP)
     IF(M.EQ.100B)GOTO555
     IF(ARRAY(1).NE.PARENT)GOTO555
     GOTO511
C
520  ARRAY(4)=0.
     IFLAG=1
     CALL REPLC(LFIT,ARRAY,900,ARRAY(1))
     GOTO512
C
555  IF(IFLAG.EQ.0)PRINT115,UNIT
C
510  CALL CLOSEM(LFIT)
515  PRINT*,"REMOVE ANOTHER UNIT?"
     READ1,INX
     IF(IRUN.EQ.1)WRITE(5,1)INX
     IF(IRUN.EQ.3)PRINT8,INX
     IF(INX.EQ."Y")GOTO500
     IF(INX.EQ."N")GOTO020
     PRINT2
     GOTO515
C
C
300  PRINT*,"ENTER PARENT ID OF UNIT BEING ATTACHED -"
     READ10,PARENT
     IF(IRUN.EQ.1)WRITE(5,10)PARENT
```

Figure 0-1. OVLY10 (FORCE) program code (continued).

```
      IF(IRUN.EQ.3)PRINT18,PARENT
C
      PRINT*,"ENTER UNIT ID -"
      READ10,ARRAY(2)
      IF(IRUN.EQ.1)WRITE(5,10)ARRAY(2)
      IF(IRUN.EQ.3)PRINT18,ARRAY(2)
C
      CALL OPENM(LFIT,3LI-O,1LR)
      ARRAY(1)=PARENT
      ARRAY(7)=90909.
      CALL GET(LFIT,ARRAY,ARRAY(1))
      IF(ARRAY(7).EQ.90909.)GOTO320
C
      PRINT*,"ENTER NEW PARENT ID -"
      READ10,PAR
      IF(IRUN.EQ.1)WRITE(5,10)PAR
      IF(IRUN.EQ.3)PRINT18,PAR
      ARRAY(1)=PAR
      CALL PUT(LFIT,ARRAY,900,ARRAY(1))
      ARRAY(1)=PARENT
      CALL DLTE(LFIT,ARRAY(1))
      CALL CLOSEM(LFIT)
C
  310 PRINT*,"ATTACH ANOTHER UNIT?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT18,INX
      IF(INX.EQ."Y")GOTO300
      IF(INX.EQ."N")GOTO20
C
  320 PRINT115,UNIT
      ARRAY(7)=0.
      GOTO310
C
C     DISPLAY SECTION
  600 PRINT*,"ENTER TYPE OF DISPLAY -"
      READ*,INX
      IF(IRUN.EQ.1)WRITE(5,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."T")GOTO601
      IF(INX.GE.1.AND.INX.LE.4)GOTO606
      PRINT*,"INCORRECT ENTRY!!!"
  601 PRINT*,"ENTER 1 TO DISPLAY ALL PARENT UNITS IN FORCEFILE"
      PRINT602,ASECT,AC1
  602 FORMAT("         2 TO DISPLAY ALL PARENT UNITS IN SECTOR ",F3.0,
     1" IN C1 ",A10)
      PRINT*,"         3 TO DISPLAY UNITS IN A SPECIFIC PARENT"
      PRINT*,"         4 TO DISPLAY WEAPONS IN A UNIT"
      GOTO600
C
  606 GOTO(610,610,660,690),INX
C
C     DISPLAY ALL PARENT UNITS
  610 PARENT="ALL"
      IFLAG=0
      JEFF=0.
      TEMS=0.
```

Figure 0-1. OVLY10 (FORCE) program code (continued).

```
      CALL OPENM(LFIT,3LI-O,1LR)
  605 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,2LFP)
      IF(M.EQ.100)GOTO625
      IF(ARRAY(1).NE.PARENT)GOTO620
  614 XSECT=ARRAY(4)
      XCI=ARRAY(5)
      IF(INX.EQ.1)GOTO615
      IF(ASECT.NE.XSECT.OR.ACI.NE.XCI)GOTO605
  615 J=ARRAY(3)
      TEPS=TEPS+ARRAY(6)
      DO 616 I=1,K3
      IF(ARRAY(I+10).LE.0.)GOTO616
      UEFF=UEFF+ARRAY(I+10)*EPS(I,J)
  616 CONTINUE
      GOTO605

  620 IF(IFLAG.EQ.1)GOTO625
      PRINT*
      IF(INX.EQ.1)GOTO621
      PRINT*,"FORCE     ID          EFF"
      GOTO622
  621 PRINT*,"FORCE     ID          EFF   SECT   CI"
  622 IFLAG=1
      GOTO625
  625 IF(M.EQ.100.AND.IFLAG.EQ.0)GOTO690
      IF(INX.EQ.2.AND.(ASECT.NE.XSECT.OR.ACI.NE.XCI))GOTO650
      IF(TEPS.GT.0.)GOTO26
      UEFF=99.
      GOTO27
   26 UEFF=UEFF/TEPS*100.
   27 IF(INX.EQ.1)GOTO640
      PRINT630,J,PARENT,UEFF
  630 FORMAT(" ",I3,4X,A10,2X,F4.0)
      GOTO650
  640 PRINT645,J,PARENT,UEFF,XSECT,XCI
  645 FORMAT(" ",I3,4X,A10,2X,2(F4.0,2X),A10)
  650 PARENT=ARRAY(1)
      UEFF=0.
      TEPS=0.
      IF(M.EQ.100)GOTO690
      GOTO614
C
C     DISPLAY UNITS IN SPECIFIC UNIT
  660 PRINT*,"ENTER PARENT ID -"
      READ10,PARENT
      IF(IRUN.EQ.1)WRITE(5,10)PARENT
      IF(IRUN.EQ.3)PRINT18,PARENT
      ARRAY(1)=PARENT
C
      ARRAY(7)=99999.
      IFLAG=0
      CALL OPENM(LFIT,3LI-O,1LR)
      CALL GET(LFIT,ARRAY,ARRAY(1),0,10)
      IF(ARRAY(7).NE.99999.)GOTO665
      PRINT115,PARENT
      GOTO630
```

Figure O-1. OVLY10 (FORCE) program code (continued).

```
  663 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,"LFP")
      IF(M.EQ.100B)GOTO690
      IF(PARENT.NE.ARRAY(1))GOTO690
  665 UEFF=0.
      J=ARRAY(3)
C
      DO 670 I=1,80
      IF(ARRAY(I+10).LE.0.)GOTO670
      UEFF=UEFF+ARRAY(I+10)*FPS(I,J)
  670 CONTINUE
      IF(ARRAY(6).GT.0.)GOTO71
      UEFF=999.
      GOTO72
   71 UEFF=UEFF/ARRAY(6)*100.
   72 IF(IFLAG.EQ.1)GOTO680
      IFLAG=1
      PRINT*
      PRINT*,"FORCE  PARENT      UNIT          EFF  SECT  CI"
      PRINT675,J,PARENT,ARRAY(2),UEFF,ARRAY(4),ARRAY(5)
  675 FORMAT(" ",I3,4X,2(A10,2X),2(F4.0,2X),A10)
      GOTO663
  680 PRINT685,ARRAY(2),UEFF,ARRAY(4),ARRAY(5)
  685 FORMAT(" ",19X,A10,2X,2(F4.0,2X),A10)
      GOTO663
  690 PRINT*
      CALL CLOSEM(LFIT)
      GOTO699
  695 CALL DISPLAY
  699 PRINT*,"ANOTHER DISPLAY?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO600
      IF(INX.EQ."N")GOTO20
      PRINT2
      GOTO699
C
  400 CALL OPENM(LFIT,"I-O",1LR)
  405 PRINT*,"DO YOU WISH TO SEE UNITS LOADED INTO SECTOR?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y".OR.INX.EQ."N")GOTO415
      PRINT2
      GOTO405
  415 IF(INX.EQ."N")GOTO410
C
      PRINT490,ASECT,ACI
  490 FORMAT(" UNITS LOADED INTO SECTOR ",F3.0," FOR CI ",A10,//,
     1 " FORCE     PARENT          UNIT")
  410 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,"LFP")
      IF(M.EQ.100B)GOTO151
C
      IF(ARRAY(4).NE.ASECT.OR.ACI.NE.ARRAY(5))GOTO410
```

Figure 0-1. OVLY10 (FORCE) program code (continued).

```
      J=ARRAY(3)
      IF(INX.EQ."N")GOTO425
      PRINT 495,J,ARRAY(1),ARRAY(2)
  495 FORMAT(" ",I3,6X,A10,3X,A10)
  425 DO 420 I=1,80
      IF(ARRAY(I+10).EQ.0)GOTO420
      ELMT(I,J)=ELMT(I,J)+ARRAY(I+10)
  420 CONTINUE
      GOTO410
C
  151 CALL CLOSEM(LFIT)
      END
```

Figure 0-1. OVLY10 (FORCE) program code (concluded).

APPENDIX P

OVLY 11 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX P

OVLY 11 PROGRAM CODE AND LIST OF VARIABLES

Appendix P contains the program source code listing and a table of the program variables for OVLY 11 (APPORT), the Jiffy Game loss apportionment overlay. The list of program variables is contained in table P-1. The listing of the FORTRAN program source code is presented in figure P-1.

Table P-1. Program variables for APPORT. (Continued next page).

Variable	Description
AIRKO	Quantity of given type weapons lost to TACAIR being apportioned to unit
AKO	Quantity of given type weapons lost to ground actions being apportioned to unit
CBTINT	Combat intensity factor
CIL	Combat intensity level factor
CLOST	Number of crew personnel lost
CUMLOS	Parent unit loss array
DAIR	Quantity of weapon systems subject to apportionment for TACAIR losses
I	File record word index; weapon system index
ICIL	Combat intensity level index
IFLAG	Logic flag
IHOLD	Automatic CIL allocation indicator
II	Weapon system index
INT	Gamer response variable
J	Force identifier
JJ	Force identifier
K	File record word index
KIND	Force color

Table P-1. Program variables for APPORT (concluded).

Variable	Description
M	Index-sequential file status variable
PAREFF	Parent unit effectiveness
PARENT	Parent unit identifier
PARFPS	Parent unit firepower score
PARINIT	Initial firepower score of parent unit
PERS	Number of non-infantry personnel casualties
TFPS	Total firepower score
UEFF	Unit effectiveness
XL	Packed weapon system losses to all type of combat
XN	Unpacked weapon system losses to all types of combat

NOTE: All COMMON variables are defined in table F-1.

```
      OVERLAY(APPORT,13,0)
      PROGRAM OVLY11
      COMMON IA,ID,IP,IENGAG,ITERN,IVIS,IMOUNT,MINES,DFPR,FSFPR,FPR,
     1ATIME,IFIRST,IRUN,
     2SF(2),FSSF(2),PACK(2),
     3ELMT(80,2),ALOSS(80,80),SHOTS(35,2),SKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),EPOS(6),
     1 PSN(6,2,2),PLT(10),KEY(41)
      COMMON/ONE/LFIT(35),ARRAY(93),MYBUF(1024),D(80,2),ACI,
     1ASCENE,ASECT
      COMMON/THREE/IHIST(35),AH(90),IYBUF(1024)
      DIMENSION XL(80),CIL(6),IHOLD(80,2),CUMLOS(80,2),DAIR(80,2)
      DATA(CIL(I),I=1,6)/1000.,5.,2.,1.33,1.,0./
C
      DO 70 I=1,80
      ARRAY(I)=0
      XL(I)=0.
      DO 50 J=1,79
   50 XL(I)=XL(I)+ALOSS(J,I)
      DO 60 J=1,2
      IHOLD(I,J)=0
      DAIR(I,J)=0.
   60 D(I,J)=0.
   70 CONTINUE
C
      CALL OPENM(LFIT,3LI-O,1L-)
   18 CALL GETN(LFIT,ARRAY,ARRAY(1))
      M=IFETCH(LFIT,2LFP)
      IF(M.EQ.100B)GOTO19
      IF(ASECT.NE.ARRAY(4).OR.ACI.NE.ARRAY(5))GOTO18
   14 PRINT15,ARRAY(2)
   15 FORMAT(" ENTER C I INTENSITY FOR ",A10,"-")
       READ*,ICIL
      IF(IRUN.EQ.1)WRITE(5,*)ICIL
      IF(IRUN.EQ.3)PRINT*,ICIL
      IF(ICIL.EQ."I")GOTO16
      IF(ICIL.GE.0.AND.ICIL.LE.5)GOTO17
      PRINT*,"INVALID CI INTENSITY LEVEL ENTERED."
   16 PRINT*,"COMBAT INTENSITY LEVELS"
      PRINT*,"ENTER 0 FOR UNCOMMITTED UNITS"
      PRINT*,"      1 FOR UNITS OUTSIDE OF DIRECT FIRE"
      PRINT*,"      2 FOR RESERVE UNITS COMMITTED LATE"
      PRINT*,"      3 FOR UNITS ON PERIMETER OF MBA"
      PRINT*,"      4 FOR UNITS IN MAIN BATTLE AREA"
      PRINT*,"      5 FOR UNITS HIT BY TACAIR"
      GOTO14
C
   17 ARRAY(7)=CIL(ICIL+1)
      J=ARRAY(3)
      DO 85 I=1,80
      IF(ICIL.NE.5)GOTO80
      DAIR(I,J)=DAIR(I,J)+ARRAY(I+10)
   80 CSTINT=ARRAY(7)
      IF(CSTINT.EQ.0.)CSTINT=1.
      D(I,J)=D(I,J)+ARRAY(I+10)/CSTINT
   85 CONTINUE
      CALL REPLC(LFIT,ARRAY,93,ARRAY(1))
```

Figure P-1. OVLY11 (APPORT) program code.(Continued next page)

```
      GOTO16
   19 CALL CLOSEM(LFIT)
      DO 35 I=1,80
      D(I,1)=D(I,1)-IFIX(ALOSS(80,I)/PACK(1))/10.
      IF(D(I,1).LT.0.)D(I,1)=0.
      D(I,2)=D(I,2)-(ALOSS(80,I)-IFIX(ALOSS(80,I)/PACK(1))*PACK(1))/10.
      IF(D(I,2).LT.0.)D(I,2)=0.
   35 CONTINUE
C
      IFLAG=0
      DO 10 J=1,2
      DO 10 I=1,80
      IF(J.EQ.2)GOTO13
      KIND="BLUE"
      XN=IFIX(XL(I)/PACK(1))/10.
      IF(D(I,J).GT.0..OR.XN.LE.0.)GOTO9
      XL(I)=XL(I)-XN*10.*PACK(1)
      XN=0.
      PRINT12,I,KIND
   12 FORMAT(" APPORTIONMENT OF ITEM ",I2," LOSSES TO ",A4," FORCE CANNO
     1T BE MADE")
      GOTO9
   13 XN=(XL(I)-IFIX(XL(I)/PACK(1))*PACK(1))/10.
      KIND="RED"
      IF(D(I,J).GT.0..OR.XN.LE.0.)GOTO9
      XL(I)=XL(I)-XN*10.
      XN=0.
      PRINT12,I,KIND
    9 IF(XN.LE.D(I,J))GOTO10
      IF(I.GT.3.AND.I.LT.16)GOTO10
      D(I,J)=0.
      DO 90 II=1,79
      IF(J.EQ.2)GOTO91
      D(I,J)=D(I,J)+IFIX(ALOSS(II,I)/PACK(1))/10.
      GOTO90
   91 D(I,J)=D(I,J)+(ALOSS(II,I)-IFIX(ALOSS(II,I)/PACK(1))*PACK(1))/10.
   90 CONTINUE
      D(I,J)=D(I,J)+ELMT(I,J)
      IFLAG=1
      PRINT25,I,KIND
   25 FORMAT(" INSUFFICIENT OBT INTENSITY LEVELS HAVE BEEN ASSIGNED FOR
     1ITEM ",I2," OF ",A4," FORCE")
      INVLD(I,J)=1
   10 CONTINUE
      IF(IFLAG.EQ.0)GOTO21
      PRINT*,"AUTO-ALLOCATION OF ABOVE HPN SYSTEMS HAS BEEN INITIATED"
C
   21 IF(... .EQ.1) GO TO 130
      WRITE(6,600) AGI,SECT
  600 FORMAT("UNIT STATUS FILE FOR GI ",A10," / SECTOR ",F2.0)
      PAREN="ALL"
      CALL OPENM(LFIT,... ,ILK)
   11 CALL GETN(LFIT,ARRAY,ARRAY(1))
      ...(LFIT,?LF)
      IF(... .EQ. ...) GO TO ...
      IF(ARRAY(...).NE. ...) GO TO 11
      IF(ARRAY(6).NE.SECT) GO TO 11
```

Figure P-1. OVLY11 (APPORT) program code (continued).

```
      IF(PARENT.EQ."ALL") GO TO 702
      IF(ARRAY(1).EQ.PARENT) GO TO 701
 605  IF(PARINIT.GT.0.) PAREFF=PAREPS/PARINIT*100.
      PRINT606,PARENT,PAREFF
 606  FORMAT(" CUMMULATIVE EFFECTIVENESS OF ",A10,"=",F4.0)
      WRITE(6,602)PARENT
 602  FORMAT("0",50X,"PARENT=",A10//55X,"ELMTS   LOST  REMAIN")
      DO 620 I=1,80
      IF(CUMLOS(I,1).EQ.0..AND.CUMLOS(I,2).EQ.0.)GOTO620
      WRITE(6,615)I,CUMLOS(I,1),CUMLOS(I,2)
 615  FORMAT(" ",56X,I3,4X,F5.1,2X,F6.1)
 620  CONTINUE
      WRITE(6,610)PARENT,PAREFF
 610  FORMAT(" ",50X,"EFFECTIVENESS OF ",A10,"=",F4.0)
      IF(M.EQ.1000)GOTO300
 702  PARENT=ARRAY(1)
      PAREFF=0.
      PAREPS=0.
      PARINIT=0.
      DO 625 I=1,80
      DO 625 J=1,2
 625  CUMLOS(I,J)=0.
 701  TEPS=0.
      JEFF=0.
      JJ=ARRAY(3)
      WRITE(6,601) ARRAY(1),ARRAY(2)
 601  FORMAT("0PARENT=",A10,3X,"UNIT=",A10//25X
     1 "ELMTS",3X,"LOST",2X,"REMAIN")
      PEPS=0.
      IF(ARRAY(3).NE.1) GO TO 120
      DO 20 I=1,80
      AIRKO=0.
      IF(ARRAY(I+10).EQ.0.)GOTO20
      IF(ARRAY(7).NE.0.)GOTO80
      XN=IFIX(ALCSS(80,I)/PACK(1))/10.
      IF(XN.GT.DAIR(I,JJ))XN=DAIR(I,JJ)
      AIRKO=ARRAY(I+10)*XN/DAIR(I,JJ)
      AIRKO=IFIX(AIRKO*10.+.5)/10.
      ARRAY(I+10)=ARRAY(I+10)-AIRKO
      IF(I.NE.2)GOTO80
      PEPS=PEPS+AIRKO
 80   IF(I.EQ.2)GOTO20
      IF(D(I,JJ).LE.0.)GOTO20
      XN=IFIX(YL(I)/PACK(1))/10.
      IF(XN.GT.D(I,JJ)) XN=D(I,JJ)
      COTINT=ARRAY(7)
      IF(COTINT.EQ.0.)COTINT=1.
      IF(IHCL2(I,JJ).EQ.1)COTINT=1.
      AKO=(ARRAY(I+10)*XN)/(COTINT*D(I,JJ))
      AKO=IFIX(AKO*10.+.5)/10.
      ARRAY(I+10)=ARRAY(I+10)-AKO
      CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)
      AKO=AKO+AIRKO
      IF(I.NE.2)WRITE(6,604) I,AKO,ARRAY(I+10)
      CUMLOS(I,1)=CUMLOS(I,1)+AKO
      IF(ARRAY(I+10).LT.0) ARRAY(I+10)=0
      IF(I.LT.13)GOTO20
```

Figure P-1. OVLY11 (APPORT) program code (continued).

```
      CLOST=(AKO-AIRKO)*CREWS(I-12,JJ)
      ARRAY(12)=ARRAY(12)-CLOST
      PERS=PERS+CLOST
   20 CONTINUE
      GO TO 200
  120 CONTINUE
      JJ=2
      DO 30 I=1,89
      AIRKO=0.
      IF(ARRAY(I+10).EQ.0.)GOTO30
      IF(ARRAY(7).NE.0.)GOTO85
      XN=(ALOSS(80,I)-IFIX(ALOSS(80,I)/PACK(1))*PACK(1))/10.
      IF(XN.GT.DAIR(I,JJ))XN=DAIR(I,JJ)
      AIRKO=ARRAY(I+10)*XN/DAIR(I,JJ)
      AIRKO=IFIX(AIRKO*10.+.5)/10.
      ARRAY(I+10)=ARRAY(I+10)-AIRKO
      IF(I.NE.2)GOTO85
      PERS=PERS+AIRKO
   85 IF(I.EQ.2)GOTO30
      IF(D(I,JJ).LE.0.)GOTO30
      XN=(XL(I)-IFIX(XL(I)/PACK(1))*PACK(1))/10.
      IF(XN.GT.D(I,JJ)) XN=D(I,JJ)
      CBTINT=ARRAY(7)
      IF(CBTINT.EQ.0.)CBTINT=1.
      IF(INCLD(I,JJ).EQ.1)CBTINT=1.
      AKO=(ARRAY(I+10)*XN)/(CBTINT*D(I,JJ))
      AKO=IFIX(AKO*10.+.5)/10.
      ARRAY(I+10)=ARRAY(I+10)-AKO
      CUMLOS(I,2)=CUMLOS(I,2)+ARRAY(I+10)
      AKO=AKO+AIRKO
      IF(I.NE.2)WRITE(6,604) I,AKO,ARRAY(I+10)
      CUMLOS(I,1)=CUMLOS(I,1)+AKO
      IF(ARRAY(I+10).LT.0) ARRAY(I+10)=0
      IF(I.LT.13)GOTO30
      CLOST=(AKO-AIRKO)*CREWS(I-12,JJ)
      ARRAY(12)=ARRAY(12)-CLOST
      PERS=PERS+CLOST
   30 CONTINUE
  200 IF(ARRAY(12).LT.0)ARRAY(12)=0
      CUMLOS(2,2)=CUMLOS(2,2)+ARRAY(12)
      I=2
      WRITE(6,604)I,PERS,ARRAY(12)
      CUMLOS(2,1)=CUMLOS(2,1)+PERS
  604 FORMAT(25X,I3,4X,F5.1,2X,F6.1)
      DO 700 I=1,89
  700 TEPS=TEPS+ARRAY(I+10)*EPS(I,JJ)
      IF(ARRAY(6).GT.0.) UEFF=TEPS/ARRAY(6)*100.
      WRITE(6,603) ARRAY(2),UEFF
      PRINT603,ARRAY(2),UEFF
  603 FORMAT(" EFFECTIVENESS OF ",A10,"=",F4.0)
      ARRAY(8)=UEFF
      PAEPS=PAEPS+TEPS
      PAINIT=PAINIT+ARRAY(6)
      CALL REPLC(LFIT,ARRAY,900,ARRAY(1))
      DO 40 I=11,90
   40 ARRAY(I)=0.
      GOTO11
```

Figure P-1. OVLY11 (APPORT) program code (continued).

```
300 CALL CLOSFM(LFIT)
    CALL OPENM(IHIST,"L1-C,1L")
    AH(1)="CI LOSSES"
    AH(2)=ACI
    AH(3)=1.
    AH(4)=99999.
    CALL GET(IHIST,AH,AH(1))
    IF(AH(4).EQ.99999.)GOTO035F
    GOTO320
310 CALL GETN(IHIST,AH,AH(1))
    M=IFETCH(IHIST,2LFP)
    IF(M.EQ.100B)GOTO0390
    I=AH(3)
320 DO 330 K=1,KC
330 AH(K+10)=ALOSS(I,K)+AH(K+10)
    CALL REPLC(IHIST,AH,900,AH(1))
    IF(AH(3).EQ.80)GOTO0340
    GOTO0310
340 AH(1)="CI AMMO"
    AH(3)=1.
    AH(4)=99999.
    CALL GET(IHIST,AH,AH(1))
    IF(AH(4).EQ.99999.)GOTO0385
    DO 345 I=1,35
345 AH(I+10)=SHOTS(I,1)+AH(I+10)
    DO 350 I=1,35
350 AH(I+45)=SHOTS(I,2)+AH(I+45)
    CALL REPLC(IHIST,AH,900,AH(1))
    GOTO0390
355 AH(4)=0.
    DO 355 I=1,80
    AH(3)=I
    DO 370 K=1,KC
370 AH(K+10)=ALOSS(I,K)
    CALL PUT(IHIST,AH,900,AH(1))
365 CONTINUE
    AH(1)="CI AMMO"
    AH(3)=1.
    DO 375 I=1,35
375 AH(I+10)=SHOTS(I,1)
    DO 380 I=1,35
380 AH(I+45)=SHOTS(I,2)
    CALL PUT(IHIST,AH,900,AH(1))
    GOTO0390
385 PRINT385,ACI
386 FORMAT("CUMULATIVE AMMO STATS FOR CI ",A10," IS NOT ON FILE")
390 CALL CLOSEM(IHIST)
13. PRINT*,"DISPLAY A UNIT?"
153  READ(5,5)INT
   5 FORMAT(A1)
    IF(ISUN.EQ.1)WRITE(5,5)INT
    IF(ISUN.EQ.3)PRINT8,INT
   8 FORMAT(" ",1A1)
    IF(INT.EQ.1HN) GO TO 1555
    IF(INT.EQ.1HY) GO TO 155
    PRINT7
   7 FORMAT(" INCORRECT  RESPONSE MUST BE YES OR NO -TRY AGAIN")
```

Figure P-1. OVLY11 (APPORT) program code (continued).

```
      GO TO 133
C
  155 CALL DISPLAY
      GOTO 0130
 5555 CONTINUE
      END
```

Figure P-1.. OVLY11 (APPORT) program code (concluded).

APPENDIX Q

OVLY 12 PROGRAM CODE AND LIST OF VARIABLES

APPENDIX Q

OVLY 12 PROGRAM CODE AND LIST OF VARIABLES

This appendix contains the FORTRAN source code and a listing of the program variables for OVLY 12 (BUILD), an overlay which creates and maintains the SRC file during interactive processing of the Jiffy Game. Table Q-1 is a list of the program variables of the overlay, and figure Q-1 is a listing of the program's source code.

Table Q-1. Program variables for BUILD.

Variable	Description
AHOLD	First word of SRC record
AJ	Quantity of weapons being entered
ASRC	SRC identifier
I	SRC record word index
IID	Weapon system item code
INX	Gamer response variable
M	Index-sequential file status variable
MM	Weapon item code being entered
NN	Weapon item code word index

NOTE: All COMMON variables are defined in table F-1.

```
      OVERLAY(BUILD,14,0)
      PROGRAM OVLY12
      COMMON IA,ID,IP,IENGAG,ITERRN,IVIS,IMOUNT,MINES,CFPR,FSFPR,FPR,
     1ATIME,IFIRST,IRUN,
     2SF(2),FSSF(2),PACK(2),
     3ELMT(85,2),ALCSS(85,80),SHOTS(35,2),CKILL(53,2)
      COMMON/DATA/FPS(80,2),CREWS(53,2),APOS(12),DPOS(5),
     1 PSN(6,2,2),PLT(15),KEY(41)
      COMMON/ONE/LFIT(35),ARRAY(90),MYBUF(1024),D(80,2),ACI,
     .ASCENE,ASECT
      COMMON/TWO/IFIT(35),BRRAY(46),NYBUF(1024)
20    BRRAY(1)="SRC"
      AHOLD=BRRAY(1)
      CALL OPENM(IFIT,3LI-O,1LR)
      DO 11 I=2,46
      BRRAY(I) = 0
   11 CONTINUE
C
C              ABOVE DO LOOPS ZERO OUT WORK ARRAYS
C
C
   14 PRINT*,"ENTER SRC ACTION TYPE -"
      READ*,INX
      IF(IRUN.EQ.1)WRITE(5,*)INX
      IF(IRUN.EQ.3)PRINT*,INX
      IF(INX.EQ."T")GOTO111
      IF(INX.GE.0.AND.INX.LE.4)GOTO102
      PRINT*,"ACTION CODE ERROR - TRY AGAIN!"
  111 PRINT*,"VALID ACTION CODES."
      PRINT*,"ENTER 0 TO RETURN TO DECISION POINT"
      PRINT*,"      1 TO ADD A NEW SRC"
      PRINT*,"      2 TO DELETE A SRC"
      PRINT*,"      3 TO DISPLAY A SPECIFIC SRC"
      PRINT*,"      4 TO DISPLAY ALL SRC'S"
      GOTO14
C
  102 GOTO(501,600,800,500,1000),INX+1
  500 PRINT*,"ENTER SRC TO BE DISPLAYED -"
      READ502,ASRC
  502 FORMAT(1A10)
      IF(IRUN.EQ.1)WRITE(5,502)ASRC
      IF(IRUN.EQ.3)PRINT518,ASRC
  518 FORMAT(" ",1A10)
      BRRAY(2) = ASRC
      BRRAY(3) = 90909
      CALL GET(IFIT,BRRAY,BRRAY(1))
      IF(BRRAY(3).EQ.90909) GO TO 550
      PRINT 503 ,BRRAY(2)
  503 FORMAT(1X,"SRC=",A10,5X," ID  QTY")
      DO 505 I=3,45,2
      IF (BRRAY(I).EQ. 0)  GO TO 505
      IID=BRRAY(I)
      PRINT 504, ( IID,BRRAY(I+1)  )
  504 FORMAT(20X,I3,F5.0)
  505 CONTINUE
C
  550 PRINT*,"DISPLAY ANOTHER SRC?"
```

Figure Q-1. OVLY12 (BUILD) program code. (Continued next page.)

```
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
    1 FORMAT(1A1)
    8 FORMAT(" ",1A1)
      IF(INX.EQ."Y")GOTO500
      IF(INX.EQ."N")GOTO14
      PRINT2
      GOTO555
    2 FORMAT(" INCORRECT! RESPONSE MUST BE YES OR NO - TRY AGAIN.")
  550 PRINT 551 , ASRC
      BRRAY(3)=0
  551 FORMAT(1X,"SRC ",A10," NOT ON FILE")
      GO TO 555
  600 PRINT*,"ENTER SRC TO BE ADDED -"
      READ502,ASRC
      IF(IRUN.EQ.1)WRITE(5,502)ASRC
      IF(IRUN.EQ.3)PRINT518,ASRC
      BRRAY(2)=ASRC
      BRRAY(3) = 90909
      CALL GET(IFIT,BRRAY,BRRAY(1))
      IF(BRRAY(3).NE.90909) GO TO 610
      NN=1
      PRINT 7001
7001  FORMAT(1X,"ENTER WPN ID,QTY--0,0 IF DONE ")
  889 READ*, MM,AJ
      IF(MM.EQ.0) GO TO 888
      NN=NN+2
      BRRAY(NN)=MM
      BRRAY(NN+1)=AJ
      PRINT*,"NEXT-"
      GO TO 889
  888 CONTINUE
      CALL PUT(IFIT,BRRAY,400,BRRAY(1))
      M=IFETCH(IFIT,3LIRS)
      IF (M.EQ.4453)   GO TO 610
  603 FORMAT(1X,"SRC-",A10," ALREADY ON FILE")
      GO TO 612
  61. PRINT 603, ASRC
      BRRAY(3)=0
  612 DO 611 I=2,40
      BRRAY(I)=0
  611 CONTINUE
  620 PRINT*,"ADD ANOTHER SRC?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO600
      IF(INX.EQ."N")GOTO14
      PRINT2
      GOTO620
  800 PRINT*,"ENTER SRC TO BE DELETED -"
      READ1205,ASRC
1205  FORMAT(A10)
      BRRAY(2)=ASRC
      BRRAY(3)=90909
      CALL GET(IFIT,BRRAY, BRRAY(1))
```

Figure Q-1. OVLY12 (BUILD) program code (continued).

```
      IF(BRRAY(3).EQ.90909) GO TO 840
      CALL DLTE(IFIT,BRRAY(1))
      GO TO 15
 840  PRINT*,"SEC  ",BRRAY(2),"   NOT ON FILE   "
      BRRAY(3)=0
  810 PRINT*,"DELETE ANOTHER SEC?"
      READ1,INX
      IF(IRUN.EQ.1)WRITE(5,1)INX
      IF(IRUN.EQ.3)PRINT8,INX
      IF(INX.EQ."Y")GOTO800
      IF(INX.EQ."N")GOTO14
      PRINT2
      GOTO810
 1000 CALL REWND(IFIT)
 1100 CONTINUE
      CALL GETN(IFIT,BRRAY,BRRAY(1))
      M=IFETCH(IFIT,2LFP)
      IF(M.EQ.100B) GO TO 15
      IF(AHOLD.EQ.BRRAY(1)) GO TO 1200
      GO TO 1100
   15 CALL CLOSEM(IFIT)
      GO TO 20
 1200 PRINT 503,BRRAY(2)
      DO 1205 I=3,45,2
      IF(BRRAY(I).EQ.0) GO TO 1205
      IID=BRRAY(I)
      PRINT 504,(IID,BRRAY(I+1))
 1205 CONTINUE
      DO 1206 I=2,46
      BRRAY(I)=0
 1206 CONTINUE
      GO TO 1100
 901  CONTINUE
      CALL CLOSEM(IFIT)
      END
```

Figure Q-1. OVLY12 (BUILD) program code (concluded).

APPENDIX R

DISTRIBUTION

DISTRIBUTION LIST

Organization	No. of Copies
HQDA (SAUS-OR) Washington, D.C. 20310	1
Commander US Army Training and Doctrine Command Fort Monroe, VA 23651	
ATCD-SI (Mr Christman)	1
ATCD-AO	1
OCG (LTC Pokorny)	1
ATCD-C	1
Director USATRASANA ATTN: ATAA-D White Sands Missile Range, NM 88002	2
Commander Defense Documentation Center Cameron Station Alexandria, VA 22314	12
Commandant Personnel and Administration Center ATTN: ATCP-CD Ft Benjamin Harrison, IN 46216	3
Commander USA Logistics Center Ft Lee, VA 23801	
ATCL-C	1
ATCL-CF	2
ATCL IE	1
Commander USA Field Artillery School ATTN: ATSF-CTD-S Fort Sill, OK 73503	5

Organization	No. of Copies
Commander USA Infantry School ATTN: ATSH-CD-CS Fort Benning, GA 31905	3
Commander USA Armor School ATTN: ATSB-CD-S Fort Knox, KY 40121	3
Commander USA Aviation School ATTN: ATZQ-DA-A (Ms Godwin) Fort Rucker, AL 36360	2
Commander USA Engineer School ATTN: ATSEN-CTD-CS Fort Belvoir, VA 22060	2
US Army Research Institute - Field Unit Bldg 802 ATTN: Dr Jacobs Fort Leavenworth, KS 66027	1
Commander USAFORSCOM ATTN: AFOP-PL-WP Fort McPherson, VA 30330	2
Commander US XVIII Abn Corps ATTN: AFZADPT-O Fort Bragg, NC 28307	3
US Air Force Tactical Fighter Weapons Center/SATC ATTN: MAJ Blackledge Nellis AFB, NV 89191	2
Commander USA Signal School ATTN: ATSN-CTD-OR Fort Gordon, GA 30905	2

Organization	No. of Copies
Commander USA Combined Arms Combat Developments Activity Fort Leavenworth, KS 66027	
ATCA-ADC	1
ATCA-SW	5
ATCA-CF	1
ATCA-CC	1
ATCA-CA	13
Commander USA Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014	1
Commander USAECOM Systems Analysis Office (Mr Tyburski) Fort Monmouth, NJ 07703	1
Commander USAIID ATTN: ATISE-TD-TS-CD (LT Boyer) Fort Deven, MASS 01433	1
Commander USA Air Defense School ATTN: ATSA-CD-SS Fort Bliss, TX 79916	2
Commander USA Intelligence Center and School Fort Huachuca, AZ 86611	
ATSI-CTD-CS	2
ATSI-CTD-MS	1
Commander USA Quarter Masters School ATTN: ATMS-AR-C Fort Lee, VA 23801	1

Organization	No. of Copies
Commander USA Transportation School ATTN: ATSP-CTD-CS Fort Eustis, VA 23604	2
Commander USA Ordnance Center and School ATTN: ATSL-CTD-CS Aberdeen, MD 21005	2
Commander USA Institute of Military Assistance Dcomdt Cmt Tng Div Fort Bragg, NC 28307	1
Commander USA Military Police School ATTN: ATSJ-CTD-CS Fort McClellan, AL 36201	1
Commander Command and General Staff College ATTN: ATSW-TA Fort Leavenworth, KS 66027	5
Deputy Commander USAMSAA ATTN: AMXSY-T Aberdeen Proving Ground, MD 21005	1
HQ USAREUR Office Dep C/S Opns ATTN: (MAJ Lowe) APO New York 09055	2